ACCESS POINT

CONFIGURATION GUIDE

This manual describes how to configure and manage Access Point systems.

Product: Access Point™ Routers

Version: Version 2.3

Part Number: 610-0137-063

Lucent Technologies October 2000

IMPORTANT - PLEASE READ

NOTICE

The information in this manual is provided without warranty of any kind and is subject to change without notice. Lucent Technologies Inc. assumes no responsibility, and shall have no liability of any kind arising from supply or use of this publication or any material contained herein.

Company and product names are trademarks or registered trademarks of their respective companies.

Copyright © 1997 - 2000 by Lucent Technologies Inc. All rights reserved. No part of this publication may be reproduced, photocopied, or transmitted without express, written consent of Lucent Technologies Inc.

Lucent Technologies Inc.
50 Nagog Park
Acton, MA 01720
USA
http://www.lucent.com

FEDERAL COMMUNICATIONS COMMISSION WARNING

This device complies with Part 15 of the FCC Rules and Regulations. Operation is subject to the following two conditions:

- The device may not cause harmful interference.
- The device must withstand any interference received, including interference that may cause undesired operation.

The Access Point router has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC Rules and Regulations. These limits are designed to provide reasonable protection against harmful interference when this equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio and television communications. Operation of this equipment in a residential area is likely to cause interference in which case the user will be required to correct the interference at his or her own expense.

Shielded cables must be used with this unit to ensure compliance with the FCC Class A limits.

CONTENTS

Preface ..... **xix**

1 Product Overview **1**

About Access Point Systems 1
Architectural Overview 2
Bandwidth Management Capabilities 3
Class-Based Queuing and Link Sharing 3
CBQ at the Network Edge 4
Traffic Classes, Parameters and Packet Matching 4
Managing the Access Point System 7
About the Command Line Interface 7
About Access View 7
Supported RFCs and Working Drafts 8

2 Getting Started **11**

Defining Objectives 11
Routing Considerations 12
Managing Bandwidth 14
Locating Access Point Systems in Your Network 15
Setting Up the Access Point System 18
Logging In 18
Adding an Administrative Community 18
Running a Status Check 19

Establishing WAN Side Connectivity 19
Testing System Connectivity 20
Managing With the Command Line Interface 22
Using Command Completion 24
Changing the Screen Display Area 25
Getting Help on Commands or Parameters 26
Understanding the Interface Architecture 27
Getting Global Information on Interfaces 27
Refining Searches By Attributes 32
Changing Configuration Components Dynamically 33

3 Managing the System 37
Setting Up SNMP Communities 38
About Communities and Passwords 38
Using the SNMP Community Table 38
Designating an Administrative Community 42
Configuring SNMPv3 43
Configuring SNMP Traps 45
Restricting Telnet and Web Access 46
Configuring the System Name, Date and Time 46
Configuring the Watchdog Timer 46
Resetting the Software 47
Configuring and Using the File Transfer Table 47
About the File Transfer Table 47
Creating Table Entries 49
Displaying File Transfer Table Information 51
Transferring Files 51
Interpreting Errors 53
Saving the Configuration 54
Configuring Boot Order 55
Displaying a Configuration Summary 56
Displaying Configuration Changes 56
Displaying Configuration Status 57
NVRAM Compression 58
Saving Command Output In a File 59
Customizing a Pre-Login Banner 60
Control Character Restrictions 60

Configuring a Simple Message 60
Configuring a Formatted Message 61
Configuring a Multiline Full Text Border Message 61
Configuring the DNS Client 62
Configuring Your Site's Domain Name 62
Using Other DNS Commands 62
Getting Information on the DNS Client 64
Using the System Scheduler 66
Purpose and Environment 66
Using the Scheduler Parameters 66
Summarizing the Scheduler Status and Batch Jobs 68
Understanding Scheduler Passes 69
Using TCL Scripts 70
Creating TCL Scripts 70
Moving TCL Scripts to the Access Point System 70
Running Scripts 71
Autosourcing Scripts 71
Viewing TCL Scripts Available on the Access Point 71
Sample TCL Script 72
Managing System Events 73
Displaying the System Event Log Table 73
Setting Filtering Options By Level 74
Setting Output Filtering Options By Time 75
Displaying the Filtering Options 76
Setting Filtering Options By Subsystem 76
Displaying Subsystem Options 77
Configuring Event Parameters 81
Displaying Event Parameter Settings 83
Sample Event Filtering Configuration 84
Generating Configuration Script Commands 85
Using the Sync Command 86
Accessing Target Systems 86
Syncing Components With Security Features 86
Using sync 87
Configuring User Authentication 89
Configuring Users 90
Example: Adding Users 93
Enabling User Password Authentication 95

4 Configuring Interfaces 97

Configuring Ethernet and Gigabit Ethernet 98
- Changing Ethernet Settings 98
- Adding an IP Address to Each Interface 99
- Configuring the ARP Table Size 99
- Configuring Gigabit Ethernet 100

Configuring HSSI and MSSI Interfaces 101
- Setting Up the HSSI Interface 101
- Setting Up the MSSI Interface 102

Configuring T1 and E1 Links 103
- About T1/E1 Connections 103
- Displaying Slot Components 104
- Setting Up T1 Links 105
- Setting Up E1 Links 108
- Setting Up T1 and E1 Links in Loopback Mode 110
- Configuring Fractional T1/E1 112

Configuring T3 Interfaces 115
- Examining Physical Layer Defaults 115
- Setting Line Length 116
- Using the Payload Scrambler 117

Configuring ATM 117
- ATM Overview 117
- ATM Implementation in the Access Point Systems 120
- Setting Up ATM Interfaces 122
- Configuring the ATM Physical Interface 123
- Configuring the ATM Data Link Layer 128
- Configuring the IPOA Layer 137
- Setting Up ATM Subinterfaces 139
- Selecting Specific VCs for Traffic Classes 141
- Getting Information on CBQ Datalink Parameters 143

Configuring Integrated Services Digital Network 144
- Configuring the ISDN Interface 144
- Configuring Dial Services 151
- Verifying Proper Operation for the S/T Interface 157
- Verifying Proper Operation for the U Interface 158

Configuring Packet over SONET 159
- Configuring the SONET Physical Layer 159
- Configuring the Packet Over SONET Interface 161

SONET Path Labels 162
Configuring Frame Relay 163
Setting Up the Frame Relay Interface 164
Setting Up Frame Relay PVCs 164
Setting Up Frame Relay Subinterfaces 165
Selecting Specific DLCIs for Traffic Classes 167
Configuring PPP 168
Configuring Multilink PPP 170
Configuring PPP Authentication 172
Configuring Switched Multimegabit Data Service 175
Adding SMDS Interface 175
Configuring the Subnet 176
Displaying the SMDS Configuration 177
Configuring Static SMDS Addresses 177
Configuring the ARP Table Size 178
Displaying the SMDS and MAC Address Mapping 178
Configuring IP 178
Adding an IP Address to Each Interface 178
Using Unnumbered Interfaces 179
Using the Loopback Interface Address 180
Using the ip.sink Address 181
Displaying IP Addresses 181

5 Configuring Routing 183
Configuring IP Interface Parameters 184
IP Multinetting 184
Using Unnumbered IP Addresses 184
Using the Router Address 184
Verifying the Source of a Packet 185
Using Proxy ARP 185
Configuring General IP Parameters 187
Support for ICMP Redirect 187
Forwarding of Directed Broadcasts 187
Support for a Default Network 188
Displaying a List of Routes for a Specific Destination 188
Advanced IP Memory Configuration 189
Supported IP Protocols 190

Configuring Static Routes 191
Adding a Static Route 191
Adding a Default Route 191
Showing Static Routes 192
Configuring RIP 192
Activating and Deactivating RIP 193
Equal-Cost Multipath Load Balancing for RIP 193
Changing Default Implicit Policies 193
Configuring RIP Route Policies 195
Configuring OSPF 196
Overview 196
Configuring OSPF Interfaces 199
Configuring Area Aggregates 208
Setting Interface Cost 209
Debugging OSPF Connectivity Problems 209
Configuring OSPF Authentication 210
Configuring ISIS 211
Overview 211
Summary of Key Points 212
Configuring ISIS 213
Sample ISIS Configuration 216
Configuring BGP 218
Overview 218
Basic BGP Commands 219
Route Reflector Server 221
Overview of BGP Route Policy 222
Default BGP Behavior 222
BGP Route Selection 223
Complex BGP Route Policy 224
Route Policy 227
Overview 227
Route Policy Components 227
Policy Matching 228
Peer Groups 230
Route Policy Actions 232
RIP-Specific Capabilities 233
OSPF-Specific Capabilities 233
ISIS-Specific Capabilities 233

BGP-Specific Capabilities 234
Controlling Policy Execution With Policy Names 236
Configuring Administrative Distance 239
Using a Prefix To Influence Policy Execution 240
Sample Route Policy Scenarios 241
A RIP POP Network 241
A BGP Stub Network 244
A BGP Multihomed Network With MED and Local Preference 245
BGP Multihomed Network With Communities and AS Prepend 247
Creating and Selecting a Forwarding Policy 249
Defining Forwarding Actions 249
Creating a Forwarding Profile 251
Defining a CBQ Class 252
Configuring VRRP 253
About VRRP 253
How It Works 253
Configuring VRRP 255

6 Configuring Bridging 259
Overview 260
Converting a Router to a Bridge 260
Creating a Bridge-Router Configuration 261
Converting a Bridge to a Router 262
Configuring WAN Bridging 263
Bridging Over Frame Relay 263
Bridging Over PPP 264
Bridging Over ATM 264
Routing Over Bridged Encapsulation 266
Configuring Bridge Redundancy 267
About Virtual Bridge Redundancy Protocol 267
How It Works 267
Configuring VBRP 269

7 Configuring CBQ for Traffic Management 271
Traffic Management Overview 272
CBQ as a Bandwidth Manager 272
CBQ as a Firewall 273

About Traffic Classification 273
Configuring Traffic Classes 277
A Sample Class Definition 277
A Simple Configuration....278
Basic CBQ Components....281
Using Traffic Parameters....283
Classifying on Host or Host Range 284
Classifying on Applications....285
Classifying on Protocol or Protocol Range 286
Classifying on Port or Port Range 287
Using an Operator....288
Using Domain Name Classification 289
Using the Max Bandwidth Parameter....290
Setting the Q-elasticity Factor 290
Managing Traffic Class Queues with Random Early Detection (RED)....291
Classifying on BGP Community 293
Classifying on BGP AS Path 295
Classifying on Datalink 298
Using Traffic Parameters in Combination....300
Configuring Peer Classification Order 301
Using Stateless and Stateful Classification 305
Configuring Stateful Applications....308
Sample Configurations 310
Configuring Stateful Classification of FTP 313
Stateful Classification for TCP and UDP Applications....314
Configuring Stateful Classification for Multimedia 315
Managing CBQ....317
Setting Interface-Wide CBQ Parameters....317
Limiting the Number of Classes on an Interface 319
Collecting Statistics for Traffic Classes 319
Showing Statistics....321
Using AutoClass....324
Sample AutoClass Configuration 325
AutoClass Naming Conventions 329
AutoClass Name Restrictions....330
Using the ToS Parameters to Implement Diff-Serv....335

8 Configuring CBQ as a Firewall 339
About the CBQ Firewall 340
Stateful Classification and the CBQ Firewall 340
Examples: QoS Plus Security 342
Sample Firewall Configurations 345
Configuration 1- LAN to LAN Scenario 345
Configuration 2 - LAN to WAN Scenario 348
Configuration 3 - Securing the DMZ 353
Monitoring Stateful Connections 361
Viewing Information on Stateful Connections 361
Viewing Connection Information By Flow 362
Sorting Connection Information By Time 363
Viewing Summary Information About Connections 365
Configuring Connection Accounting 365
Specialized Firewall Capabilities 366
Classifying ICMP Packets 366
Classifying IP Fragments 372
SYN Flood Protection 377

9 Configuring QVPN Tunnels 379
Introduction to VPNs 380
What are VPNs? 380
Advantages of VPNs 380
Common VPN Applications 382
About Access Point QVPN 383
About the Virtual Tunnel Transport Layers 384
About IPSec 386
Overview: IPSec Site-to-Site Tunnels 387
Overview: IPSec Client Tunnels 388
Overview: IPSec Transport Mode 392
Configuring General VPN Parameters 393
Designating a Router Address 393
Configuring IPSec Security Profiles 394
Configuring IPSec Site-to-Site Tunnels 399
Configuring IPSec Client Tunnels 404
Monitoring IPSec Tunnels 409
Using the IKE Keepalive Protocol 414

How Keepalive Works....414
Configuring IKE Keepalive....415
Configuring IPSec Transport Mode Connections....417
Configuring IPSec Site-to-Site Transport Mode Connections....417
Configuring IPSec Client Transport Mode Connections....420
Configuring L2TP Tunnels....424
About L2TP....424
How L2TP Works....426
L2TP in the Virtual Tunnel Transport....427
Configuring L2TP....431
Configuring L2TP Tunnel Domains....432
Showing L2TP Information....435
Authenticating Dial-in Users....437
Configuring User Authentication....439
Configuring Address Pools....441
Configuring Client Groups....443
Configuring PKI....446
Configuring PKI Certificates....446
Configuring Certificate Revocation Lists....449

10 Managing With Access View....453
Overview of Access View....453
Starting Up Access View....454
About the Access View Window....455
Managing the Access Point System....457
Assigning a System Name and Setting the Date and Time....458
Setting Up SNMP Communities....459
Configuring DNS....460
Configuring WAN Interfaces....461
Configuring HSSI....461
Configuring Frame Relay....462
Configuring Class-Based Queuing....463
Overview of the Traffic Status Application....469
Accessing the Applets From Access View....469
Mapping Information From CBQ....471
Mapping the CBQ Tree to the Bandwidth Allocation Applet....472
Using the Traffic Equalizer....473

Traffic Class Graphs 482

A Advanced Configuration 485

Configuring the Network Address Translator 486
- The Translator in a Nutshell 486
- Configuring NAT 487
- Displaying NAT Configuration Information 490
- Using Dynamic or Static Bindings 491
- Displaying Translation Sessions 492
- Using Session Timers 493
- Displaying NAT Counters 493
- Enabling and Disabling NAT 493
- Sample Configurations 494

Configuring DHCP Relay 499
- About DHCP Relay 499
- Global DHCP Configuration Parameters 500
- Configuring Server Destinations 501
- Showing Destinations 502

Configuring the Network Time Protocol 503

Configuring RADIUS Authentication 507
- Configuring the RADIUS Server 508
- Configuring the RADIUS Client 511
- Displaying Information 515

Configuring and Using SSH 517
- Configuring the SSH Server 518
- Tuning the SSH Server Configuration 533
- Showing Information on the SSH Server 534
- Configuring the SSH Client 534
- About the SSH Server Applications 543
- Using the SSH File Transfer Protocol (SFTP) 543
- Using the Secure Copy Protocol (UNIX System) 548

Configuring IP Multicast 549
- About Multicast 549
- Configuring Multicast 549
- Configuring DVMRP 551
- Multicast over IP Tunnels 552
- Monitoring Multicast 554

Configuring Virtual LANs 557
Setting Up Your VLAN Configuration 557
Displaying the VLAN Configuration 559
Setting Up Ethernet Subinterfaces 559
Prioritizing VLAN Traffic 562
Configuring the AP Stack With SNMP 563

Figures

Figure 1 General Interface Architecture 3
Figure 2 Elementary Link Sharing Between Service Classes 4
Figure 3 Sample Network Diagram 13
Figure 4 Access Point 10 QoS Appliance, LAN-Only Implementation 16
Figure 5 Access Point Implemented as an Edge Device (LAN-WAN) 17
Figure 6 Access Point Implemented in a Regional POP 17
Figure 7 Status Command 19
Figure 8 Sample File Transfer Table 48
Figure 9 User Table 89
Figure 10 Status Screen, Access Point T1/E1 104
Figure 11 ATM Cell Structure 118
Figure 12 ATM and OSI Reference Models 119
Figure 13 Layers in the Access Point ATM Implementation 121
Figure 14 Access Point 45 in a Frame Relay Network 163
Figure 15 OSPF Areas and Backbone 197
Figure 16 Sample OSPF Neighbor Configuration 204
Figure 17 ISIS Domain 212
Figure 18 Sample ISIS Configuration 216
Figure 19 RIP Policy Route Example, Network Provider POP 242
Figure 20 BGP Stub Network 244
Figure 21 BGP Multihomed Network 245
Figure 22 Complex BGP Multihomed Network 247
Figure 23 VRRP Configuration Showing Default and Backup Gateways 254
Figure 24 Status Screen Following Router to Bridge Conversion 261
Figure 25 Bridging Router Configuration 261
Figure 26 Bridging Encapsulated Traffic 266
Figure 27 Bridge Configuration Showing Master and Backup Bridges 268
Figure 28 Sample CBQ Tree Showing Parent-Child Relationships 274
Figure 29 Traffic Management, WAN-to-LAN 275
Figure 30 Root Input and Output Trees on CBQ Instances 276

Figure 31 Simple CBQ Tree 279
Figure 32 BGP AS_Path Sample Configuration 296
Figure 33 Using Peer Classification Order 301
Figure 34 Using Peer Classification to Enhance Security 302
Figure 35 Stateful Connection Through the AP 305
Figure 36 Sample Stateful Classification for Telnet 311
Figure 37 Sample Stateful Classification, Without Firewall 313
Figure 38 Sample Stateful Classification for FTP 314
Figure 39 AutoClass Generation, 1 Level Subtree 327
Figure 40 AutoClass Generation, 3 Level Subtree 328
Figure 41 AutoClass Generation, 2 Level Subtree 332
Figure 42 CBQ on Input: Single WAN to LAN Scenario 343
Figure 43 CBQ on Input: LAN to Multiple WAN Scenario 344
Figure 44 Configuration 1 - LAN to LAN Scenario 346
Figure 45 Configuration 2 - LAN to WAN Scenario 349
Figure 46 Configuration 3 - Securing Hosts on the DMZ 353
Figure 47 Traffic Class Tree: DMZ Example 360
Figure 48 Sample Configuration: ICMP Fragment Filtering 368
Figure 49 Virtual Tunnel Transport Stacks 383
Figure 50 Virtual Tunnel Transport Stack Components 385
Figure 51 Site-to-Site Tunnels Among Three Access Point QVPN Systems 388
Figure 52 Sample IPSec-Based Client Scenario 389
Figure 53 IPSec Tunnels Summary 410
Figure 54 IPSec Security Associations Summary 411
Figure 55 IPSec Subinterface Summary 412
Figure 56 L2TP Connection: Enterprise LAN with L2TP LAC Client 425
Figure 57 L2TP Connection: ISP Service 425
Figure 58 L2TP Operations: Tunnel and Session Establishment 427
Figure 59 L2TP Components in the Virtual Stack 428
Figure 60 L2TP Stack: Single Tunnel and Session 428
Figure 61 L2TP Stack with Tunnels 430
Figure 62 Sample RADIUS Entry 439
Figure 63 Access View Login Dialog Window 454
Figure 64 Access View Window 455
Figure 65 Management Navigator Folder Tree 456
Figure 66 CBQ Policies 464
Figure 67 Add Traffic Class 466
Figure 68 Sample Data Mapped in the Bandwidth Allocation Applet 473

Figure 69 Sample Mapping With the Traffic Equalizer 474
Figure 70 Sample NAT and NAPT Configuration 494
Figure 71 Sample LSNAT Configuration 497
Figure 72 Access Point as a DHCP Relay Agent 499
Figure 73 RADIUS Configuration 507
Figure 74 User Table with Sample Entries 522
Figure 75 Basic Multicast Configuration 550
Figure 76 IP Multicast Via IP Tunnel 552

Tables

Table 1 Supported RFCs 8
Table 2 Standard CLI Commands 22
Table 3 Transferrable Components 86
Table 4 Variables and Their Uses 88
Table 5 Ethernet and Gigabit Ethernet Differences 98
Table 6 Configurable T3 Physical Layer Parameters and Values 116
Table 7 DS3 Attributes and Values 124
Table 8 DS3 Status Attributes and Values 125
Table 9 SONET Attributes and Values 127
Table 10 SONET Status Attributes and Values 127
Table 11 Traffic Descriptors and Service Categories 133
Table 12 SONET Attributes and Values 160
Table 13 RIP Completion Values 194
Table 14 Characters Permitted in Regular Expression Matching 226
Table 15 Peer Matching Per Protocol 229
Table 16 Common Route Policy Actions 232
Table 17 VRRP and VBRP Options 256
Table 18 VRRP and VBRP Options 270
Table 19 Statefully Classifiable Applications 308
Table 20 Invalid CBQ-Class Name Prefixes 331
Table 21 Parameters: IP Fragment Classification 376
Table 22 Security Profile Parameter Settings for Site-to-Site Connections 396
Table 23 Security Profile Parameter Settings for Client Connections 398
Table 24 IPSec Subinterface Objects 412
Table 25 Supported RADIUS Attributes 509
Table 26 SSH Command Options 541
Table 27 Common SFTP Commands 547

PREFACE

The Access Point product family comprises a set of bridging routers with advanced bandwidth management and VPN server capabilities. This product family contains four products.

- Access Point 450 router
- Access Point 1000 router
- Access Point 10 QoS appliance
- Access Point 100 QoS appliance

This guide explains how to install, configure and manage these products. Chapter titles and their subject areas are outlined below.

Chapter	Title	Areas Treated
1	Product Overview	Provides an overview of the AP's major features, architecture, and management
2	Getting Started	Provides guidelines on implementation issues, including routing, CBQ, and placement of the AP
3	Managing the System	Explains how to manage the AP
4	Configuring Interfaces	Explains how to configure the interfaces
5	Configuring Routing	Describes the AP's implementations of RIP, OSPF, BGP4, and router redundancy
6	Configuring Bridging	Describes the AP's implementations of LAN and WAN bridging as well as bridge redundancy
7	Configuring CBQ for Traffic Management	Explains how to use CBQ to provide IP QoS for IP-based traffic on AP interfaces

Chapter	Title	Areas Treated
8	*Configuring CBQ as a Firewall*	*Explains how to use CBQ to firewall incoming and outgoing IP-based traffic*
9	*Configuring QVPN Tunnels*	*Explains how to configure secure site-to-site virtual private networks across public networks*
10	*Managing With Access View*	*Explains how to manage the AP from a Web browser with Access View*
A	*Advanced Configuration*	*Explains how to configure advanced features, including:* • *Configuring the Network Address Translator* • *Configuring DHCP* • *Configuring the Network Time Protocol* • *Configuring RADIUS Authentication* • *Configuring IP Multicast*

Audience

The Access Point product is a bridging router with advanced bandwidth management and VPN capabilities. The product's overall design and default configuration values facilitate rapid setup and management. However, configuring any IP routing engine requires considerable experience with routers, hubs, bridges, and other networking devices. In particular, Lucent Technologies assumes that persons installing, configuring, and managing the Access Point product have several years experience in the following areas:

- Data communications protocols
- Routing and bridging protocols and practices
- IP networking
- Wide area communications using Frame Relay, PPP, ATM, T1/E1 or ISDN interfaces
- A working knowledge of VPN standards, including IPSec and L2TP

Experience configuring routing policies, downloading software images, configuring networking devices via serial line or Telnet, and managing SNMP-based devices is highly desirable.

Conventions

This guide uses the following conventions.

- *Italics* indicate command names and textual callouts.
- A vertical bar | between elements indicates that you must choose one of them.
- Square brackets [ ] indicate optional elements.
- Parentheses ( ) contain a group from which you make a selection.
- Angle brackets < > contain an element that you specify, such as a name or an address.
- Ellipses ... following a component, subcomponent, or parameter indicate that it can be repeated.

Safety Symbols

The following safety symbols are used to call attention to certain topics. To avoid equipment damage or possible injury, please devote special attention to these areas and follow all applicable procedures and warnings.

WARNING:



This symbol calls attention to issues or practices that could cause serious injury to yourself or others if safety precautions are disregarded.

CAUTION:



This symbol calls attention to issues or practices that could damage the equipment or cause loss of data if you disregard the required safety precautions.

Contacting Lucent Support

For questions or problems with the Access Point router, refer to this manual or to the Lucent Technologies NetworkCare Web site at:

http://www.lucent.com/networkcare

If you are not able to find the help you need, contact Lucent customer support at 1-800-237-0016.

Recommended Reading

The following documents provide more information.

- "Link-sharing and Resource Management Models for Packet Networks," Sally Floyd and Van Jacobson, IEEE/ACM Transactions on Networking, Vol.2, No.4, August 1995
- *Routing in the Internet*, Christian Huitema, Prentice-Hall PTR, Englewood Cliffs, New Jersey 07632, 1995
- *Internetworking with TCP/IP*, Douglas E. Comer, Prentice-Hall, Englewood Cliffs, New Jersey, 1995, 3rd Edition
- *Internet Routing Architectures*, Bassam Halabi, New Riders Publishing, Indianapolis, Indiana, 1997
- *OSPF: Anatomy of an Internet Routing Protocol*, John T. Moy, Addison Wesley Longman, Inc., Reading, Massachusetts, 1998

1

PRODUCT OVERVIEW

This chapter discusses the Access Point family's major features, architecture, and management. It contains the following sections.

About Access Point Systems 1

Architectural Overview 2

Bandwidth Management Capabilities 3

Managing the Access Point System 7

ABOUT ACCESS POINT SYSTEMS

The Access Point product line is a unique family of Internet access platforms that enable network providers to deliver managed IP QoS to their business class customers. Access Point systems (APs) offer full-featured IP routing, powerful bandwidth management and traffic shaping, secure virtual private networks (VPNs), and easy management in a single purpose-built platform. These capabilities enable the AP to deliver committed levels of service and fine-grained management control to network providers and their customers.

The APs support different network configurations, such as the:

- Access Point 450 VPN router which supports WAN access applications up to DS3 rates, including support as an ATM end station (PVC support only), and support for Frame Relay, SMDS, and PPP over T1 or E1 links.
- AP 1000 VPN router which adds higher packet-routing throughput, greater scaleability, and higher-speed interface options to the AP 450

VPN router's capabilities.

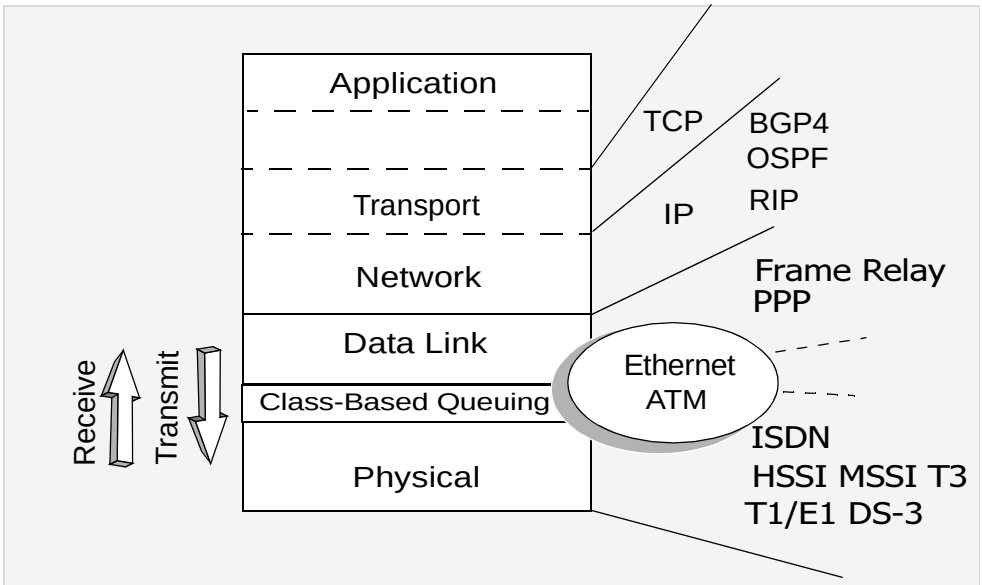
- AP 10 QoS appliance which only supports bridged 10 Mbps Ethernet LAN-to-LAN connections and VPN gateway capabilities.
- AP 100 QoS appliance which supports routed or bridged 100 Mbps Ethernet LAN-to-LAN connections and VPN gateway capabilities.

ARCHITECTURAL OVERVIEW

The Access Point system (AP) implements a layered architecture that supports a variety of physical layer connection schemes and data links. Figure 1 provides a generalized picture of the Access Point architecture. This figure shows the functional relationships between supported connection schemes, data links, and routing protocols within the OSI reference model. The main points to consider here are the following:

- The AP supports Frame Relay or PPP connections over any of the WAN interfaces listed.
- Ethernet and ATM connections are also supported. These networks incorporate both physical and data-link layer characteristics.
- Each of the routing protocols listed is supported on all Access Point platforms. The exception is the Access Point 10 QoS appliance, which is a bridge-only LAN device.
- Class-Based Queuing (CBQ) is a distinct layer supported for each local and wide area network type. From a conceptual standpoint, CBQ can be represented as sitting on top of the physical layer.

Figure 1 General Interface Architecture



BANDWIDTH MANAGEMENT CAPABILITIES

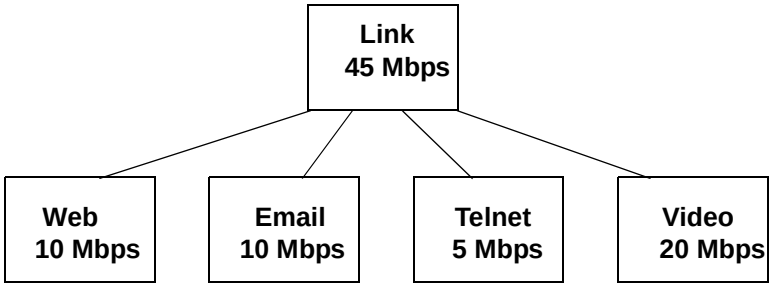
The AP's implementation of Class-Based Queuing (CBQ) provides administrators with flexibility and fine-grained control of bandwidth utilization on local and wide area links. CBQ is a *link-sharing mechanism* based on the 1995 paper, *Link-sharing and Resource Management Models for Packet Networks* (Floyd and Van Jacobson).[1]

CLASS-BASED QUEUING AND LINK SHARING

The general idea behind link sharing is to apportion available bandwidth on a deterministic basis between different users, or services, each of which is represented by a *traffic class*. An elementary picture of link sharing is shown in Figure 2. Here, the total amount of bandwidth available on the link (45 Mbps) is divided among a number of traffic classes. In this simple case, each traffic class represents a specific service, such as HTTP (Web). However, CBQ allows for detailed class definitions involving a number of factors. For a detailed description of CBQ and instructions on configuring traffic classes, see Chapter 7.

Unlike weighted fair queuing and other methods for IP traffic, CBQ works by providing a fixed amount of bandwidth to defined types of IP traffic. CBQ accomplishes this using a set of *traffic classes* bound together in a hierarchical structure. Often called a *traffic classification hierarchy,* this structure is separately configured on each of the AP's interfaces.

Figure 2 Elementary Link Sharing Between Service Classes



CBQ AT THE NETWORK EDGE

The AP is designed as an access router or edge device, located at the perimeter of the enterprise LAN and wide area network. It can also be implemented in bridging-only mode as an intranet bridge, and used to manage traffic between contiguous LAN segments, or between the enterprise LAN and an internal Web server farm. (See Chapter 2 for several sample implementations of the AP).

When implemented as an edge device, the AP can use CBQ to classify and schedule IP-based traffic in two directions: from the LAN toward the WAN, and from the WAN link inbound. Moreover, you can configure CBQ to classify and schedule packets in two directions (input and output) on each interface. See Chapter 7 for a detailed discussion of CBQ.

TRAFFIC CLASSES, PARAMETERS AND PACKET MATCHING

As mentioned earlier, Class-Based Queuing (CBQ) is a link-sharing mechanism that matches IP packets against the *traffic classes* and overall *traffic classification hierarchy* you implement on a link. You can think of each class as a kind of traffic profile. Each class specifies a set of match criteria. IP packets are compared against these criteria as they pass through Access Point interfaces. When a match occurs, the matched packet is scheduled — that is, it is pathed onto some portion of the CBQ tree. Usually (but not always) the class has a certain amount of bandwidth associated with it; this bandwidth is avail-

able to all packets that match (belong to) this class.

CBQ provides a host of different parameters for classifying traffic. For examples showing how to use these parameters, see the sample configurations provided in Chapter 7.

Whether you use all or a subset of these parameters depends on your business needs. They enable you to build traffic classes based on the following criteria:

- *Source or destination IP address*
- *Source or destination port*
- *Application*
- *Type of protocol used (for example, TCP, UDP, ICMP, or some other)*
- *Source or destination domain name*
- *The value in the ToS byte of IP packets*

What makes CBQ powerful is that you can use any or all of these classification parameters in combination. For instance, CBQ is designed to make it easy to create classes that match definitions like these.

- *Match traffic that originates in a particular host range, is destined for a single host in my enterprise, and is using a single type of service.*
- *Match all Web traffic bound for a particular URL, such as www.microsoft.com, and mark it with a particular Diff-Serv (ToS) value.*
- *Match all outbound UDP-based traffic from a set of hosts to a particular Web server farm that has a particular Diff-Serv (ToS) setting.*

You can also schedule traffic in different ways, and according to different criteria that may change while the AP is running. For example, administrators can also allow certain (or all) classes to borrow additional bandwidth as needed, and can limit borrowing to a fixed rate. Additional (borrowed) bandwidth can be made available to needy classes, depending on overall bandwidth utilization on the link. Examples of the borrowing/non-borrowing behavior that can be configured are illustrated by the three statements that follow.

- *Always grant this class 10 Mbps, and ONLY 10 Mbps. Drop all packets that match to this class, but exceed the10 Mbps allocated to the link.*
- *Grant this class 10 Mbps, but allow the class to borrow up to 5 Mbps more if extra bandwidth is needed, and is available on the link.*
- *Guarantee the class 10 Mbps, and allow it to borrow all it needs above this, depending on the availability of unused bandwidth.*

To prevent traffic that does not fall into any defined class from being dropped, you can define a default class. For information on defining default classes, see "Default Classes" on Page 282.

Individual traffic classes and the overall traffic classification hierarchy can be configured at the command line or via Access View, a Web-based configuration management tool. In addition, traffic classes can be generated automatically using the AutoClass feature as explained in "Using AutoClass" on Page 324.

MANAGING THE ACCESS POINT SYSTEM

The AP contains an SNMP agent that supports SNMP commands (including SNMPv3) so the product can be managed via any of the following methods:

- Using the command line interface, or CLI. See Chapter 2 for information on managing with the CLI.
- Using Access View, a Web-based management utility. See Chapter 10 for complete information.
- Using the MIB browser incorporated into third party management platforms, such as HP OpenView or SunNet manager.

ABOUT THE COMMAND LINE INTERFACE

The ASCII-based command line interface provides a powerful and expedient way to configure and monitor the AP.

Initial connection to the AP is via a serial port. After assigning an IP address to the AP, administrators can connect to and remotely manage the device via Telnet.

See Chapter 2 for information about connecting to the AP and using the CLI.

ABOUT ACCESS VIEW

Access View enables you to configure, monitor and manage Access Point products via compatible browsers. Netscape Communicator 4.05, and Microsoft Internet Explorer V4.0 and 4.01 are recommended. Any browser that supports the following features should work as well:

- Frames
- Java
- HTML V 3.2

See Chapter 10 for information on configuring and managing the AP via a Web browser.

SUPPORTED RFCS AND WORKING DRAFTS

The AP supports the following RFCs:

Table 1 Supported RFCs

Area	RFC Number	Topic
IP	RFC 791	Internet Protocol
PoS	RFC 2615	Packet over SONET
PPP MP	RFC 1990	The PPP Multilink Protocol
CHAP	RFC 1994	Challenge Handshake Authentication Protocol
RADIUS	RFC 2138/2139	Remote Access Dial In User Services
Gigabit Ethernet	RFC 1643	Managed Objects for Ethernet-like Interfaces
IP Tunnel	RFC 2667	IP Tunnel MIB
PPP	RFC 1661 RFC 1662	The Point to Point Protocol Managed Objects in HDLC-like Framing
TCP	RFC 793/1323	Transmission Control Protocol TCP Extensions for High Performance
UDP	RFC 768	User Datagram Protocol
ICMP	RFC 792	Internet Control Message Protocol
ARP	RFC 826	Ethernet Address Resolution Protocol
BootP Client/Gateway	RFC 951	Bootstrap Protocol
RIP-2	RFC 1723	RIP Version 2
BGP-4	RFCs 1771, 1745	A Border Gateway Protocol 4
OSPF-2	RFC 2328	OSPF Version 2
ISIS	RFC 1195	Use of OSI IS-IS for Routing in TCP/IP and Dual Environments
VRRP	RFC 2338	Virtual Router Redundancy Protocol
SNMP	RFC 1157	Simple Network Management Protocol

Area	RFC Number	Topic
TFTP	RFC 783	TFTP Protocol (Revision 2)
Telnet	RFC 854	Telnet Protocol Specification
IP over Ethernet	RFC 1042	Standard for the Transmission of IP Datagrams over IEEE 802 Networks
IP over PPP	RFC 1332	The PPP Internet Protocol Control Protocol (IPCP)
IP over Frame Relay	RFC 1490	Multiprotocol Interconnect Over Frame Relay
IP over ATM	RFCs 1483, 2225	Multiprotocol Encapsulation Over ATM Adaptation Layer 5 Classical IP and ARP Over ATM
DS3	RFC 1407	Definitions of Managed Objects for the DS3/E3 Interface Type
SONET	RFC 1595	Definitions of Managed Objects for the SONET/SDH Interface Type
RTSP	RFC 2326	Real Time Streaming Protocol
H.323	ITU-T Recommendation H.323	"Visual Telephone Systems and Equipment for Local Area Networks Which Provide A Non-Guaranteed Quality of Service"
ATM	RFC 1695	Definitions of Managed Objects for ATM Management Version 8.0
ATM	RFC 2320	Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2 (IPOA MIB)
VPN	RFC 1321	MD5 Digest Algorithm
VPN	RFC 2104	HMAC: Keyed-Hashing for Message Authentication
VPN	RFC 2401	Security Architecture for the Internet Protocol
VPN	RFC 2403	The Use of HMAC-MD5-96 within ESP and AH
VPN	RFC 2404	The Use of HMAC-SHA-1-96 within ESP and AH

Area	RFC Number	Topic
VPN	RFC 2405	The ESP DES-CBC Cipher Algorithm With Explicit IV
VPN	RFC 2406	IP Encapsulation Security Payload (ESP)
VPN	RFC 2407	The Internet IP Security Domain of Interpretation for ISAKMP
VPN	RFC 2408	Internet Security Association and Key Management Protocol (ISAKMP)
VPN	RFC 2409	The Internet Key Exchange (IKE)
VPN	RFC 2410	The NULL Encryption Algorithm and its use With IPSec
VPN	RFC 2412	The Oakley Key Determination Protocol
VPN	RFC 2451	ESP CBC-Mode Cipher Algorithms

2

GETTING STARTED

This chapter discusses some of the issues you should consider before installing and configuring the Access Point system (AP) in your network.

Defining Objectives 11
Routing Considerations 12
Managing Bandwidth 14
Locating Access Point Systems in Your Network 15
Setting Up the Access Point System 18
Managing With the Command Line Interface 22
Getting Help on Commands or Parameters 26
Understanding the Interface Architecture 27

DEFINING OBJECTIVES

Before you install the AP, make an outline of the business requirement you need to satisfy. Following are a few of the questions you should raise.

- *What is the main goal of your business and implementation? If you are a network provider, do you plan to install an AP in a regional POP? Are you using the AP to guarantee certain levels of access to a Web server farm?*
- *Do you need to monitor and manage bandwidth utilization on a local or wide-area link? If yes, is your chief interest in controlling inbound or outbound traffic?*
- *Are you implementing the AP as an intranet bridge, or as a router?*

Which routing protocols do you need to run?

- *Which wide-area data links does your business currently support?*
- *Do you intend to implement the AP as an edge device, or within the core of your corporate network?*

These are only some of the questions you may want to raise before installing and configuring your AP.

ROUTING CONSIDERATIONS

Routing is another area in which up-front planning can help produce desired results. The questions and suggestions that follow are meant to help in the planning process. However, this treatment is not intended as a primer on network routing. Lucent assumes that the person installing the AP is network-knowledgeable, experienced in configuring standard IP routing protocols, knows where the AP will be installed in the network.

STEP 1: BEGIN AT THE BEGINNING. BRIEFLY DESCRIBE HOW THE AP WILL PARTICIPATE IN YOUR OVERALL ROUTING STRATEGY.

For example, before configuring the device you should be able to craft a statement like the following:

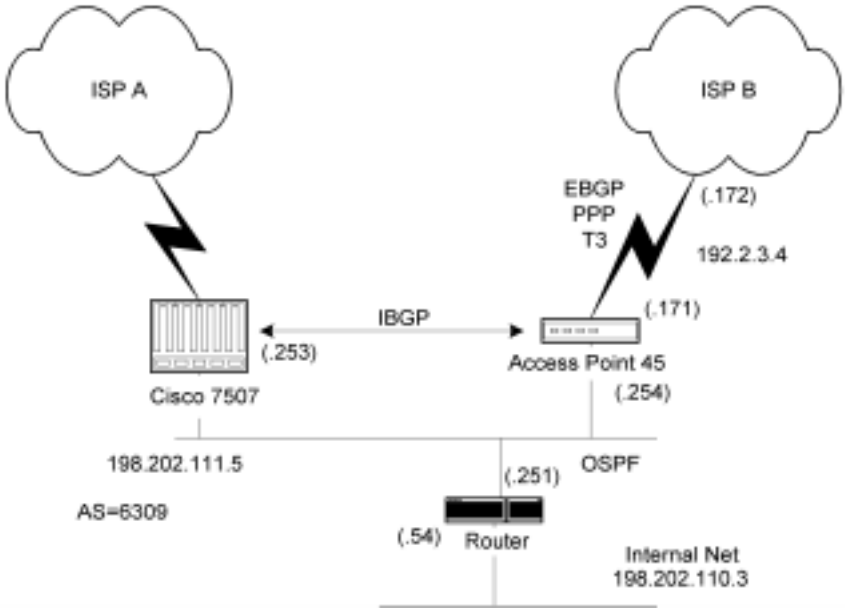
> *The AP will be used to provide a redundant path to the Internet through <our network provider>. BGP will be used to receive all and advertise some routes to <provider>. The AP will run OSPF and use static routes internally.*

STEP 2: DRAW A DETAILED NETWORK MAP.

It is usually helpful to draw a network map showing the major devices in your network and the connections between them. Each device should show the number of interfaces in use and their associated IP addresses. In addition, all routing protocols, WAN protocols and WAN speeds should be clearly identified.

Figure 3 is for illustrative purposes only. Use whatever diagram you are comfortable with that provides at least the amount of information shown here.

Figure 3 Sample Network Diagram



STEP 3: RECORD AS MUCH DETAILED INFORMATION AS POSSIBLE.

After drawing the diagram, fill in as many of the blanks as possible. Again, the more thoroughly you plan, the fewer decisions you will make on-the-fly. Chances are, your installation will also be quicker and go more smoothly. To be complete, your diagram should address the following questions and provide specific answers to them.

1. What are the local addresses of the AP and their associated interfaces? You can provide this information in the form of a table. The table shown here is derived from the information given in Figure 3.

Interface	IP Address	Mask
eth.1	198.202.111.5	255.255.255.0
hssi.2	192.2.3.4	255.255.255.0

2. List the routing protocols you will use, and how you will use them.
3. For OSPF or BGP, list the local AS number.
4. Will the AP be set up with a single Internet connection or will it be multi-

homed to the Internet?

- If multihomed, how many peers will the AP have? What are their addresses?
- What type of routers are the peers?

5 Do you plan to use multipath routing or load balancing? If yes, describe your implementation of these features in detail.

6 Will you implement any policies to advertise routes between protocols? If yes, describe these policies in detail.

MANAGING BANDWIDTH

Class-Based Queuing (CBQ) is a separately configurable aspect of each Access Point interface. CBQ is basically a fast and extremely sophisticated mechanism whose purpose is to decide what to do with IP-based packets before transmitting them.

The decision-making process consists of *classification* and *scheduling* operations. Each packet is *classified by* matching it against a set of criteria configured for the interface. Packets *scheduling* is the second phase of the operation. Here, packets are assigned some portion of available bandwidth, as well as other characteristics. For example, some packets may be allowed to borrow a certain amount of additional bandwidth. Others may not be given any bandwidth at all, unless spare bandwidth is available.

CBQ's richness and flexibility make full coverage of the topic impossible here. For complete and detailed information on configuring the feature, see Chapter 7, "Configuring CBQ for Traffic Management."

CBQ makes decisions on a packet-by-packet basis, according to the following criteria:

- *The information contained in the packet header*
- *The way in which you've configured CBQ on the interface*

Like the address space on an envelope, the header area of an IP packet contains all the information needed to get it to its destination. Information contained in packet headers includes the following:

- *Packet source and destination address*
- *Protocol number (or range)*
- *Service number (or range)*
- *Source or destination Domain name*
- *The value assigned to the ToS byte*

CBQ allows you to classify IP packets based on any combination of these factors. It is an extremely rich solution to the problem of traffic management. However, you have to know what your needs are to manage bandwidth in ways that bring your enterprise the most benefit. Reviewing your needs and goals is a good way to get the most out of the features offered by CBQ.

LOCATING ACCESS POINT SYSTEMS IN YOUR NETWORK

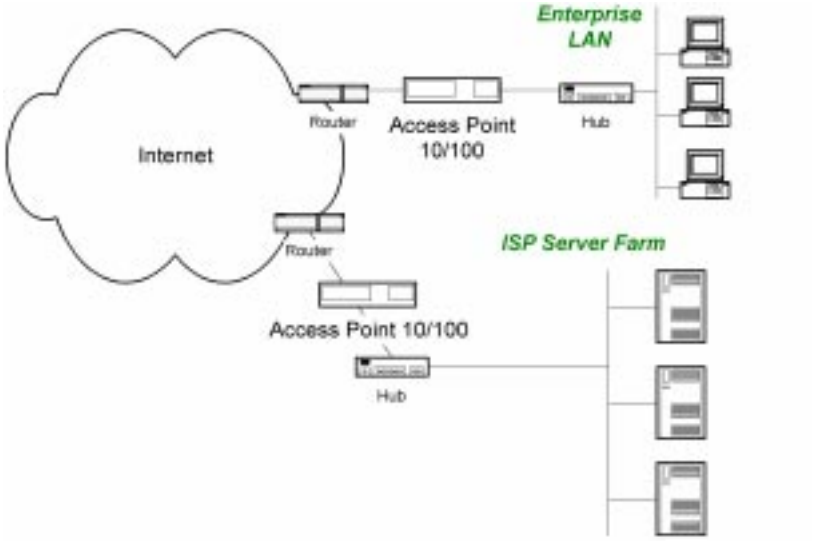
All networks are not created equal. The nature of your network setup should depend on the kinds of goals you are trying to serve, the number of users you must support, the amount of throughput and special services these users require, and other factors. Otherwise, it is unlikely that what you have in place will serve your present needs, or scale well to meet new demands later on.

The AP can be deployed in a variety of ways. Three general cases are the following:

- *as a LAN-only, intranet bridge*
- *as an edge device, or access router*
- *as a premises router within regional POP*

Figure 4 shows a LAN-only implementation. Notice that each AP in the figure sits between a private network and a router.

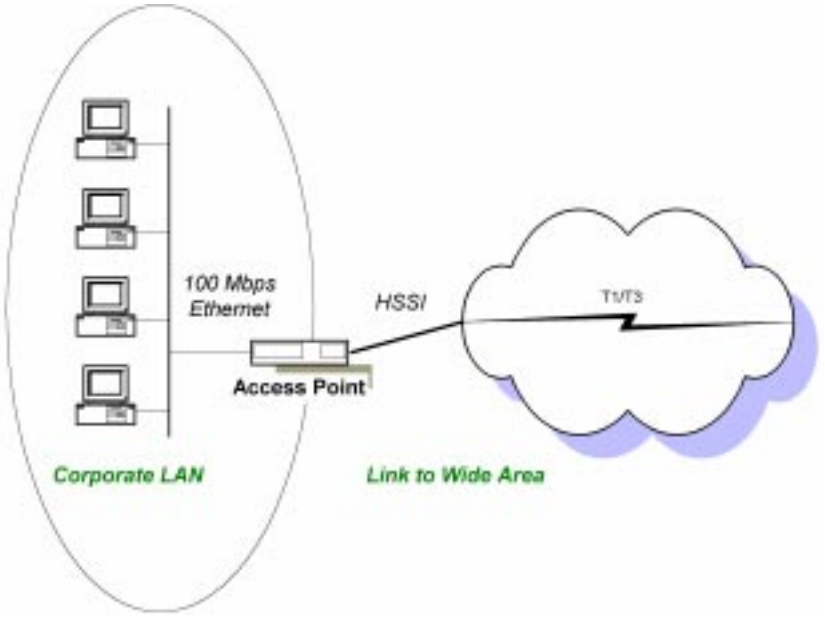
Figure 4 Access Point 10 QoS Appliance, LAN-Only Implementation



In this scenario, the AP is deployed in two locations as a bandwidth management platform. Its purposes are to manage bandwidth utilization on the link between an ISP server farm, and the link between the enterprise LAN and the public internet. In each case, the AP need not perform any routing, since each AP sits behind a router located at the network edge.

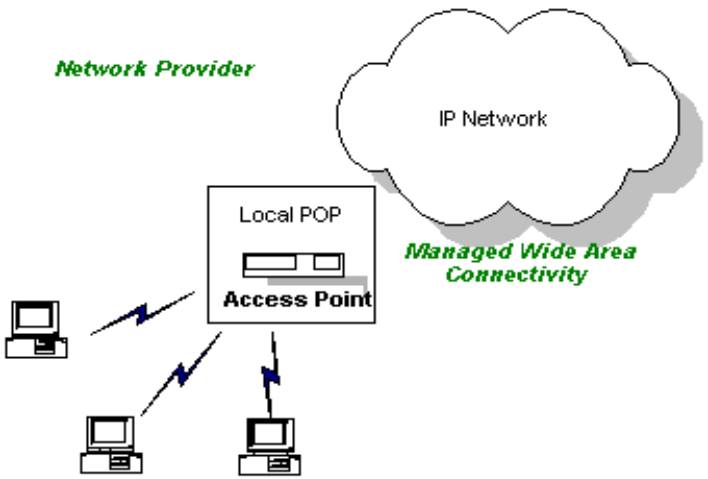
However, the AP can function as an edge device or access router, and simultaneously provide bandwidth management for traffic on the link. This is the case in Figure 5. Here, the AP provides bandwidth management for traffic transmitted on the wide-area link.

Figure 5 Access Point Implemented as an Edge Device (LAN-WAN)



The AP is also commonly implemented as a premises router within regional POPs so network providers can provision bandwidth among users on the wide area link. Since CBQ can be configured to allow users to borrow additional bandwidth up to the limit available on the link, providers can use the AP to charge for varying levels of service.

Figure 6 Access Point Implemented in a Regional POP



SETTING UP THE ACCESS POINT SYSTEM

You should perform these steps before configuring wide area connectivity but after you have connected a local management console, powered up the AP, and established LAN side connectivity as part of the installation process.

- *Logging In* (Page 18)
- *Adding an Administrative Community* (Page 18)
- *Running a Status Check* (Page 19)
- *Establishing WAN Side Connectivity* (Page 19)

Review these sections before you attempt to bring the AP on line in your network.

LOGGING IN

If your AP is connected correctly, you will see the following indicators when you first power up the AP:

- The appropriate LEDs are lit.
- The banner and Username prompt are displayed on the console screen. Some messages will appear on the console screen during boot-up. Then you will see this login prompt.

  Lucent AccessPoint (hw: 1.7; sw: V2.2.0.R1 12/24/1999 02:02:14:00)

  Username:
  Password:

  The default configuration for the device requires no passwords. So enter *admin* for username and press *Return* when prompted for a password.

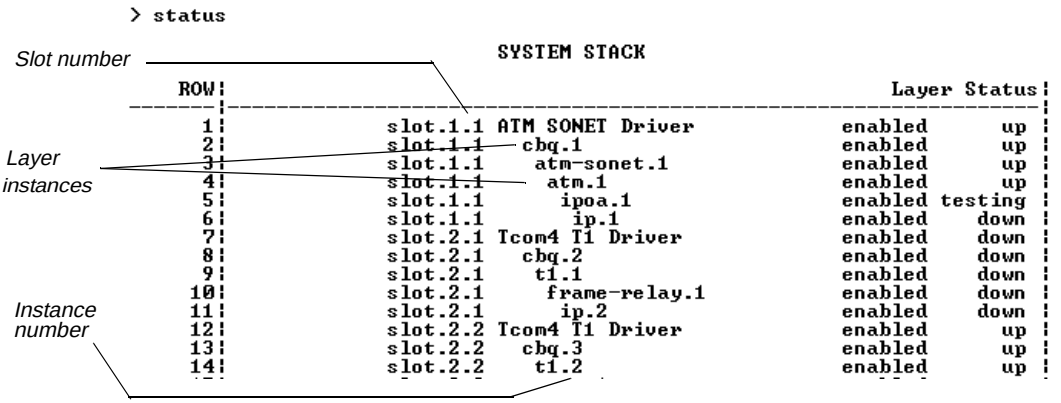
ADDING AN ADMINISTRATIVE COMMUNITY

For the sake of security, you should designate an administrative community as soon as you log in. See Page 42 for details.

RUNNING A STATUS CHECK

First, check to see how the cards resident in your device map to the device's slots. Do this by running the *status* command. The screen shown in Figure 7 shows the status of a device with an Ethernet card in slot 2, and HSSI cards in slots 3 and 4.

Figure 7 Status Command



Notice the layers associated with each slot. You will need to know which instances map to the device's slots when you configure layers later on.

ESTABLISHING WAN SIDE CONNECTIVITY

ADD AN IP ADDRESS

Next, add an address on the WAN side of the AP. Adding an IP address to this interface is required to bring up the wide area link.

```
> add ip.2 address.192.168.2.1
```

To check that IP addresses have been added successfully, display the IP address table using this command.

```
> show ip.* address-table summary
```

TESTING SYSTEM CONNECTIVITY

After you have connected your AP and set up your addresses, you can test system connectivity using the *ping* and *traceroute* commands.

USING THE PING COMMAND

The *ping* command checks IP connectivity with a remote host by sending an Internet Control Message Protocol (ICMP) echo request datagram to elicit an ICMP echo response from the specified IP host or gateway. The *ping* command has the following syntax:

```
ping [-s] [-S <source address>] <host> [-c <count>] [-l <length>] [-t <timeout>]
  [-d <delay>] [-r <start length> <end length>]
```

Using the *ping* <host> command, where <host> is the system name or IP address sends one ICMP request and results in a "<ip address> is alive" or "no answer from <ip address>" message. The following table describes the options that you can set:

-c	Specifies the number of packets sent, or the number of range sweeps sent (see -r option below). The default is 1.
-d	Specifies the amount of time in milliseconds to delay after receiving a reply before sending the next request.
-l	Specifies the length of the packet in bytes. This length includes the IP and ICMP headers. The default is 64 bytes.
-r	Specifies a range of request packet lengths to sweep through. One request of each length is sent starting from the specified start length and incrementing by one until reaching the specified end length. The -r option overrides the -l option. When the -c option is specified with the -r option, the -c option defines the number of times to repeat the sweep.
-s	Continuously sends one request per second (a one second delay) until you type q on the keyboard. This option also changes the form of output generated. The length, IP address, sequence number, and round-trip time in milliseconds is printed for each response received.
-S	Specifies the IP address of the source host.
-t	Specifies the amount of time (in milliseconds) to wait before determining that the remote host is not reachable. The default is 1000 milliseconds.

You can interrupt the *ping* command by typing any character on the keyboard. Whenever the above options result in more than one request, the minimum, average, and maximum round-trip time in milliseconds is displayed when the command completes or is terminated.

USING THE TRACEROUTE COMMAND

The *traceroute* command traces the route a packet takes to a remote host. Each hop is a port on a router on the network. The *traceroute* command has the following syntax:

```
traceroute [-nv] [-m <max ttl>] [-p <port number>] [-q <number queries>]
  [-s <source address>] [-t <tos>] [-w <wait>] <host> [<data size>]
```

The following table describes the options and parameters that you can set:

-m	Specifies the maximum number of hops. The default is 30 hops.
-n	Print hop addresses numerically rather than symbolically (and numerically save a nameserver address-to-name lookup for each gateway found on the path).
-p	Specifies the UDP port number used in the probe.
-q	Specifies the number of packets to send.
-s	Specifies the IP address of the source host.
-t	Specifies the Type of Service as an integer from 0 - 255. The default is 0.
-v	List verbose output (so received ICMP packets other than TIME_EXCEEDED and PORT_UNREACHABLE are listed).
-w	Specifies the time to wait for a response, in seconds. The default is 3 seconds.
host	Specifies the destination host name or IP address.
data size	Specifies the packet length. The default is 40 bytes.

MANAGING WITH THE COMMAND LINE INTERFACE

Table 2 lists standard CLI commands and gives examples of their use. Once you begin using these commands, you will find them easy to understand.

Table 2 Standard CLI Commands

Command	What it is For
add	Add a component to the system or to a table <component> [<subcomponent>...] [<parameter> [=] <value>...] > add ip bgp peer.<number>
colwidth	Sets the number of columns supported by the terminal, optionally displays current value if number is not specified <number columns> > colwidth 90
config	Change the value of a parameter <component> [<subcomponent>...] <parameter>[=]<value> [<parameter>[=]<value>...] > config bridge transparent-bridge backup-mode enabled
describe	Print help text for one or more parameters <command> > describe ipoa.1 arp-config arp-cache-timeout
disable	Disable (turn off) an interface <interface> > disable eth.2
enable	Enable (turn on) an interface <interface> > enable eth.2
help	Get help on commands and editing keys <command> > help sync
log	Save output from commands to a file <command> [<command-specific argument>] > log open 127.0.0.1 maria/test.tcl

ping	Check IP connectivity with a remote host <host> [<timeout>] > ping <ip address> > ping <system name>
remove	Remove a component from the system or from a table <component> [<subcomponent>...] > remove cbq.1 traffic-class.finance-out
reset	Execute a software reset, optionally resetting box-wide parameters [defaults] > reset
show	Print a report on a (sub)component of the system <component> [<subcomponent>...] [<report>]
stack	Add a layer component or IP instance to a system interface <component> [<subcomponent>...] > stack slot.4.1 cbq.3 eth.2 ip.3
status	Show the status of the system interfaces <command> > status
sync	Transfers selected objects from one configured AP to another system <component> <target IP address>
telnet	Connect to a remote host in VT100 emulation mode <host> > telnet <ip address> > telnet <host name>
traceroute	Trace the route a packet takes to a remote host [-nv] [-m <max_ttl>] [-p <port#>] [-q <nqueries>] [-s <src_addr>] [-t <tos>] [-w <wait>] <host> [<data size>] > traceroute <ip address> > traceroute <system name>

USING COMMAND COMPLETION

Command completion is a basic feature of the CLI, one that makes it easier to navigate through configuration options. You can use this feature to list available command parameters that have lexical characteristics in common.

COMPLETING COMMAND STRINGS

For example, to type the *config* command (config), just type the first three letters and hit the spacebar. The rest of the command appears at the command line. If you were to type just the first two letters, the CLI would respond in this fashion.

```
> co
ambiguous command name "co"
completions are:
        colwidth
        config
atw-bottom> co
```

Here, as in other areas, the CLI prompts you with possible completions when it cannot determine what you are trying to enter.

Let's say, however, that you are in the process of configuring a BGP routing policy. To configure a BGP peer, you might enter the following:

```
> add ip bgp peer.4.1.1.2
```

To do this, you would actually need to type only fragments of the command string, separated by spacebars:

```
>ad [spacebar] i [spacebar] bg[spacebar] p[spacebar]
```

If you enter enough letters before each spacebar entry, the CLI is smart enough to move you forward. As you configure the AP at the command line, you will find this feature saves a great deal of time.

DISPLAYING RELEVANT PARAMETERS

Command completion is especially valuable as a tool for displaying available parameters in cases when you are unsure what these are. As an example, let's say you are configuring an instance of CBQ (CBQ.1), and are unsure which CBQ parameter to use. In this case, you would type traffic-class and hit the spacebar. Since the CLI organizes all related parameters lexically, this action displays all relevant CBQ parameters:

```
> show cbq.1 traffic-class
completions are:
        traffic-class-display.<inst>
        traffic-class-tree
        traffic-class.<inst>
        traffic-classes
```

Once you display the parameters, their names indicate which to choose for your present needs.

COMMAND RECALL

Command recall is another ease-of-use feature that is basic to the AP. As you enter commands, the CLI saves the last twenty command strings in its buffer. To recall any of these, press the up-arrow key until you reach the desired string. You can then modify and enter this string by pressing return.

CHANGING THE SCREEN DISPLAY AREA

Enabling *more* breaks up lengthy displays of information into viewable chunks. To enable *more*, use this command:

```
> more enabled
```

By default, the first 24 lines are displayed, followed by the word More, as shown here:

```
--More--
```

With *more* enabled, pressing any key displays the next 24 lines, and so on, until you reach the end of the display. You can type Ctrl+c or “q” to abort the display. If you want to display more or less than the default 24 lines, adjust the terminal lines setting specific to your terminal, as follows:

```
> termlines 30
```

Since *more* works on a per-console basis, settings for the feature are not saved to NVRAM. If you are running with More enabled, the feature is disabled on reboot.

GETTING HELP ON COMMANDS OR PARAMETERS

You can use the *help* and *describe* commands to get information on CLI commands and parameters.

USING HELP

Help gives you access to detailed information on many CLI commands. To get help, enter the *help* command followed by the command name.

```
> help sync
```

USING DESCRIBE

Use *describe* to get detailed information on a MIB object. Enter the full text of the command relating to the object, as shown here.

```
> describe ipoa.1 arp-config arp-cache-timeout
```

```
MIB name:                xipIfArpCacheTimeout
Type:                    number, min=2, max=60
Access required:         readwrite
SNMP OID:                1.3.6.1.4.1.838.3.4.1.3.1.1

The timeout value, in minutes, that is used to age out
entries from the ARP cache (the ipNetToMediaTable).
```

The *describe* command displays information on the object from the MIB. The information includes the MIB object name and OID, object type, required access, and a brief description of the object's function.

NOTE *If you are using Access View, you can get help on any object by clicking on the blue question mark that accompanies it on the interface. See Chapter 10 for detailed information on using Access View.*

UNDERSTANDING THE INTERFACE ARCHITECTURE

All interfaces supported by the AP share a common architecture. Within this architecture, each interface has a *general case* and *specific instances*, a set of *groups*, and *attributes*.

You can think of interface instances as specific cases of an interface type. Groups provide a way of organizing attributes. Attributes are the values that, on the whole, make up the interface's configuration.

From an abstract perspective, each interface can be represented as follows:

interface-name.<instance> <group> <attribute>

where

interface-name	specifies the type of interface
instance	specifies the instance number for the interface object
group	provides a way of organizing a set of attributes
attributes	are the set of specific values associated with the interface

GETTING GLOBAL INFORMATION ON INTERFACES

You can get global information on all interface objects by issuing the *status* command. The display lists all interface instances detected in the AP.

Each of the interface objects has an instance number. This number may conform to the physical slot in which the instance resides. Issuing the *status* command is the easiest way to see how interface objects relate to the slots in the box.

> status

SYSTEM STACK

ROW	Layer	Status	
1	slot.1.1	Ethernet Driver	enabled up
2	slot.1.1	cbq.1	enabled up
3	slot.1.1	eth.1	enabled up
4	slot.1.1	bridge.1	enabled up
5	slot.1.1	ip.1	enabled up
6	slot.2.1	Ethernet Driver	enabled up
7	slot.2.1	cbq.2	enabled up
8	slot.2.1	eth.2	enabled up
9	slot.2.1	bridge.2	enabled up
10	slot.2.1	ip.1	enabled up
11	virtual	Virtual Tunnel Transport	enabled down
12	virtual	cbq.3	enabled down
13	virtual	ipsec.1	enabled down
14	virtual	ip.4	enabled down
15	virtual	Virtual Tunnel Transport	enabled up
16	virtual	iptnl.1	enabled up
17	virtual	Virtual Tunnel Transport	enabled up
18	virtual	cbq.4	enabled up
19	virtual	l2tp.1	enabled up
20	virtual	l2tp.1.1	enabled up
21	virtual	l2dial.1.1	enabled up
22	virtual	ip.5	enabled down

In this case, the AP has two slots. Ethernet, CBQ, bridge, and IP are present on each slot and each instance of these objects matches the slot number in which they reside.

The following display shows the interfaces organized quite differently.

> status

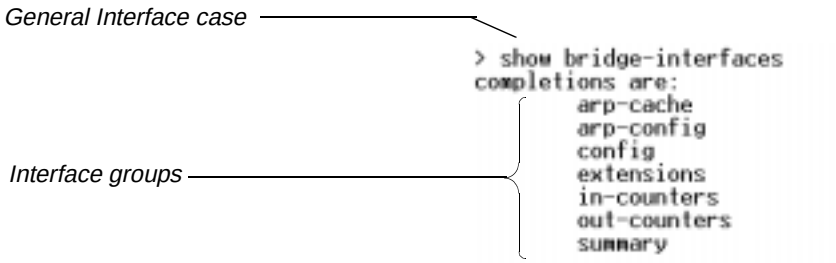
SYSTEM STACK

ROW	Layer	Status	
1	slot.1.1	Ethernet Driver	enabled up
2	slot.1.1	cbq.1	enabled up
3	slot.1.1	eth.1	enabled up
4	slot.1.1	ip.1	enabled up
5	slot.3.1	LMC5200 HSSI Driver	enabled up
6	slot.3.1	cbq.2	enabled up
7	slot.3.1	hssi.1	enabled up
8	slot.3.1	ppp.1	enabled up
9	slot.3.1	ip.2	enabled up
10	slot.4.1	Ethernet Driver	enabled up
11	slot.4.1	cbq.3	enabled up
12	slot.4.1	eth.2	enabled up
13	slot.4.1	ip.3	enabled up
14	virtual	Virtual Tunnel Transport	enabled up
15	virtual	cbq.4	enabled up
16	virtual	ipsec.1	enabled up
17	virtual	ip.6	enabled down

In this case, three of the AP's slots are populated. Slot 3 contains a HSSI card, and an instance of IP (ip.2). However, this slot contains no instance of Ethernet. The next instance of Ethernet (eth.2) therefore appears in slot 4.

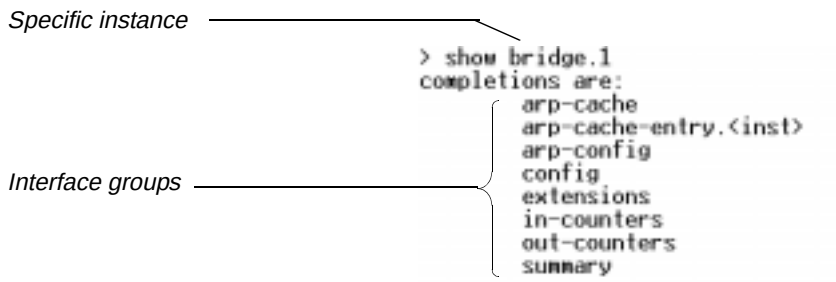
SHOWING INTERFACE GROUP INFORMATION

To show the groups of information available for an interface, type the *show* command and an interface name, followed by the spacebar as shown here.



Note that this example shows the groups subsumed beneath the general case *bridge-interfaces*. If you enter a *show* command for a specific bridge instance (such as bridge.1), you will see the same set of groups information.

This structure allows you to access general information on any interface, or specific information on any interface instance.



SHOWING ATTRIBUTE INFORMATION

As mentioned earlier, information for each group is stored in a set of attributes. Like group names, many attributes are common across all Access Point interfaces.

The CLI uses tables to display attribute information. To show the information for any group, type a command in this form:

> show interface.<instance> group

For example, to show information in the summary group for interface IP.1, Ethernet.1, ATM-DS3.1, or HSSI.1, enter one of these commands:

> show ip.1 summary

> show eth.1 summary

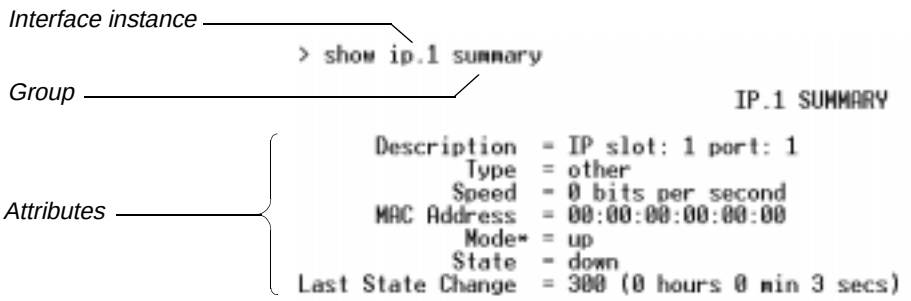
> show atm-ds3.1 summary

> show hssi.1 summary

Attributes associated with the interface-instance and group entered are presented in a table. For example, the attributes associated with the summary group for IP are:

Description	**MAC Address**	**State**
Type	**Mode**	**Last State Change**
Speed		

The CLI displays these same attributes for each supported interface. For instance, here is a sample display of the attributes associated with the IP summary group.



The same attributes are displayed for the ATM summary group (shown below)

and for all other interfaces.

```
                                    ATM.1 SUMMARY

      Description  = ATM VC slot: 1 port: 1
             Type  = atm
            Speed  = 44182000 bits per second
      MAC Address  = 00:00:00:00:00:00
             Mode* = up
            State  = up
Last State Change  = 283754 (0 hours 47 min 17 secs)
```

SETTABLE AND READ-ONLY ATTRIBUTES

Every interface has both settable and read-only attributes associated with it. The CLI uses an asterisk (*) to indicate which attributes are settable, as shown below.

```
> show eth.1 config
```

```
                                    ETH.1 CONFIG

                MTU* = 1500 bytes
       Up Down Trap* = enabled
              Alias* = ""
       Config Speed* = mbps100
        Duplex Mode* = half
          Media Type  = tx
Capture Effect Resol* = disabled
```

Some of the attributes shown in this display are shared by other interface types. For example, a *show atm.1 config* command may produce the following:

```
>show atm.1 config
```

```
                                    ATM.1 CONFIG

          MTU* = 9180 bytes                    VPI Bits* = 2
Up Down Trap* = enabled                        VCI Bits* = 10
       Alias* = ""                  Remote IP Address* = 0.0.0.0
    Max VPCs* = 0                Remote Interface Name* = unknown
    Max VCCs* = 4096                        Empty Cells* = idle
         VPCs  = 0                           Cell Align* = hec
         VCCs  = 1
```

The MTU, Up Down Trap, and Alias attributes are common to ATM, Ethernet, and all other interface objects. However, the remaining attributes are specific to ATM.

REFINING SEARCHES BY ATTRIBUTES

The definitions used to configure certain features of the AP contain many attributes. For instance, a single CBQ class may incorporate a dozen attributes, including name, bandwidth allocation, source and destination addresses, and others. Similarly, each IPSec security profile contains many pieces of information.

The *where* command provides you with the ability to make targeted searches within a set of information, based on specific attributes. You can use this command in combination with the *show* command to find specific information quickly. The command allows you to make comparative searches, using these operators:

- ==
- !=
- >
- >=
- <
- <=

For example, let's say you need to find information on an L2TP domain configuration that uses a secret of 0xffffff. The following command locates this information quickly:

```
> show l2tp.1 domain-configurations domain-configuration where secret ==
0xffffff
```

In this case, the search retrieves the information as shown here.

```
L2TP.1 DOMAIN-CONFIGURATIONS DOMAIN-CONFIGURATION WHERE SECRET == 0XFFFFFF

INTERFACE|      IDENTIFIER|                 Admin State|
                           |                Authentication|
                           |                       Secret|
                           |               Tunnel Security|
                           |           Tunnel Hello Interval|
                           |             Tunnel Idle Timeout|
                           |                   Control RWS|
                           |        Control Retransmissions|
                           |             Payload Sequencing|
                           |                   Payload RWS|
                           |          Delayed Ack Timeout|
                           |           Reassembly Timeout|
                           |                        Status|
---------------|----------|-----------------------------|
        l2tp.1|     kansas|                      enabled|
                           |                    challenge|
                           |                    0xfffffff|
                           |                         none|
                           |                           10|
                           |                           60|
                           |                           30|
```

CHANGING CONFIGURATION COMPONENTS DYNAMICALLY

You can use the *stack* command to make dynamic changes to the configuration of the AP's physical or virtual interfaces. These changes can include adding new layer components or IP instances to an interface, or moving existing components from one location to another.

Use of the *stack* command is required for some aspects of VPN configuration. The *stack* command lets you add and link interface or protocol components with a single command string, or to move existing components to different Access Point slots.

ADDING AND REMOVING COMPONENTS WITH THE STACK COMMAND

Using *stack* entails entering a set of linked layers (that is, a "stack") and supplying (or deleting) a layer in a specific location.

As an example of adding a component, refer to the status display shown below.

Note that no IP instance exists in slot.4.1.

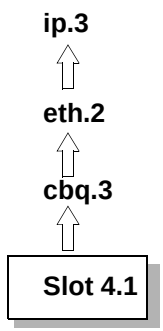
ROW		Layer	Status
1	slot.1.1 Ethernet Driver	enabled	up
2	slot.1.1 cbq.1	enabled	up
3	slot.1.1 eth.1	enabled	up
4	slot.1.1 ip.1	enabled	up
5	slot.3.1 LCM5200 HSSI Driver	enabled	down
6	slot.3.1 cbq.2	enabled	down
7	slot.3.1 hssi.1	enabled	down
8	slot.3.1 frame-relay.1	enabled	down
9	slot.3.1 ip.2	enabled	down
10	slot.4.1 Ethernet Driver	enabled	up
11	slot.4.1 cbq.3	enabled	up
12	slot.4.1 eth.2	enabled	up
13	virtual Virtual Tunnel Transport	enabled	up
14	virtual cbq.4	enabled	up
15	virtual ipsec.1	enabled	up
16	virtual ip.4	enabled	down
17	virtual ipsec.1.1	enabled	up
18	virtual ip.5	enabled	down
19	virtual ipsec.1.2	enabled	down
20	virtual ip.6	enabled	down

In this case, *stack* can be used to add an IP instance to slot 4.1, as follows:

```
> stack slot.4.1 cbq.3 eth.2 ip.3
```

As shown in this example, the order in which you list components specifies the linkages that exist between them. Adding an IP instance (ip.3) entails listing the components, in order, from the lowest layer (driver) upward.

The command used here specifies a chain of uplinks beginning with a CBQ instance (cbq.3) and ending with a new IP instance (ip.3). The linkages established by the command establish the functional relationships between the components shown at right.



The *stack* command also makes it easy to add a new component without specifying an instance number, or to delete existing components. For instance, adding the IP instance in the previous example could have been accomplished by specifying ip.new, as in this command:

```
> stack slot.4.1 cbq.3 eth.2 ip.new
```

Here, the IP instance added — whether ip.3, ip.23, or some other — depends

on the next instance that is available. Removing the IP instance just added to slot 4.1 can be accomplished as follows:

```
> stack slot.4.1 cbq.3 eth.2 none
```

Using none in lieu of an instance removes the existing IP instance at this level in the protocol stack.

ADDING MULTIPLE CBQ INSTANCES

You can add multiple CBQ instances to each AP stack. Doing so allows you to use each instance for a specific purpose. This can be helpful in cases where more than one person is administering the AP.

A sample *stack* command showing addition of multiple CBQ instances is the following:

```
> stack slot.2.1 cbq.1 cbq.10 cbq.11
```

SYSTEM STACK

ROW		Layer	Status
1	slot.1.1 LMC5200 HSSI Driver	enabled	down
2	slot.1.1 cbq.1	enabled	down
3	slot.1.1 hssi.1	enabled	down
4	slot.1.1 frame-relay.1	enabled	down
5	slot.1.1 ip.1	enabled	down
6	slot.2.1 Ethernet Driver	enabled	down
7	slot.2.1 cbq.2	enabled	down
8	slot.2.1 cbq.10	enabled	down
9	slot.2.1 cbq.11	enabled	down
10	slot.2.1 eth.1	enabled	down
11	slot.2.1 ip.2	enabled	down

In this example, each CBQ instance can be used for a different purpose. For instance, CBQ.2 can be configured as an anti-spoofing mechanism, CBQ.10 can implement other filtering criteria, and CBQ.11 can manage bandwidth on remaining traffic.

MORE ON STACK

While use of the *stack* command is required when configuring site-to-site tunnels (see "Add and Configure IPSec Subinterfaces" on Page 399), you can also use *stack* to make dynamic changes to other aspects of the AP's configuration. Other areas that require use of the *stack* command include:

- Replacing a Frame Relay data link with PPP
- Converting a router to a bridge, or vice versa
- Creating a bridge-router configuration

See "Configuring PPP" on Page 168 for instructions on replacing Frame Relay with PPP. For information on configuring the AP as a bridge, or converting a bridge to a routing configuration, see "Configuring Bridging" on Page 259.

GETTING HELP

Extensive command line help on *stack* is available by entering:

```
> stack help
```

3

MANAGING THE SYSTEM

This chapter explains how to manage the Access Point system (AP). It contains the following sections:

Setting Up SNMP Communities 38
Configuring SNMPv3 43
Restricting Telnet and Web Access 46
Configuring the System Name, Date and Time 46
Configuring the Watchdog Timer 46
Resetting the Software 47
Configuring and Using the File Transfer Table 47
Saving the Configuration 54
Configuring Boot Order 55
Displaying a Configuration Summary 56
NVRAM Compression 58
Saving Command Output In a File 59
Customizing a Pre-Login Banner 60
Configuring the DNS Client 62
Using the System Scheduler 66
Using TCL Scripts 70
Managing System Events 73
Generating Configuration Script Commands 85
Using the Sync Command 86
Configuring User Authentication 89

SETTING UP SNMP COMMUNITIES

The AP supports SNMP v1, v2, and v3 packets as inbound requests, outbound responses, and outbound traps. This section contains information on adding, modifying, and deleting SNMP community names and passwords. Using SNMP v3 requires additional configuration. See Page 43 for instructions on configuring SNMPv3.

By default, security is disabled on the AP. When security is disabled, anyone with knowledge of one of the device's IP addresses can log in and change its configuration. For this reason, it is recommended that you designate an administrative community when you first log in to the AP as explained on Page 42.

ABOUT COMMUNITIES AND PASSWORDS

SNMP community names are used to provide access to the AP. Community names are specified in the following format:

<community-name>!<password>

The AP has three default community users and passwords. These are *public*, *admin*, and *user.*

Anyone logged on with a community having read/write privileges (such as *admin*) can make changes to the SNMP Community table. To provide secure access to the AP, designate one community as the administrative community as described on Page 42.

USING THE SNMP COMMUNITY TABLE

The Access Point SNMP Community is located under services. The table allows up to ten entries.

The first three Community Name entries are the AP defaults; an additional default entry (*private*) is displayed if you upgraded your software. The *public* and *private* community names can be used by any SNMP manager to access the AP. The default *admin* and *user* community names have null passwords. When you log in to the AP, use admin or user as your user name, and enter no password.

To display the SNMP community table, enter this command.

```
> show services snmp community-table
```

```
                                SERVICES SNMP COMMUNITY-TABLE

INDEX|    Community/user Name|                                    Net Address|
     |                 Access|                                       Net Mask|
     |           Snmp Version|                                V3 Security Models|
     |      V3 Security Modes|                                        View Bits|
     |    Authentication App|                                Authentication Key|
     |           Privacy App|                                        Privacy Key|
     |  Privacy Key Algorithm|                                         Row Status|
------|-----------------------|------------------------------------------------|
    1|         public!--------|                                         0.0.0.0|
     |                readonly|                                     255.255.255.0|
     |   snmpv1+snmpv2+snmpv3|                         snmpv1+snmpv2+snmpv3|
     |noAuthNoPriv+authNoPr>|                                              s1|
     |              authNone|                                              --|
     |              privNone|                                              --|
     |       privKeyAuthApp|                                          active|
    2|        private!--------|                                         0.0.0.0|
     |               readwrite|                                     255.255.255.0|
     |   snmpv1+snmpv2+snmpv3|                         snmpv1+snmpv2+snmpv3|
     |noAuthNoPriv+authNoPr>|                                              s2|
     |              authNone|                                              --|
     |              privNone|                                              --|
     |       privKeyAuthApp|                                          active|
    3|         admin!--------|                                         0.0.0.0|
     |               readwrite|                                     255.255.255.0|
     |   snmpv1+snmpv2+snmpv3|                         snmpv1+snmpv2+snmpv3|
     |noAuthNoPriv+authNoPr>|                                              s3|
     |              authNone|                                              --|
     |              privNone|                                              --|
     |       privKeyAuthApp|                                          active|
    4|          user!--------|                                         0.0.0.0|
     |                readonly|                                     255.255.255.0|
     |   snmpv1+snmpv2+snmpv3|                         snmpv1+snmpv2+snmpv3|
     |noAuthNoPriv+authNoPr>|                                              s4|
     |              authNone|                                              --|
     |              privNone|                                              --|
     |       privKeyAuthApp|                                          active|
```

The community name can also be configured for certain IP addresses as an added level of safety. Note that the password portion of the community name is not displayed for security reasons; it is replaced by dashes (--------). The Access parameter indicates the privilege level of the community. There are four access levels: *readonly*, *writeonly*, *readwrite*, or *notaccessible*. All of the default community names are either *readonly* or *readwrite*.

The Net Address parameter indicates which source IP address the community is accepted on. The default value is 0.0.0.0; communities with this value will be accepted for any address.

The other fields display information about SNMPv3 configuration parameters that are described in “Configuring SNMPv3” on Page 43. Notice that the Authentication Key and Privacy Key are not displayed on the screen. For secu-

rity reasons, a NULL string ("") appears in place of each password.

To display summary information for a particular SNMP community, enter the community index along with this command.

```
> show snmp community.<index> summary
```

ADDING AND MODIFYING A COMMUNITY

Add a community name and password by adding a user to the System SNMP Community table.

1 To add and modify communities for the system, first log in with administrative community read/write privileges by typing *admin* at the *Username:* prompt and press *Return* at the *Password:* prompt.

```
Username: admin
```

2 Add a new community name and password with the *add* command as shown below:

```
> add services snmp community.6 community/user-name MIS!Jeepers \
  access readwrite
```

Here, *add* is the command name, *community* is the component, 6 is the instance of the community component, *MIS* is the name given to the community, and *Jeepers* is the password.

Community names are specified in the following format:

```
<community-name>!<password>
```

The community name can be up to 256 characters consisting of any printable characters but must not contain spaces.

NOTE *Periods (.) are allowed in community names and passwords. They are accepted from an SNMP manager, but are not accepted at the Login screen.*

You can specify the other parameters (such as the Access parameter) as arguments to the *add* command, or you can add a community and then configure these parameters using the *config* command. If you do not specify an access level for the community, it receives readonly access by default. Refer to "Configuring SNMPv3" on Page 43 for information about configuring SNMPv3 parameters.

To display the SNMP Community table entry for the new community

name, use this command:

```
> show services snmp community.6
```

```
                                SERVICES SNMP COMMUNITY.6

  Community/user Name* = MIS!--------
          Net Address* = 0.0.0.0
               Access* = readwrite
             Net Mask* = 255.255.255.0
        Snmp Version** = snmpv1+snmpv2+snmpv3
  V3 Security Models* = snmpv1+snmpv2+snmpv3
   V3 Security Modes* = noAuthNoPriv+authNoPriv+authPriv
             View Bits* = s6
  Authentication App* = authNone
  Authentication Key* = ""
          Privacy App* = privNone
          Privacy Key* = ""
Privacy Key Algorithm* = privKeyAuthApp
            Row Status* = active
```

If you enter the *show services snmp community.<index>* command from any community name other than the designated administrative community or the default admin community, an empty table is displayed.

MODIFYING A COMMUNITY

You can modify an existing entry in the SNMP Community table by using the *config* command as follows.

```
> config services snmp community.6 net-address 127.0.0.1 access readwrite
```

This command string modifies the community definition added earlier by substituting the net-address 127.0.0.1 for 0.0.0.0.

Refer to “Configuring SNMPv3” on Page 43 for detailed information about configuring SNMPv3 parameters.

REMOVING A COMMUNITY

To remove a community, use the *remove* command, as shown in this example:

```
> remove services snmp community.6
```

DESIGNATING AN ADMINISTRATIVE COMMUNITY

Designate an administrative community by configuring the *admin-community* parameter. The designated administrative community can view and modify the SNMP Community table. In addition, the administrative community has the privileges indicated by the Access parameter. Any community with read/write privileges can be the administrative community; however, there can be only one administrative community.

If you create an administrative community and then delete it without specifying another administrative community, the system reverts to the default admin community when you exit to ensure continued access to the system. Any user with read/write privileges can view and modify the SNMP Community table if no administrative community is configured.

You can add a new name to the community table for use as administrative community, or you can use a community name that already exists. In either case, the community must have read/write privileges.

1. Use the *config* command as shown in the following example, which adds a password to the default admin community:

   ```
   > config services snmp community.3 \
     community/user-name admin!password
   ```

2. You can designate this community as the administrative community by setting the admin-community parameter, for example:

   ```
   > config services snmp admin-community admin!password
   ```

3. To specify a different community as the administrative community, use the *config* command with the admin-community parameter, as shown here:

   ```
   > config services snmp admin-community MIS!Jeepers
   ```

SAFEGUARD THE IDENTITY OF THE ADMINISTRATIVE COMMUNITY

The SNMP Community table does not indicate which community is the administrative community; therefore, write down the administrative community name and keep it in a safe place.

When you use the *admin-community* parameter to designate an administrative community, the command takes effect immediately. If you are logged in with a different community name, you cannot access the SNMP Community table.

When you exit and log on as an administrative community, you can display the SNMP Config report to see the administrative community's name and password by using the following command:

```
> show services snmp config
```

CONFIGURING SNMPV3

This section explains how to secure the AP and restrict user access by configuring SNMPv3 on the AP. Configuring SNMPv3 requires you to perform these steps:

1. Configure an engine ID.
2. Configure a user for SNMPv3.
3. Specify authentication and privacy applications.
4. Configure authentication and privacy keys from the password.
5. Restrict user access. For example, restrict SNMPv1 and SNMPv2 users to minimal use or allowing access via CLI only.

The instructions given in this section are specific to SNMPv3. See "Setting Up SNMP Communities" on Page 38 for general instructions on adding, modifying and deleting SNMP community names and password.

CONFIGURING AN ENGINE ID

Using an engine ID provides an extra level of security by ensuring that PDUs are only received by intended recipients. The engine ID used is a unique name in the SNMP domain.

To configure an engine ID, use this command:

```
> config services snmp engineid-cfg <text string>
```

CONFIGURING A USER FOR SNMPV3

Use the following commands to configure a user for SNMPv3.

```
> config services snmp community.10 community/user-name chart
```

```
> config services snmp community.10 v3-security-modes authNoPriv
```

Specifying Authentication and Privacy Applications

Creating a user with SNMPv3 access also requires you to specify an authentication application and a privacy application, and a privacy key algorithm, as shown here:

```
> config services snmp community.10 authentication-app authMd5
> config services snmp community.10 privacy-app privNone
> config services snmp community.10 privacy-key-algorithm privKeyMd5
```

Generating Authentication and Privacy Keys

In addition to supporting manual creation of authentication and privacy keys, the AP can generate these keys, based on the following pieces of information:

- The engine ID
- The application type
- A unique password

After providing this information, you can cause the AP to choose authentication and privacy keys for the user by issuing these commands.

```
> config services snmp community.10 authentication-key-from-password
> config services snmp community.10 privacy-key-from-password privkey
```

It is important to remember that keys generated in this way depend on the engine ID and application type chosen. If this information changes after keys have been generated, keys must be regenerated using the new information.

Restricting User Access

To restrict user access to the AP, disable SNMPv1 and SNMPv2 connections for a specific community, as follows:

```
> config services snmp community.10 snmp-version !snmpv1 + !snmpv2
```

You can also restrict certain users to command line (CLI) access, by assigning the loopback address as the net address as shown:

```
> config services snmp community.10 net-address 127.0.0.1
```

CONFIGURING SNMP TRAPS

If you are using SNMP traps, to support generic traps or send events as traps, you must configure the trap destination. To configure the trap destination, you must configure the network address with this command:

```
> config services snmp trap-destination.<instance> net-address <IP address>
```

The default community/user name is public. The SNMP version is SNMPv1. If you are using SNMPv3, you can configure the authentication and privacy applications in addition to the authentication and privacy keys.

To display the trap destinations:

```
> show services snmp trap-destination-table
```

```
                    SERVICES SNMP TRAP-DESTINATION-TABLE

INDEX|   Community/user Name|                                  Net Address|
     |              SNMP Version|                                 Udp Port|
     |         Authentication App|                       Authentication Key|
     |               Privacy App|                             Privacy Key|
     |   Privacy Key Algorithm|                                  Row Status|
-----|--------------------------|-----------------------------------------|
   1|                   public|                                  0.0.0.0|
     |                    snmpv1|                                      162|
     |                  authNone|                                       --|
     |                   privNone|                                       --|
     |              privKeyAuthApp|                                  active|
```

OPTIONAL CONFIGURATION PARAMETERS

You can configure the agent to provide additional information, such as interface names and descriptions, in the standard LinkUp/Down traps with this command:

```
> config services snmp trap-options extendedLinkTrapInfo
```

If you are using SNMPv1 or SNMPv2, you can configure the agent's IP address for all outbound traps with this command:

```
> config services snmp trap-source-ip-address <IP address>
```

If this address is left at 0.0.0.0, then any one of the active IP addresses is used. If a router address is available, that will be used first.

NOTE *Once this address has been learned, it remains in use until the system is reset.*

RESTRICTING TELNET AND WEB ACCESS

To disable Telnet and Web access, you can issue these commands:

```
> config services telnet-server-desired-status down
> config services web-server-desired-status down
```

CONFIGURING THE SYSTEM NAME, DATE AND TIME

You can assign a name to the AP by using this command:

```
> config system name <string>
```

The new name you assign to the system becomes effective upon reboot. You can log out and log in to see the change.

To set the system date and time, enter the following command:

```
> config system date-and-time <string>
```

where <string> is in the format: "mm/dd/yyyy hh:mm:ss:tt". Use quotation marks, as in this example:

```
> config system date-and-time "10/05/1996 22:35:00:00"
```

CONFIGURING THE WATCHDOG TIMER

The watchdog timer is a mechanism designed to monitor the health of the system. You can think of the watchdog timer as a clock that "watches" the system. When the AP is functioning properly, the timer is updated at a specified interval. The default interval is 20 seconds. If the timer is not updated within three update intervals, it times out. If this occurs, the AP automatically resets itself.

Normally, a value between 20-30 seconds is appropriate. If you want to set the update interval to some other value, use this command:

```
> config system watchdog_update <seconds>
```

To disable the watchdog timer, set the update interval to 0 to ensure that the timer never times out.

NOTE *Any change to the update interval requires a system reboot before it becomes effective.*

RESETTING THE SOFTWARE

You can use the *reset* command to reset the Access Point software.

```
> reset
```

The following command resets the software, but leaves configuration parameters intact.

```
> config system boot software-reset reset
```

This command clears the non-volatile RAM (NVRAM) configuration and resets the system to the factory defaults.

```
> config system boot software-reset resetToDefaults
```

CONFIGURING AND USING THE FILE TRANSFER TABLE

You can use the file transfer table to backup system configuration files to a remote server, or to download saved configuration files in the event that you need to restore an AP configuration. You can also use the file transfer table for other purposes, such as downloading and storing TCL scripts.

Ideally, you should set up table entries shortly after initial configuration of the AP so you can restore a configuration quickly if a problem occurs. You can configure up to 10 table entries. These might include:

- Entries that support download or backup of a configuration from one or more remote TFTP servers
- Entries that support secure upload to or download from a TFTP server
- An entry set up to restore the existing AP configuration (in NVRAM) from Flash memory
- An entry set up to download OS images, to be used in upgrading the AP

ABOUT THE FILE TRANSFER TABLE

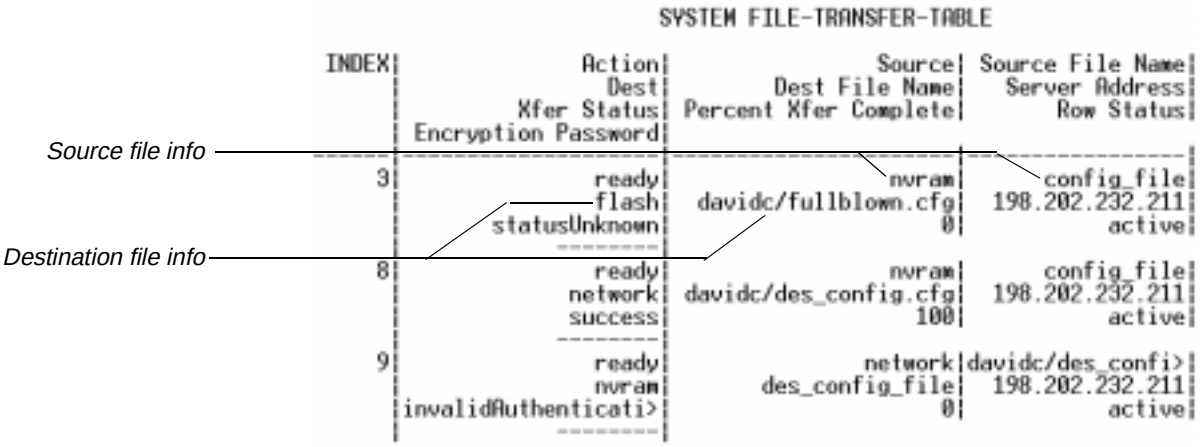
You can think of the file transfer table as a mechanism designed to facilitate file transfers. The table defines a set of entries, each of which identifies a source, a source file, a destination file, and a destination for the transfer.

To show the entries in the file transfer table, enter this command:

```
> show system file-transfer-table summary
```

Figure 8 shows a sample file transfer table.

Figure 8 Sample File Transfer Table



UNDERSTANDING TABLE ENTRIES

The file transfer table shown here contains three entries. The first entry (index number 3) specifies the transfer of a file called fullblown.cfg in NVRAM (the current configuration file) to Flash memory. The purpose of this entry is to place a copy of the working configuration file in Flash memory and to ensure that the AP configuration can be restored if the current configuration (in NVRAM) is corrupted.

A server address has been provided with the first entry, but it is unnecessary because the transfer will occur within the AP rather than via a network server. However, the next entry (index number 8) is set up to transfer a file (config_file) from NVRAM to a network location. In this case, the server location is required. Similarly, the third entry (index number 9) is set up to transfer a file to NVRAM from a network location so the server location and file name (the table needs to know which file on the network server to download) is also necessary in this case.

USING ENCRYPTION PASSWORDS

Table entries set up for secure transfer must have an associated encryption password. Note that the encryption password for each entry in Figure 8 is expressed as ---------. For security reasons, the password is never displayed at

the console.

When using the secure transfer method to upload or download a file, you must use the same encryption password when uploading or retrieving a file. Ordinarily, you assign an encryption password when setting up entries in the file transfer table so you do not have to key in the password at transfer time. For more information, see "Performing a Secure File Transfer" on Page 52.

SECURITY CONSIDERATIONS

The encryption password you provide can be 0-32 characters long. That's right: an encryption password of 0 characters will work. However, this is not advisable. You should use a password of at least 8 characters. This is simply good practice, and will make your encryption more difficult to foil.

Setting the password is another important security concern. Always set encryption passwords at the local console. Do not assign them over the network, using Telnet. Doing so transmits the password in the clear, exposing them to capture on the network.

Once you have created entries in the table and set up an encryption password for each, it is easy to securely transfer files to and from the AP. See "Performing a Secure File Transfer" on Page 52 for more information.

CREATING TABLE ENTRIES

As mentioned earlier, it is best to set up file transfer entries shortly after installing and configuring the AP. The ability to restore the AP configuration quickly and securely is basic to the administration of the device. To add entries to the file transfer table, use this command:

```
> add system file-transfer-info.<index> source <source> source-file-name
  <file name> dest dest-file-name <file name> encryption-password
  <password> server-address <address> action <action>
```

This sample command adds an entry (index number 5) set up to support secure upload of a configuration file from NVRAM to a network server.

```
> add system file-transfer-info.5 source nvram source-file-name config-file
  dest network dest-file-name des-config.cfg server-address 2.3.4.5
  encryption-password justjunk
```

Note that no action has been specified. By default, the action associated with the table is ready. This action will change when the file is securely transferred

at a later time. The command used to transfer the file is:

```
> config system file-transfer-info.5 action secureTransfer
```

The following table specifies completion values for all parameters used to set up and execute file transfers.

source	Is one of the following values: • network • flash • ram • nvram
source-file-name	Is a string (up to 127 characters) that specifies the name of the source file.
dest	Specifies the destination of the source file: • network • bootflash • flash • ram • nvram
dest-file-name	Is a string (up to 32 characters) that specifies the name of the destination file.
encryption-password	Is the password used to DES-encrypt the configuration file. Encryption and decryption operations can only occur in NVRAM, and only under two circumstances: • When a file is sent from NVRAM to Flash memory or a network location • When a file is downloaded to NVRAM from a network location or Flash Your password can be 0-32 characters in length. It is strongly recommended that you use an 8-character password, at minimum. It is also recommended that you only set the encryption password at the local console, or by using SNMPv3 sets with encryption enabled. Setting the password remotely without encryption — via Telnet, for instance — passes it over the network in the clear for anyone wishing to sniff it.
server-address	Is the IP address of the server to or from which the configuration file is to be transferred.

action	Specifies the action to take at execution time. Choices are: • ready • execute • executeAndCreateBootRecord • abort • secureTransfer The default action at setup time is ready.

DISPLAYING FILE TRANSFER TABLE INFORMATION

To display the System File Transfer Table, use this command:

```
> show system file-transfer-table
```

The table provides information on all entries. To display information for a particular entry in the table, specify the entry's index using this command:

```
> show system file-transfer-info.<index number>
```

A sample command and display of information associated with a particular entry (index number 2) is shown here.

```
> show system file-transfer-info.2
```

```
                        SYSTEM FILE-TRANSFER-INFO.2

           Source* = network                    Xfer Status   = 0
Source File Name* = V1.5.0R.2.bin Percent Xfer Complete   = 13
             Dest* = flash                         Action*  = execute
   Dest File Name* = boot_file2                  Row Status* = active
   Server Address* = 10.0.0.1
```

TRANSFERRING FILES

As mentioned earlier, the AP supports two types of file transfers. These are:

- Secure transfers, using DES encryption to create a secure form of TFTP
- Ordinary transfers that send all data to or from the AP without encryption

It is wise to use the AP's secure transfer capability when transferring configuration files. You can use the ordinary, unsecured method of transfer for TCL scripts and OS images.

A Note On Basic TFTP Security

When transferring a file from the AP to a network server, keep in mind that the destination file name must already exist on that server, with read/write access privileges. Otherwise, you will not be able to write the file to the server. This is a basic feature of TFTP security.

Performing a Secure File Transfer

To perform a secure configuration file transfer to or from the AP, you must create an entry with an encryption password, or use a NULL password. Using an encryption password is more secure, and is therefore the recommended method for encrypting sensitive information. See "Using a NULL Password" on Page 53 for information about using a NULL password.

To transfer a configuration file to or from the AP, perform these steps:

- Provide an encryption password (or use the NULL password) when setting up an entry in the file transfer table.
- Choose secureTransfer as the action (rather than execute).

Here is a sample command that sets up a file for secure transfer later on.

```
> add system file-transfer-info.5 source nvram source-file-name config-file
  dest network dest-file-name des_config.cfg server-address 2.3.4.5
  encryption-password justjunk
```

As an added security measure, the encryption password provided when setting up a file for transfer is not displayed at the console.

Here is the command required to transfer the file config-file securely. Note that an action of secureTransfer is used, rather than execute.

```
> config system file-transfer-info.5 action secureTransfer
```

Note also that an encryption password has not been provided within this command string. Providing the password is unnecessary because it was already provided when setting up the transfer associated with index number 5.

A Note on Password Agreement

When securely retrieving a file from a network location, it is vital that the encryption passwords used to securely transfer and retrieve the file match exactly. Otherwise, the transfer will not succeed. In this case, the error mes-

sage invalidAuthentication shows up in the Xfer Status field of the file transfer table. An example of this is shown in Figure 8.

USING A NULL PASSWORD

You can perform a secure file transfer using a NULL password by providing no password as the value for the encryption-password parameter. If you do this, an internally-generated password is used to effect the transfer, and both DES encryption and MD5 checksums are used to transfer the file. However, this method is not recommended. Providing a password of at least 8 characters is strongly preferred, as it raises the barrier for those wishing to decrypt a message.

EXECUTING AN ORDINARY FILE TRANSFER ACTION

Once you set up entries in the file transfer table, you can execute any entry with a simple execute command. For example, to transfer a file transfer with an index of 3, you would issue this command:

```
> config system file-transfer-info.3 action execute
```

You should always verify the file transfer before performing it. Do this by issuing a *show system file-transfer-table* command as explained previously. This is a good way to avoid transferring the wrong file.

Keep in mind that you cannot securely transfer a file (via the secureTransfer action) unless you have assigned an encryption password to it. If you attempt to do this, the transfer fails. A message of incompatibleFile appears in the Xfer Status field in the file transfer table.

INTERPRETING ERRORS

Attempts to transfer files can fail for a number of reasons. When this occurs, an error message appears in the Xfer Status field of the file transfer table. Once you identify the cause of a failure, correcting the problem is usually a simple matter. Here are the errors that can appear in the file transfer table, together with their meanings:

- invalidAuthenticationHash

  If this message appears, you might have provided the wrong encryption password for the file when attempting to transfer it. This message will also appear if the file has been corrupted or modified. Any of these issues cause problems because MD5 produces an authentication on the inner portion of

the decrypted file, and on the entire file. The authentication hash must agree on both encryption and decryption operations, or the transfer will not succeed.

- incompatibleFile

  This message appears if you try to transfer a file encrypted with security information using an action of execute rather than secureTransfer. Another cause is an attempt to securely transfer a file that has not been encrypted.

- incompatibleSecureParams

  This message appears if you try to securely transfer a file to the AP, but the file identified within the transfer does not contain valid security information.

- InvalidDecryptionResult

  This error appears if the encryption algorithm failed to produce the expected file type or format when run on the file transferred. This can be caused by a number of factors, including file corruption or parameter mismatch.

SAVING THE CONFIGURATION

You can back up the configuration by performing a TFTP file transfer from NVRAM to a network server. Follow these steps:

1 Identify the current configuration file in NVRAM using this command:

```
> show system storage-area.nvram files summary
```

Your display might resemble the following:

```
                    SYSTEM STORAGE-AREA.NVRAM FILES SUMMARY

STORAGE AREA|              NAME|         File Type|           Source|
            |                  |      Source Name|    Source Server|
            |                  |             Date|        File Size|
            |                  |         Running?|                 |
------------|------------------|------------------|-----------------|
       nvram|       config_file|            config|              ram|
            |                  |  not applicable|  0.0.0.0|
            |                  |  not applicable|   130943|
            |                  |            false|                 |
```

Look for a file type of "config" and note the name (for example, config_file) and file size (for example, 130943).

2 Configure the file transfer record to back up the configuration file using this command:

```
> add system file-transfer-info.<instance> source nvram dest network
  source-file-name <config_file_name> dest-file-name
  <dest_config_file_name> server-address <ip address>
```

Note that the server-address parameter is the IP address of the TFTP server on which you will store the backup copy. For example:

```
> add system file-transfer-info.1 source nvram dest network source-file-
  name config_file dest-file-name my_config_file server-address
  198.202.232.20
```

3 Execute the file transfer set up in step 2 with this command:

```
> config system file-transfer-info.1 action execute
```

4 You can observe the progress of the file transfer by issuing the following command:

```
> show system file-transfer-info.1 summary
```

Once the transfer is complete, verify that the file size on the TFTP server matches the size noted in step 1.

CONFIGURING BOOT ORDER

To show a summary of bootable images:

```
> show system boot boot-images summary
```

The resulting table provides an index for each image.

To configure boot priority:

```
> config system boot boot-image.<index> boot-priority.<number>
```

where the number specifies the priority, with 1 indicating the highest priority.

DISPLAYING A CONFIGURATION SUMMARY

You can use the *dump-config* or *status* command to display a configuration summary for different components.

DISPLAYING CONFIGURATION CHANGES

The *dump-config* command displays a formatted summary of the configuration for the entire router or for a selected component. The summary lets you check all or only specific parameters of an operational configuration.

The *dump-config* syntax is:

```
> dump-config [<component>] [defaults|no-defaults]
```

where <component> is one of the following: all, atm, atm-ds3, atm-sonet, cbq, bridge, dns, eth, frame-relay, hssi, ip, ipoa, ppp, system, t1, t3.

Both command arguments are optional, and the default values are *all* and *defaults*. When you use the *no-defaults* argument, the summary lists only the component's parameters that a user has previously modified.

The following is a sample configuration summary for a bridge component:

```
> dump-config bridge
Dumping bridge configuration parameters INCLUDING defaults...

bridge transparent-bridge
  Aging Time = 10 seconds Backup Mode = disabled
  bridge.1
  MTU = 1500 bytes Promiscuous Mode = false Mode = up Alias = ""Up Down
  Trap = enabled
  bridge.* arp-cache-entry.*
  <None>
  bridge.1 arp-config
  Arp Cache Timeout = 20 minutes
```

DISPLAYING CONFIGURATION STATUS

The *status* command displays the status for the specified component and lists its:

- Configuration parameters
- State parameters
- Counters

The syntax for this command is:

```
> status [<component>]
```

where <component> can be: all, atm, atm-ds3, atm-sonet, bridge, cbq, dns, eth, frame-relay, hssi, ip, ipoa, ppp, system, t1, or t3. If you do not specify a component, the display contains the summary of the system's layer status.

The following is a sample configuration summary for a frame-relay component:

```
> status frame-relay
  frame-relay.1
  Mode = up, State = down, Last State Change = 280 (0 hours 0 min 2 secs),
  Bytes In = 0, Discards In = 0, Errors In = 0, Bytes Out = 0, Discards Out = 0,
  Multicast = broadcast
  frame-relay.1 dllci.200
  State = inactive, Rx FECNs = 0, Rx BECNs = 0, Tx Frames = 0, Tx Octets =
  0, Rx Frames = 0, Rx Octets = 0, Creation Time = 285 (0 hours 0  min 2
  secs), Last Time Change = 285 (0 hours 0 min 2 secs)
  frame-relay.* arp-cache-entry.* <None>
```

NVRAM COMPRESSION

The AP compresses NVRAM to create additional free space when NVRAM begins to fill. Compression occurs automatically when NVRAM is approximately 80% full. No user configuration or subsequent action is required to enable this feature.

COMPRESSION AND FULLBOOT VERSION

Compression is only supported for fullboot versions 3.0 and above. The AP checks its fullboot version prior to compressing NVRAM. The fullboot version is displayed on the AP console during system boot.

DISPLAYING COMPRESSION STATE

Use this command to display whether compression is active or inactive:

```
> show system nvram-configuration-file

                    SYSTEM NVRAM-CONFIGURATION-FILE

          Last Write  = 09/09/1999 15:53:32:00
  Last Write Source   = 0.0.0.0
    Num Config Recs   = 328
       Percent Full   = 3
        Compression   = inactive
```

SAVING COMMAND OUTPUT IN A FILE

The *log* command lets you save the output from commands in a file. The *log* syntax is as follows:

```
> log <command> [ <command-specific argument> ]
```

where *command* is:

Command	Description
open <server address> <filename>	Opens the log. The <filename> argument must use a .tcl file extension, exist in the tftpboot directory of the TFTP server whose address is the <server address> argument, and be writeable.
close [nosave]	Closes the log. If you use the nosave argument, logging stops without writing the output to a file.
status	Displays the current status of logging.
dump	Displays the current contents of the log to the screen.

The following sample commands would save the output of the *gen-script* command to the test.tcl file:

```
> log open 127.0.0.1 maria/test.tcl
> gen-script all defaults
> log close
```

CUSTOMIZING A PRE-LOGIN BANNER

You can use the pre-login banner to display information immediately below the generic system description from the CLI Welcome page. You can display a variety of messages. Messages can be up to 255 bytes long.

Simple format characters such as \n (new line char) and \t (tab char) can be used to control the output of displayed text. Banners can be a single line, multiline, or multiline with text border.

CONTROL CHARACTER RESTRICTIONS

Different machines (terminals, Telnet sessions) do things differently regarding format characters, so "simple is better". In some cases, a \n (new line char) is sufficient to advance lines, in other cases a return character and a new line char is needed.

And depending on the previous state of the Telnet terminal session, or how the terminal is setup, the \t (tab char) does not always produce the expected indentation. It is beyond the scope of this particular subject to make sure that the terminal is set up accordingly. But, in most cases, it is not a problem and the expected indents do happen.

CONFIGURING A SIMPLE MESSAGE

To configure a simple, one-line message, proceed as follows:

```
> config system pre-login-banner "Hi, you are currently connected to an
  Access Point system"
```

This produces the following output on the login screen:

```
AccessPoint (hw: 1.7; sw: V2.3.0.R1 8/14/2000 18:12:09:00)

  Hi, you are currently connected to an Access Point system
```

CONFIGURING A FORMATTED MESSAGE

To configure a formatted message using vertical spaces, use the \n characters along with text. A sample is shown here.

```
> config system pre-login-banner "Hi, \n\n you are currently connected to an
  Access Point"

  Hi,

  you are currently connected to an Access Point
```

CONFIGURING A MULTILINE FULL TEXT BORDER MESSAGE

To configure a bordered message with text using the sample template shown here, proceed as follows.

```
> config system pre-login-banner
  "\n\t++++++++++++++++++++++++++++\n\t+      AP \
  Pre-Login-banner  +\n\t+                    +\n\t+  text-message \
  +\n\t+                  +\n\t+  contact info...       +\n\t+ \
  +\n\t++++++++++++++++++++++++++++"
```

This produces the following message:

```
++++++++++++++++++++++++++++
+      AP Pre-Login-banner  +
+                           +
+  text-message             +
+                           +
+  contact info...          +
+                           +
++++++++++++++++++++++++++++
```

CONFIGURING THE DNS CLIENT

You can configure the AP's DNS client to talk to a list of DNS servers. To do this, provide the IP address of each DNS server in dotted decimal format, and assign each name server a level of preference (1 is highest) as illustrated by these commands:

```
> config services dns nameserver.1.2.3.5 preference 1
> config services dns nameserver.1.2.3.7 preference 2
> config services dns nameserver.1.2.3.9 preference 2
```

CONFIGURING YOUR SITE'S DOMAIN NAME

You should also configure your site's domain name. Doing this will allow you to use ping, telnet, or traceroute for internal hosts without specifying the host's fully qualified domain name.

To specify your domain name, proceed as shown here:

```
> config services dns domain ABCplastics.com
```

USING OTHER DNS COMMANDS

These commands are used to manage the AP's native DNS client:

- *config dns admin-status*
- *config dns cache-status*
- *config dns configuration-reset*

The following table explains the use of these commands:

Command	Completions
config dns admin-status	• up Up is the default state, and indicates the DNS client is running. • down Turns off the DNS client.
config dns cache-status	• enabled Enabled is the default state, and indicates that the cache is being written to and read by the DNS client. • disabled Causes the client to stop reading the cache. Resolution will be handled by requests sent to the designated DNS name server. Though unread, information continues to be added to the cache. • clear Clears the DNS cache.
config dns configuration-reset	• running The default state, indicates the DNS client is running. • reset The reset command flushes the cache and resets all counters. Non-volatile information (including designated name servers) are preserved. • other not applicable • initializing not applicable

GETTING INFORMATION ON THE DNS CLIENT

The *show* command has DNS-specific parameters that provide detailed information on the following:

- cache
- counters
- nameservers and nameserver.<instance>
- summary

Examples of each command are provided below.

SHOW DNS SUMMARY

> show services dns summary

```
                                SERVICES DNS SUMMARY

Implementation Description  = Xedia Corp Domain Name Service (DNS) Resolver.

                              rsion 1.0.
                    Domain* = abcplastics.com
                   Service  = recursiveOnly
     Time Since Last Boot   = 22053 seconds
    Time Since Last Reset   = 22053 seconds
      Configuration Reset*  = running
         Max Time To Live*  = 72000 seconds
              Admin Status* = up
       Operational Status   = noNameServersDefined
             Cache Status*  = enabled
```

SHOW DNS COUNTERS

> show services dns counters

```
                                SERVICES DNS COUNTERS

                                          Queries Sent  = 0
                                Query Responses Received  = 0
                            Responses Indicating No Error  = 0
    Responses Indicating Illegally Formatted Queries  = 0
                  Responses Indicating The Server Failed  = 0
          Responses Indicating The Name Does Not Exist  = 0
           Responses Indicating Lack Of Server Support  = 0
    Responses Indicating The Server Refused To Answer  = 0
                                Unparsed Responses  = 0
                                    Retransmisions  = 0
                                      No Responses  = 0
                                    Records Cached  = 0
                                Records Not Cached  = 0
```

SHOW DNS NAME SERVERS

> show services dns nameservers

SERVICES DNS NAMESERVERS

NAME SERVER ADDRESS	Preference	Row Status
1.2.3.5	1	notInServi>

SHOW DNS CACHE

> show services dns cache

SERVICES DNS CACHE

NAME	CLASS	TYPE	INDEX	Time To Live Elapsed Time Source Data(RR)
www.altavista.com	internet	ipV4	0	00 02:00:00.00 00 00:58:35.00 198.202.232.207 204.74.103.37
www.altavista.com	internet	ipV4	1	00 02:00:00.00 00 00:58:35.00 198.202.232.207 204.123.2.66
www.altavista.com	internet	ipV4	2	00 02:00:00.00 00 00:58:35.00 198.202.232.207 204.123.2.69

USING THE SYSTEM SCHEDULER

This section explains how to configure the AP's Scheduler function, and covers:

- Purpose and environment
- Using the Scheduler parameters
- Summarizing the Scheduler status and batch jobs
- Understanding Scheduler passes

PURPOSE AND ENVIRONMENT

The purpose of the Scheduler is to run batch jobs on a one-time, irregular, or regular basis. Jobs can run on a specific day, week, month of the year, even at a specific time. You can combine wild cards with specific data to set up regular run times.

The batch jobs can contain either CLI command strings, TCL scripts, or any combination of the two. By using the autosourcing option of scripts, any commands or procedures defined in the *.tclinit.tcl* and *.scheduler.tcl* scripts stored in NVRAM or flash memory on the local node are available in the Scheduler's environment. Refer to "Using TCL Scripts" on Page 70 for more information.

USING THE SCHEDULER PARAMETERS

The top-level Scheduler parameters are:

clear-log	Clears all batch job log messages. Accepted values are ready or clear. The default value is ready.
logging	Enables/disables logging of batch entries. The default value is disabled.
mode	Turns the Scheduler on or off. The default value is enabled.
sleep-interval	Sets interval (in minutes) between Scheduler passes. The default value is 5 minutes.

The batch-job.<inst> subcomponent is used to create or modify up to 10 batch jobs with the following parameters:

month	Sets the month the Scheduler is to run the batch job. Values are: jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec.
day-of-month	Sets the date the Scheduler is to run the batch job. Values are: any or d01 - d31.
day-of-week	Sets the day of the week the Scheduler is to run the batch job. Values are: any, mon, tue, wed, thur, fri, sat, sun.
hour	Sets the hour the Scheduler is to run the batch job. Values are: any or h00 - h23.
minute	Sets the minute the Scheduler is to run the batch job. Values are: any or m00 - m59.
exec-command	Executes the CLI command (can be multiple commands if separated by a ";") or sources the script files. This command line can be a text string of up to 255 characters.
row-status	Decides the status of the batch, whether: active, notInService, notReady, createAndGo, createAndWait, Destroy.

You do not need to specify these parameters in any order on the command line. However, you must specify enough parameters so that the Scheduler can execute the batch job. For example, you can specify the month and day-of-month parameters and not specify the day-of-week parameter, but you should always specify the hour, minute, and exec-command parameters.

EXAMPLE CONFIGURATION

The following are CLI commands for a simple Scheduler batch job configuration.

```
> config system scheduler mode enabled

> config system scheduler logging enabled

> config system scheduler sleep-interval 5

> config system scheduler batch-job.1 exec-command setup month any day-
  of-month d01+d15+d30 hour h01+h12+h23 minute m00 row-status active
```

NOTE *You must reboot the node for any change to the mode parameter to take effect.*

This configuration means that batch job 1 runs the *setup* TCL procedure on the first, fifteenth, and thirtieth days of each month at 1:00, 12:00 and 23:00. The Scheduler wakes up every five minutes (default) and makes a pass through the configuration. The row-status value must be "active" before the Scheduler pays attention to that batch job. You can configure a batch job but not use it immediately by assigning the row-status value of "notInService".

Refer to "Using TCL Scripts" on Page 70 for more information about configuring TCL scripts.

SUMMARIZING THE SCHEDULER STATUS AND BATCH JOBS

To view the status of the Scheduler, enter:

```
> show system scheduler summary
```

The following is an example Scheduler summary:

```
SYSTEM SCHEDULER SUMMARY
  Admin Status* = enabled
  Oper Status = up
  Sleep Interval* = 5
  Logging Admin* = enabled
  Current Time = 10/31/1998 Saturday(7) 14:54
  Scheduler Last Pass = 10/31/1998 Saturday(7) 14:45
  Scheduler Next Pass = 10/31/1998 Saturday(7) 14:50
```

You can schedule up to 10 batch jobs at one time. The exec-command parameter can execute single or multiple commands or source a script containing many commands. You can list the summary of each individual batch job and list all the batch jobs.

To view the summary of a batch job, enter:

```
> show system scheduler batch-job.<index> summary
```

The following is an example Scheduler summary:

```
SYSTEM SCHEDULER BATCH-JOB.X SUMMARY
  Date And Time = **/01+15+30/** --- 01+12+23:00:**
  Log Message = ""
  Exec Command* = setup
  Schedule Status = waiting
  Row Status* = active
```

The summary format of the Date And Time field is mm/dd/yy DayofWeek hh:mm:ss. Wildcard values are represented as asterisks (**). The plus sign (+) indicates multiple values, such as when the Scheduler runs the batch job at two different times of the same day. The hyphens (--) indicate there is no value specified for a field.

UNDERSTANDING SCHEDULER PASSES

The Scheduler lists and schedules all the scripts before running them. It continually checks the list of batch jobs to run and "sleeps" in between check passes. The sleep interval decides how long the Scheduler sleeps.

If there are more jobs to run than the Scheduler has time to run, it does not queue multiple executions of a job. Sometimes the sleep interval is longer than the interval in between identical batch jobs, therefore instead of waking up to a queue of 10 batch jobs to run, it drops the identical entries on the list.

If you reset the date or time of the AP, the Scheduler adjusts accordingly. If you reset the time to before the next scheduled pass, upon wakeup the Scheduler still runs the jobs that it had scheduled to execute on that next pass. If you reset the clock to ahead of the next scheduled pass, the Scheduler only runs the jobs it had scheduled for within two minutes of the last scheduled pass, and not all the jobs that it had scheduled between the last scheduled pass and the new time.

USING TCL SCRIPTS

A TCL script contains several TCL or CLI commands which only take effect in the operational CLI or command interpreter. This section covers:

- Creating TCL Scripts
- Moving TCL Scripts to the Access Point System
- Running Scripts
- Autosourcing Scripts
- Viewing TCL Scripts Available on the Access Point

CREATING TCL SCRIPTS

You can create a TCL script using any text editor, but you must save it with the extension `.tcl`. It is highly recommended that you place a standard header of "`# xedia-tcl-script`" at the top of your script. You can store the scripts either locally in the flash memory or NVRAM of the Access Point box, or remotely on a server.

TCL scripts can contain a combination of CLI commands and source commands that reference local or remote files.

MOVING TCL SCRIPTS TO THE ACCESS POINT SYSTEM

Once you have created your TCL files in a text editor you have to move them to the AP. You can transfer TCL files to or from the NVRAM or flash memory using the standard file transfer table. The transfers will be initiated using the standard file transfer table. TFTP protocol will be used to transfer files across the network, therefore, a tftp server must be available from the AP. See the "Configuring and Using the File Transfer Table" on Page 47 section for more information about transferring files.

RUNNING SCRIPTS

You can run a TCL script either manually from the CLI command line, or automatically using the Scheduler (see the previous section).

To run a TCL script from a command line, enter:

```
> source <src> <fileName>
```

where <src> is either flash, NVRAM, or a server on the network and <fileName> is the name of the script as shown in the following example:

```
> source server.198.195.192.165 setup.tcl
```

AUTOSOURCING SCRIPTS

Upon logging into a CLI session, up to four TCL script files can run automatically if they exist in the NVRAM or flash memory of the local AP. The command interpreter of the AP tries to access the following files in this order:

```
nvram/.tclinit.tcl

flash/.tclinit.tcl

nvram/.<username>.tcl

flash/.<username>.tcl
```

If you want specific commands to run upon login, you should put the necessary scripts in these files.

VIEWING TCL SCRIPTS AVAILABLE ON THE ACCESS POINT

You can view the list of available TCL scripts using the standard system storage-area table. From the command line, enter:

```
> show system storage-area.* files summary
```

SAMPLE TCL SCRIPT

The following is `setup.tcl`, an example TCL script. This script automatically configures CBQ classes which you could use to set up CBQ on several different Access Point systems.

```
# xedia-tcl-script
proc setup {} {

puts "
  Configuring Ethernet for 100mbps full duplex...
  "
config eth.1 duplex-mode full config-speed mbps100
puts "
  Configuring IP address and default route..
  "
config ip.1 address.10.4.94.100 net-mask 255.255.255.0
config ip static-route.0.0.0.0 mask.0.0.0.0 tos.0 next-hop.10.4.94.101 row-status active
puts "
  Building CBQ.1 tree....
  "
add cbq.1 traffic-class.root parent interface bandwidth-allocation 10000000
add cbq.1 traffic-class.jwlong parent root bandwidth-allocation 128000
add cbq.1 traffic-class.jwlongsweb parent jwlong bandwidth-allocation 0 src-ip-start 204.15.140.10 src-ip-end 204.15.140.550
add cbq.1 traffic-class.FrameRelay7 parent root bandwidth-allocation 1544000
add cbq.1 traffic-class.fcaclass parent FrameRelay7 bandwidth-allocation 768000 src-ip-start 207.16.11.128 src-ip-end 207.16.11.143
add cbq.1 traffic-class.capital parent FrameRelay7 bandwidth-allocation 256000 src-ip-start 208.53.44.92 src-ip-end 208.53.44.355
add cbq.1 traffic-class.location parent root bandwidth-allocation 512000
add cbq.1 traffic-class.defaultclass-default parent root bandwidth-allocation 0
puts "
  Done.
  "
}
```

MANAGING SYSTEM EVENTS

The AP logs events in its event log table. You can display these events using the *show* command. As events occur, you can also display them on the console or send them as SNMP traps.

For most situations, the default settings are appropriate. However, sometimes you might want to filter particular events so that system resources are not used up to log events that do not affect normal operations. You also use filtering to sort the type of event information that is logged.

You can filter events for input so that the specified events are not logged in the event log table. You can filter events for output so that they are not retrieved from the log table for display on the console or for sending as SNMP traps. You can set input and output filtering options based on event level, subsystem, and specific event ID. Output filtering can also be based upon date and time.

The rest of this section explains how to perform the following tasks:

- Displaying the System Event Log table.
- Setting filtering options by level.
- Setting output filtering options by time.
- Displaying filtering options.
- Setting filtering options by subsystem.
- Displaying subsystem filtering options.
- Configuring event parameters.
- Displaying event parameter settings.

"Sample Event Filtering Configuration" on Page 84 shows how you might use the event filtering options.

DISPLAYING THE SYSTEM EVENT LOG TABLE

To display the System Event Log table, use this command:

```
> show system event log
```

The log table provides information about the index number, date, level, message, subsystem, and ID for the events. If you wished to display the log table

for a particular event or series of events, you would use this command:

```
> show system event log-entry.<index number>
```

The following example displays the log table for the log entries for index numbers prefaced by 26.

```
> show system event log-entry.26*
```

```
                                SYSTEM EVENT LOG-ENTRY.26*

INDEX|                                                    Date|  SubSys|
     |                                                   Level|     Id|
     |                                                     Msg|       |
-----|------------------------------------------------------|-------|
  260|                                  09/22/1998 09:43:33:00|    AVC|
     |                                                    info|      5|
     |                                              atm.1 - up|       |
  261|                                  09/22/1998 09:43:33:00|   ADRV|
     |                                                    info|      9|
     |          bt8233-sar-driver.0.0 - opened AAL 5 VCC 0.40|       |
  262|                                  09/22/1998 09:43:33:00|    AVC|
     |                                                    info|      9|
     |                                        atm.1 - VC 0.40 Up|       |
  263|                                  09/22/1998 09:43:33:00|   IPOA|
     |                                                    info|     16|
```

SETTING FILTERING OPTIONS BY LEVEL

The most common change that you would make to default filtering options would be to set different input filtering options by level. Events are categorized into five levels:

- debug
- trace
- info
- warn
- fatal

By default, input filtering for the debug and trace levels is filtered (that is, not entered in the log table) because these events provide information that you would only need for special situations such as problem solving. The default option for output filtering is no filtering for any levels. This setting ensures that all events included in the log table are displayed. These default values ensure the display of events for the info, warn, and fatal levels that provide important information about state changes and errors that affect the operation of APs.

SETTING THE INPUT FILTERING OPTIONS

To set the input filtering options so that input event types for certain levels are not logged to the log table, use this command:

```
> config system event in-filter-level [debug | trace | info | !debug | !trace | !info]
```

By default, debug and trace events are filtered. However, if you wanted to filter info events so that they are not logged, you would use this command:

```
> config system event in-filter-level info
```

If you wanted to turn off filtering for a particular event level, you would use the value preceded by an exclamation point (!). For example, to turn off filtering for debug events so that they would be entered in the log table, use this command:

```
> config system event in-filter-level !debug
```

SETTING THE OUTPUT FILTERING OPTIONS

To set the output filtering options so that certain output event types can be retrieved from the log table, use this command:

```
> config system event out-filter-level [ debug | trace | info | warn | fatal | !debug
  | !trace | !info | !warn | !fatal ]
```

By default, no output event levels are filtered. However, if you wanted to filter info events so that they are not logged, you would use this command:

```
> config system event out-filter-level info
```

If you wanted to turn off filtering for a particular event level, you would use the value preceded by an exclamation point (!). For example, to turn off filtering for debug events so that they would be retrieved from the log table, use this command:

```
> config system event out-filter-level !debug
```

SETTING OUTPUT FILTERING OPTIONS BY TIME

You can filter output events so that only events on or after the set date and time are logged to the System Event Log table. To set the output filtering options based on date and time, use this command:

```
> config system event out-filter-date-and-time <mm/dd/yyyy hh:mm:ss:tt>
```

By default, no date or time is set for this option.

DISPLAYING THE FILTERING OPTIONS

To display the filtering options that you have set for the input filter, the output filter, and the output filter date and time, use this command:

```
> show system event global-filter
```

```
                              SYSTEM EVENT GLOBAL-FILTER

          In Filter Level* = debug*trace
         Out Filter Level* = ""
Out Filter Date And Time* = 00/00/0000 00:00:00:00
```

SETTING FILTERING OPTIONS BY SUBSYSTEM

You might want to configure filtering options for particular subsystems and their events. By default, subsystems do not filter events.

SETTING SUBSYSTEM FILTERING OPTIONS

To configure how the subsystem filters events, use this command:

```
> config system event sub-system-filter.<subsystem> filter [ none | input | out-
  put | inputOutput ]
```

None is the default value, input indicates input filtering, output indicates output filtering, and inputOutput indicates both input and output filtering. You can use the asterisk (*) to indicate all subsystems.

SETTING A SUBSYSTEM EVENT AND ITS FILTERING OPTIONS

To configure how the subsystem event filters events for all instances of the SNMP subsystem, use the following command:

```
> config system event sub-system-filter.SNMP id-filter.* \
  filter [ none | input | output | inputOutput ]
```

Using the wildcard character for the id-filter parameter sets the filter type for all instances of the SNMP subsystem. To configure a particular instance, specify the event ID number for the id-filter parameter as shown in the following example:

```
> config system event sub-system-filter.SNMP id-filter.7 filter input
```

DISPLAYING SUBSYSTEM OPTIONS

You can display the configured filtering options for subsystem and subsystem events.

DISPLAYING SUBSYSTEMS AND THEIR FILTERING OPTIONS

To display a list of all the subsystems and their filtering options (input, output, both, or none), use this command:

```
> show system event sub-system-filters summary
```

DISPLAYING SUBSYSTEM EVENTS AND THEIR FILTERING OPTIONS

When you want to display event IDs, event names, filtering options, and event messages for a particular subsystem, use this command:

```
> show system event sub-system-filter.<subsystem> id-filter.<event ID>
```

You can use the asterisk (*) to display all the subsystems and their event IDs.

DISPLAYING A SUBSYSTEM AND ITS FILTERING OPTIONS

To display a particular subsystem, use the following values for the appropriate subsystem:

Subsystem	Value
ATM Driver	ADRV
Address Resolution Protocol	ARP
ATM VC	AVC
Authenticator	Auth
Autoconfiguration	AutoC
Border Gateway Protocol 4	BGP
Bridging	BR
IP Backup	BkUp
Class-Based Queuing	CBQ
Configuration Manager	CFG
Client Groups	CG

Subsystem	Value
CRON Job Manager	CRON
DHCP Relay	DHCP
Data Link	DL
Domain Name Service	DNS
DS1 Driver	DS1
DS3 Driver	DS3
Distance Vector Multicast Routing Protocol	DVMRP
Dial Backup	DialB
Ethernet Data Link	ETHD
Ethernet Driver	Edrv
Event Manager	EVT
Frame Relay	FR
Gigabit Ethernet	GigEth
HTTP Server	HTPD
HSSI Driver	Hdrv
Internet Control Message Protocol	ICMP
Internet Key Exchange Protocol	IKE
IP Forwarding Policy	IPFP
IP over ATM	IPOA
IP Security Protocol	IPSC
IP	IPv4
Integrated Services Digital Network	ISDN
Intermediate System-to-Intermediate System	ISIS
Layer 2 Tunneling Protocol Dial layer	L2D
Layer 2 Tunneling Protocol	L2TP
Layer 2 Tunneling Protocol Tunnel	L2TPT
LMC5245 DS3 Driver	L3dr

Subsystem	Value
Layer Management	LM
Loopback interface	LPBK
LAN Media 5200 Driver	LmDr
Multicast	MCST
Medium Speed Synchronous Interface	MSSI
Locally-administered IP addresses	MYIP
Network Address Translator	NAT
Network Time Protocol	NTP
OSPF Version 2	OSPF
Public Key Infrastructure	PKI
Policy Routing	PLCY
Point to Point Protocol	PPP
Multilink PPP	PPPMP
PCI Driver	Pdrv
Reboot	REBO
RIP Version 2	RIP2
Remote Monitor	RMON
Routing	RTE
Radius Client	RadC
Sink interface	SINK
Switched Multimegabit Data Service	SMDS
SNMP Manager	SMGR
SNMP Agent	SNMP
System	SYS
SLA Monitor	SlaM
Serial Port	SrlP
SSH Client	SshC

Subsystem	Value
SSH Server	SshD
SSH Library	SshL
TCL Manager	TCL
Traffic Manager	TM
IP Tunnel	TNL
Telnet	TelN
User Datagram Protocol	UDP
User Interface	UI
Xedia System	XSYS

For example, if you wanted to display the filter types for all instances of the SNMP subsystem, use the following command:

```
> show system event sub-system-filter.SNMP id-filter.*
```

```
                    SYSTEM EVENT SUB-SYSTEM-FILTER.SNMP ID-FILTER.*

 SUBSYSTEM NAME| EVENT ID|       Event Name| Filter|    Event Message|
---------------|---------|-----------------|-------|-----------------|
          SNMP|        0|TraceTrapExecuti>|   none|                --|
          SNMP|        1|TraceErrorHandli>|   none|                --|
          SNMP|        2|TraceExceptionHa>|   none|                --|
          SNMP|        3|TraceNormalExecu>|   none|                --|
          SNMP|        4|TraceExcessiveEx>|   none|                --|
          SNMP|        5|          TraceOID|   none|                --|
          SNMP|        6|      TracePacket|   none|                --|
          SNMP|        7|      Initialized|   none|The SNMP Agent i>|
          SNMP|        8|ResetAdminPasswo>|   none|The SNMP Agent i>|
          SNMP|        9|UnAuthorizedComm>|   none|The SNMP Agent r>|

One or more values were truncated (last char '>').
Use 'show' with complete row identifier for "id-filter.<inst>"
```

Using the wildcard character for the id-filter parameter displays the filter types for all instances of the SNMP subsystem. To display a particular instance, specify the event ID number for the id-filter parameter. The event name, filter, and event message for that particular instance is displayed as shown in the following example:

```
> show system event sub-system-filter.SNMP id-filter.7

                    SYSTEM EVENT SUB-SYSTEM-FILTER.SNMP ID-FILTER.7

   Event Name  = Initialized
       Filter* = none
 Event Message = The SNMP Agent is Up and Running.
```

CONFIGURING EVENT PARAMETERS

This section describes how to set the maximum number of logged events for non-persistent and persistent event logs, clear the system event log table (clear-all-events), send events as SNMP traps (send-events-as-traps), send events to remote syslog servers (send-events-as-syslog-messages), and display events on the console (display-events).

SETTING THE MAXIMUM NON-PERSISTENT EVENT LOG SIZE

The maximum number of events that can be stored in RAM (non-persistent) is in the range of 400 to 1000 events. To set the maximum non-persistent event log size to 1000:

```
> config system event max-log-size 1000
```

Note that changing the non-persistent event log size requires a system reboot to take effect.

SETTING THE MAXIMUM PERSISTENT EVENT LOG SIZE

The maximum number of events that can be stored in NVRAM (persistent) is in the range of 20 to 1000 events. To set the maximum persistent event log size to 1000:

```
> config system event max-nvram-log-size 1000
```

Note that events stored in NVRAM take up approximately 300 bytes of NVRAM capacity each. Therefore, users should balance configuration storage requirements with the desire to store events over a system reboot.

CLEARING THE SYSTEM EVENT LOG TABLE

To clear the event log table, use this command:

```
> config system event clear-all-events execute
```

SENDING EVENTS AS SNMP TRAPS

By default, events are not sent as SNMP traps (value set to false). When you want to send events as SNMP traps, use this command:

```
> config system event send-events-as-traps true
```

You must configure the SNMP trap destinations if you want to send events as SNMP traps.

SENDING EVENTS TO SYSLOG SERVERS

The AP can send event log messages to remote syslog servers over the syslog protocol. You can specify up to five remote servers. All syslog messages are sent over the standard syslog port, UDP 514.

To enable syslog distribution of event messages:

```
> config system event send-events-as-syslog-messages true
```

To specify the remote syslog server with an address (for example, 19.20.21.1):

```
> config system event syslog-remote-server 19.20.21.1
```

To specify multiple remote servers, separate the addresses with commas:

```
> config system event syslog-remote-server 19.20.21.1,19.20.21.2
```

Note that you must have a syslog daemon running on the remote host, and it must not be configured to block syslog messages from the AP.

DISPLAYING EVENTS ON THE CONSOLE

By default, the system is configured to display events on the console as they occur. To disable this display, use this command:

```
> config system event display-events false
```

DISPLAYING DUPLICATE EVENTS

When identical event messages are logged consecutively within a five second period, the event manager recognizes duplicate events and issues a duplicate event message rather than a separate message for each event.

For example, if a CBQ tree is deactivated (row-status set to notInService), you

would see this display:

```
> config cbq.2 traffic-class.root-output-tree row-status notInService
Succeeded. Value(s) set to:
notInService
> # TM      (14)DEACTIVATING_CLASS   x10/1/1999 14:21:11:098 INFO
 Deactivating class root-output-tree on cbq.2.
```

Repeat the above *config* command nine more times at close intervals and then set row-status to active (to generate a different message) to see this display:

```
> config cbq.2 traffic-class.root-output-tree row-status active
Succeeded. Value(s) set to:
active
> # EVT     ( 7)DUPLICATE_EVENT      x10/1/1999 14:21:20:068 INFO
 The previous event occurred 9 times
> # TM      (13)ACTIVATING_CLASS     x10/1/1999 14:21:20:068 INFO
 Activating class root-output-tree on cbq.2.
```

Note that the DUPLICATE_EVENT message indicates how many times the preceding event occurred.

Displaying Detailed Packet Information in Events

If a packet is dropped for any reason, a detailed message about the packet is logged to an event in RAM and NVRAM. Users can see the detailed packet information such as source and destination IP address, source and destination port, and application as shown in this sample output.

```
#0 9/17/1999 14:24:54:082 IKE initialization_error WARN Failed to initialize
ISAKMP server
  #1 9/17/1999 14:24:54:095 TM FILTERING_PACKET WARN Filtering packet on
cbq.1, as the pdu size exceeds configured
maximum.
        Src IP: 1.1.1.1         Dst IP: 224.0.0.9
        Src Port: 4608          Dst Port: 3574          Application: unknown
  #2 9/17/1999 14:24:54:096 TM FILTERING_PACKET WARN Filtering packet on
cbq.1, as the pdu size exceeds configured
maximum.
        Src IP: 1.1.1.1         Dst IP: 224.0.0.9
        Src Port: rip           Dst Port: rip               Application: rip
```

Displaying Event Parameter Settings

To display the options that you have set for the max-log-size, clear-all-events, send-events-as-traps, and display-events parameters, use this command:

```
> show system event summary

                                SYSTEM EVENT SUMMARY

        Max Log Size* = 200 number of events
     Clear All Events* = ready
Send Events As Traps* = false
       Display Events* = false
```

SAMPLE EVENT FILTERING CONFIGURATION

Event filtering can be a very effective way to reduce the number and type of messages reported by the AP. In this example, all the event messages for the Ethernet subsystem and most of the event messages associated with the IP subsystem are turned off using these steps.

1. Make sure global filters are set to debug+trace.

```
> show system event global-filter

                                SYSTEM EVENT GLOBAL-FILTER

          In Filter Level* = debug+trace
         Out Filter Level* = ""
Out Filter Date And Time* = 00/00/0000 00:00:00:00
```

2. Make sure events will be displayed at the terminal.

```
> show system event summary

                                SYSTEM EVENT SUMMARY

        Max Log Size* = 200 events
     Clear All Events* = ready
Send Events As Traps* = false
       Display Events* = true
```

3. Filter the entire Ethernet subsystem (ETHD).

```
> config system event sub-system-filter.ETHD filter input
```

4. Filter all IPv4 subsystem events.

```
> config system event sub-system-filter.IPv4 id-filter.* filter input
```

5. Allow IPv4.3 and IPv4.13 subsystem events.

```
> config system event sub-system-filter.IPv4 id-filter.3 filter none

> config system event sub-system-filter.IPv4 id-filter.13 filter none
```

6. Turn on trace.

```
> config system event in-filter-level !trace
```

7 Perform ping test. You should see output similar to the following on the remote router.

```
> # IPv4 ( 3)unicastRcv x8/6/1999 13:11:10:068 TRACE
Received packet from ip.2 192.182.182.1
> # IPv4 ( 3)unicastRcv x8/6/1999 13:11:10:068 TRACE
Received packet from ip.7 192.182.182.1
> # IPSC (25)dispatch_ok x8/6/1999 13:11:10:068 TRACE
Packet SA 192.182.182.2 DA 192.182.182.1 from tunnel test dispatched
okay up to ipsec.1.1
> # IPv4 (13)slowSend x8/6/1999 13:11:10:068 TRACE
Slow path send packet from 196.57.4.1 to 192.182.182.1, next hop ip.2
192.182.182.1
>
```

This output shows that the ping packet arrived via the tunnel and the return is not through the tunnel.

8 Turn off trace.

```
> config system event in-filter-level trace
```

You should be prepared to capture the results when you turn on trace and be prepared to turn off trace as soon as your test is complete.

GENERATING CONFIGURATION SCRIPT COMMANDS

The *gen-script* command lets you display groups of configuration parameters in a CLI command format. The resulting output is displayed only on the console. You can cut and paste the configuration commands on the command line to reproduce parts of a particular configuration. You can display the configuration parameters for the entire router (all) or for a selected component (for example, atm, atm-ds3, bridge, eth, or frame-relay).

NOTE *The gen-script command does not determine hardware configurations or display the stack command, so it cannot be used to replay entire configurations.*

The *gen-script* syntax is as follows:

```
> gen-script [all|<component>] [defaults|no-defaults]
```

where <component> is one of the following: all, atm, atm-ds3, atm-sonet, cbq,

bridge, dns, eth, frame-relay, hssi, ip, ipoa, ppp, system, t1, t3.

Both command arguments are optional, and the default values are *all* and *defaults*. When you use the *no-defaults* argument, the script uses only the component's parameters that a user has previously modified.

The following sample command would display a configuration script for the ip component that lists only the modified parameters.

```
> gen-script ip no-defaults
```

USING THE SYNC COMMAND

The *sync* command is a powerful command that allows you to transfer selected objects from one configured Access Point device to another AP. The *sync* command can be used to transfer the following components:

Table 3 Transferrable Components

• backup	• dhcp-relay	• user-interface
• boot-images	• cbq	• date-and-time
• file-transfer-table	• policy-receive-table	• policy-send-table
• scheduler	• snmp-community-table	• snmp-trap-table
• system		

ACCESSING TARGET SYSTEMS

For the *sync* command to work, the target system you designate must have an SNMP community with read/write privileges. You can use *admin* (which has read/write privileges) if no other read/write community exists on the target.

In addition, you must use the *-remotecommunity* variable described in Table 4 to set the community on your system to match the name of a read/write community on the target.

SYNCING COMPONENTS WITH SECURITY FEATURES

The *sync* command does not incorporate authentication or encryption and, thus, does not transfer information in a secure manner.

For security reasons, key information and community/user information is not transferred when you sync the SNMP community table, authentication user table, or RADIUS subsystem.

While corresponding entries for other features are created, keys and community/user information cannot be created on remote nodes. This is because the sync process cannot read this information on the local node, and so cannot replicate this information on a remote (target) node.

USING SYNC

You can transfer any of the components listed in Table 3 to a target system, using the *sync* command in the form:

```
> sync  -remotecommunity <r/w community on target>
```

```
> sync <component> <target IP address>
```

Provide the IP address of the target system whose components you want to replace or update. You can choose any or all of the components listed in Table 3.

MODES OF OPERATION

The *sync* command operates in two modes, Replace and Update. The default setting is Replace mode (-R). In this mode, the source system attempts to replace all information on the designated target. This can be somewhat destructive because the Replace operation attempts to remove and create all rows that do not match its own. In contrast, Update mode (-U) is more conservative. Update seeks only to update rows shared by both source and target system.

To change the sync mode, proceed as follows:

```
> sync -mode <-U | -R>
```

Note that Update and Replace modes are denoted by capital U and R, respectively.

CUSTOMIZING SESSION VARIABLES

To make it easier to sync a number of different systems during a session, you can also customize a number of local sync variables. Settings assigned to these variables are not preserved across sessions. Variables that can be customized

are:

Table 4 Variables and Their Uses

Variable	Purpose and Meaning
-timeout	Specifies the number of seconds allowed for get/set requests. Valid range is 1 - 60 seconds. Default: 5 seconds
-remotecommunity	Specifies the SNMP community name used for the target system. Default: 'private' (only for system upgrades)
-retries	Specifies the number of retries after a timeout occurs. Valid range: 0 - 10. Default: 2
-mode	Specifies mode of operation as either -R for Replace or -U for Update. Default: -R

To set any of these variables, proceed as follows:

> sync <variable> <value>

Remember that in the case of *-mode*, entries are case sensitive.

GETTING HELP

Access Point's command line interface provides detailed help on the *sync* command, and on each sync-able system component.

To get general help on the *sync* command, enter the following:

>help sync info

To get help on any sync-able component, enter the following:

> help sync <component name>

CONFIGURING USER AUTHENTICATION

User authentication provides a means to specify access rights for particular users. The AP stores this authentication information in the user table for all users (including SSH, L2TP, PPP, and IPSec clients). Depending on the user, you might assign all or some of the following information for authentication purposes:

- User name/password pair — The user name identifies the user in the user table to be authenticated by the specified password.
- Client group — The client group specifies the group to which a user belongs. Client groups are specific to dial-in tunnels.
- Authentication method — The authentication method determines how the user is authenticated.
- SSH privileges — The SSH privileges define the SFTP access level and are specific to the SSH clients.

A sample set of entries is shown in Figure 9.

Figure 9 User Table

SERVICES AUTHENTICATION USER-TABLE

NAME	Password Ssh Privileges	Client Group Row Status	Authentication Method
(default)	--------	(default)	radiusAuthentication
	noAccess	active	
bill	--------	(default)	alwaysAccept
	sftpRead	active	
chris	--------	(default)	internalAuthentication
	sftpRead	active	
davidc_new	--------	(default)	internalAuthentication
	sftpRead	active	
davidc_old	--------	(default)	internalAuthentication
	session	active	
jim	--------	(default)	internalAuthentication
	noAccess	active	
mike	--------	(default)	internalAuthentication
	noAccess	active	

ABOUT THE USER TABLE

Let’s examine the user table in Figure 9 and note some of the user characteristics. Each user has a status of active. For security reasons, the password assigned to each user is not displayed in the user table. All users are associated with the default client group because they were not assigned to a client group.

If any users were associated with a specific client group, then packets would be sent to the subnet configured for the specified client group. Client groups are only used by dial-in clients.

Users are being authenticated with different methods. These methods and how to specify them are described in more detail on Page 91.

Different users have various levels of SSH privileges to allow or disallow them SFTP access rights. SSH privileges are only used by SSH clients.

About the Default User

The first entry in the user table is for the default user. This user is set up to accommodate all connections for which no user name/password information exists. Thus, when an unknown user connects to the AP, the AP first checks the user table. If information for the user does not exist, the user is assigned the characteristics associated with the default user. By default, the default user is authenticated with information from the RADIUS server.

You can configure this user like any other. However, for security purposes, the default user should be authenticated by using information from the RADIUS server (*radiusAuthentication*) or by never allowing access (*alwaysReject*).

Configuring Users

To configure users in the user table:

```
> config services authentication user.<name> password <string> \
  client-group <group name> authentication-method <method> \
  ssh-privileges <privileges>
```

The rest of this section explains how to configure the user name and password. It also describes how to configure the authentication method and provides some information about configuring client groups and SSH privileges.

Specifying User Name and Password

At a minimum, you need to specify the user name. You could specify the password, but that depends on the chosen authentication method. For security reasons, passwords are always displayed as dashes (-) in the user table. Use the following command to specify the user name and password:

```
> config services authentication user.<name> password <string>
```

When specifying the user name for the default user, you must include the parentheses as shown:

```
> config services authentication user.(default)
```

SPECIFYING THE AUTHENTICATION METHOD

The authentication method specifies how the user is authenticated. To specify the authentication method:

```
> config services authentication user.<name> authentication-method
  [internalAuthentication|alwaysAccept|alwaysReject|radiusAuthentication]
```

The four methods are:

- *internalAuthentication* — use internal authentication, user name and password are used to accept or reject the user
- *alwaysAccept* — always accept a user, regardless of password used
- *alwaysReject* — never allow a user access
- *radiusAuthentication* — use RADIUS authentication

You can configure the authentication method for individual users or use the default method (*internalAuthentication*) by not specifying the authentication method for a particular user. Users configured for internal authentication are authenticated with the information in the user table.

The rest of this section explains how to assign particular methods to individual users.

CONFIGURING USERS TO BE AUTHENTICATED LOCALLY

Many sites store user name/password information for each user on the AP in the user table. If you do not specify an authentication method for a user, the AP uses internal authentication by default (that is, it uses the information provided in the user table for authentication).

To specify internal authentication for a particular user:

```
> config services authentication user.<name> password <string> \
  authentication-method internalAuthentication
```

CONFIGURING USERS TO BE AUTHENTICATED BY RADIUS

Sites can store authentication information for each user on a designated RADIUS server. When a user assigned the RADIUS authentication method connects, the AP contacts the RADIUS server for authentication information.

To specify RADIUS authentication for a particular user:

```
> config services authentication user.<name> \
  authentication-method radiusAuthentication
```

You can find information on configuring user name/password and tunnel names for RADIUS in your RADIUS server documentation.

With respect to the RADIUS server, the AP functions as a client. To use RADIUS authentication, you must configure the AP as a RADIUS client, and add at least one RADIUS server configuration record. See "Configuring RADIUS Authentication" on Page 507 for more information.

CONFIGURING USERS TO BE ALWAYS ACCEPTED OR REJECTED

At times, there might be users whose connections you always want to reject. To specify that a particular user should never be allowed access:

```
> config services authentication user.<name> \
  authentication-method alwaysReject
```

For example, you can configure the default user to automatically reject unknown users (those users not listed in the user table) instead of attempting to authenticate them with this command:

```
> config services authentication user.(default) \
  authentication-method alwaysReject
```

Conversely, there might be users whose connections you always want to allow. You can accomplish this with a user definition like the following:

```
> add services authentication user.Dolly password madison \
  authentication-method alwaysAccept
```

This entry ensures that connections from Dolly will always be accepted.

SPECIFYING CLIENT GROUPS

The name of a specific client group indicates the user's membership in the group. When a user connects, the AP can authenticate the user and assign the dial-in connection to a specific client group without consulting any other server.

To add a user name/password pair and specify client group membership:

```
> add services authentication user.<user> password <string> \
  client-group <client-group-name>
```

The user name and password chosen must be unique on the AP. The client-group-name chosen determines the subnet and mask from which the AP will choose an enterprise address for the packet. For more information about authenticating dial-in users and client groups, see "Authenticating Dial-in Users" on Page 437.

SPECIFYING SSH PRIVILEGES

While the user table is used for all dial-in connections, the Ssh Privileges parameter is specific to the SSH server and is particularly relevant with respect to the use of SFTP.

To specify the SSH privileges for a user:

```
> config services authentication user.<name> ssh-privileges
  [noAccess|session|sftpRead|sftpReadWrite]
```

The default value is *noAccess*. For more details about this parameter, see "Configuring SFTP Access for Individual Users" on Page 545.

EXAMPLE: ADDING USERS

To add a user named chrisn with the password randomname, proceed as shown here. Note that a client group name (fella) has been specified. This is the client group in which this user has membership.

```
> add services authentication user.chrisn password randomname \
  client-group fella
```

Following are additional user definitions. Note that each of these users has membership in a different client group than chrisn:

```
> add services authentication user.jackhorner password satinacorner \
```

```
authentication-method alwaysAccept client-group weak

> add services authentication user.joe password okay
```

In fact, the users just added have different characteristics, and will be treated differently by the AP. How their treatment differs becomes clear if we look at the user table, which contains information on all these users. Also included in this table is the default user, which applies to any packet for which other information is unavailable.

To display the user table:

```
> show services authentication user-table summary
```

```
                    SERVICES AUTHENTICATION USER-TABLE
          NAME|       Password|     Client Group| Authentication Method|
              |  Ssh Privileges|       Row Status|                      |
--------------|----------------|-----------------|----------------------|
     (default)|        --------|        (default)|  radiusAuthentication|
              |      noAccess  |          active |                      |
        chrisn|        --------|           fella|internalAuthentication|
              |      noAccess  |          active |                      |
    jackhorner|        --------|             weak|          alwaysAccept|
              |      noAccess  |          active |                      |
          joe |        --------|        (default)|internalAuthentication|
              |      noAccess  |          active |                      |
```

Let’s examine this user table. Each of the user name/password pairs has a status of active. For security reasons, the password assigned to each user is not displayed in the user table. All users have *noAccess* as their SSH privileges.

Apart from these similarities are a number of differences. Users chrisn and jackhorner are associated with different client groups. Client groups determine the subnet and mask from which the AP will choose an enterprise address for the packet. Packets sent on connections by these users will be handled accordingly. User joe has not been assigned to a client group (hence, the (default) in the table). Unless the administrator assigns joe to a client group, packets sent by joe will be sent to the subnet configured for the default client group.

The default user will be authenticated by RADIUS. Users chrisn and joe will be authenticated using information from the user table. User jackhorner is always accepted.

ENABLING USER PASSWORD AUTHENTICATION

Before users can be authenticated with the information in the user table, you must enable user password authentication for the appropriate subsystem (for example, IPSec, PPP, or SSH).

To authenticate connections for L2TP, PPP, and IPSec clients, enable user password authentication for the PPP and IPSec interfaces (the default setting). Refer to Chapter 9 for information about configuring IPSec or L2TP dial-in tunnels. See Page 172 for information about configuring PPP authentication.

To authenticate users by user table, enable user authentication on the SSH server. Refer to Page 517 for information about configuring SSH.

4

CONFIGURING INTERFACES

This chapter explains how to configure the physical interfaces for the Access Point system (AP). The AP supports several physical interfaces and a number of data links. Certain data links, such as Frame Relay, can run over different physical interfaces. An IP instance is required to forward traffic over any data link. Accordingly, this chapter is organized into three main areas:

PHYSICAL INTERFACES

Configuring Ethernet and Gigabit Ethernet *98*
Configuring HSSI and MSSI Interfaces *101*
Configuring T1 and E1 Links *103*
Configuring T3 Interfaces *115*
Configuring ATM *117*
Configuring Integrated Services Digital Network *144*
Configuring Packet over SONET *159*

DATA LINK INTERFACES

Configuring Frame Relay *163*
Configuring PPP *168*
Configuring Switched Multimegabit Data Service *175*

IP INTERFACES

Configuring IP *178*

Note that although Ethernet, ATM, and ISDN are listed in the physical

interfaces section, those sections explain how to configure Ethernet for your LAN connections and ATM and ISDN for your WAN connections.

APs support the creation of subinterfaces for ATM, Frame Relay, and PPP (Multilink PPP) data links. Subinterfaces provide discrete instances of the connectivity provided by the main or *integral* interface type. Integral interfaces and subinterfaces provide identical functionality. You can use them together to create independent instances of the connectivity provided by the data link. Information about subinterfaces is provided in the sections for each data link.

CONFIGURING ETHERNET AND GIGABIT ETHERNET

The AP supports Ethernet and Gigabit Ethernet for connection to local area networks using either a 10/100BaseT or 1000BaseSX interface respectively. Both interfaces share most commands and parameters. Table 5 shows the key differences:

Table 5 Ethernet and Gigabit Ethernet Differences

Parameter	Ethernet	Gigabit Ethernet
duplex-mode	half (default) / full	full (default)
config-speed	10 / 100 (default) Mbps	1 Gbps (default)
media-type	tx (default)	sx (default)
autonegotiate	N/A	enabled (default)
flow-control	N/A	settable

CHANGING ETHERNET SETTINGS

To configure Ethernet to 100 Mbps and full duplex, enter this command:

```
> config eth.<instance> config-speed mbps100 duplex-mode full
  autonegotiate disabled flow-control disabled
```

To verify speed and duplex settings for the link, enter the following command:

```
> show eth.1 config
```

```
                                ETH.1 CONFIG

          MTU* = 1500 bytes                 Media Type  = tx
Up Down Trap* = enabled     Capture Effect Resol* = disabled
        Alias* = ""                     Autonegotiate* = disabled
Config Speed* = mbps100                  Flow Control* = disabled
  Duplex Mode* = half
```

ADDING AN IP ADDRESS TO EACH INTERFACE

Access Point interfaces do not come up with a default IP address. You must assign an IP address to each interface to establish IP connectivity.

Initially, you must connect to the AP from a PC or terminal via a serial line to establish LAN side connectivity as part of the installation procedure.

Add an IP address to each Ethernet interface using this command.

```
> add ip.<instance> address.<address>
```

Once you have added an IP address to a LAN side interface, you can connect to the device over Ethernet.

CONFIGURING THE ARP TABLE SIZE

You can configure the size of the ARP table for Ethernet. When you change the ARP table size, it takes effect immediately. Any static ARP entries are preserved, but dynamic entries are removed and learned again as needed. You should not make the ARP table size equal or less than the number of defined static entries because that leaves no room for dynamic entries. The allowed range for the ARP table size is 1 to 5000. The default is 1000.

You can change the ARP table size as follows:

```
> config eth.1 arp-config arp-table-size 100
```

CONFIGURING GIGABIT ETHERNET

Gigabit Ethernet uses two parameters to set up and control its link: autonegotiation and flow control.

AUTONEGOTIATION

When you enable autonegotiation, the AP exchanges messages with a peer system in an attempt to bring up a 1 Gbps, full-duplex link with symmetrical flow control. To establish a gigabit link, both the AP and peer system must have autonegotiation enabled. When autonegotiation is enabled, manually configured settings are ignored. The default is enabled.

```
> config eth.1 autonegotiate enabled
```

You can show autonegotiated settings with this command:

```
> show eth.1 negotiated-config
```

```
                          ETH.1 NEGOTIATED-CONFIG

           State  = up
    Autonegotiate* = enabled
       Nego Speed  = gbps1
 Nego Duplex Mode  = full
Nego Flow Control  = enabledXmitAndRcv
```

If autonegotiation fails, the settings appear as follows:

```
                          ETH.1 NEGOTIATED-CONFIG

           State  = down
    Autonegotiate* = enabled
       Nego Speed  = unknown
 Nego Duplex Mode  = unknown
Nego Flow Control  = unknown
```

If you have autonegotiation disabled, the interface attempts to bring up the link according to the flow-control settings that you have configured or the link will not be established. (In this case, autonegotiation should be disabled on the peer Gigabit Ethernet interface.)

FLOW CONTROL

Flow control provides lossless packet exchange on the gigabit link. When autonegotiation is enabled, the typical negotiated flow control setting is enabledXmitAndRcv, meaning the port acts on incoming pause flow control packets and generates them in the outgoing direction as needed (symmetrical

operation). Other settings for flow control include enabledRcv and enabledXmit. The default is disabled and is only used when autonegotiate is disabled.

```
> config eth.1 flow-control enabledXmitAndRcv
```

To show how flow control is configured, use this command:

```
> show eth.1 config
```

```
                                ETH.1 CONFIG

          MTU* = 1500 bytes           Media Type  = sx
Up Down Trap* = enabled    Capture Effect Resol* = disabled
        Alias* = ""                 Autonegotiate* = enabled
 Config Speed* = gbps1               Flow Control* = disabled
  Duplex Mode* = full
```

If autonegotiate is enabled, you can show the flow control negotiated with the *show eth.<instance> negotiated-config* command.

CONFIGURING HSSI AND MSSI INTERFACES

The following section explains how to configure HSSI and MSSI connections. After configuring the physical layer, you must make sure the data link (for example, Frame Relay or PPP) and IP addresses are configured properly for your interface.

Frame Relay networking is the default. A Frame Relay connection comes up automatically upon completion of the HSSI or MSSI configuration. See Page 163 for detailed instructions on setting up Frame Relay.

Configuring PPP entails replacing the default frame relay connection with an instance of PPP. See Page 168 for detailed instructions on setting up PPP.

You must add an IP address to an interface to bring up the link. See Page 178 for detailed instructions about adding IP addresses to an interface.

SETTING UP THE HSSI INTERFACE

Examine the AP defaults for the High Speed Synchronous Interface (HSSI).

These values should suffice for the majority of cases.

```
> show hssi.1 config

                                        HSSI.1 CONFIG

          MTU* = 4470 bytes
Up Down Trap* = enabled
        Alias* = ""
     Clocking* = external
     CRC Mode* = crc16
```

By default, the AP sets HSSI clocking to external. Clocking for the connection is performed by the attached DSU/CSU framing device. Connect to the DSU/CSU with a straight-through HSSI cable. HSSI cables are orderable from your distributor or reseller.

NOTE *If you are running the AP back-to-back with no DSU/CSU present, set clocking to internal on both sides and use a HSSI crossover cable to connect the devices.*

SETTING UP THE MSSI INTERFACE

The Medium Speed Synchronous Interface (MSSI) provides a high-performance WAN connection using V.35 DTE and X.21 DTE interface types. MSSI connections operate at speeds up to 10 Mbps and supports the HDLC framing required for Frame Relay and PPP connectivity. Frame Relay is the default data link for MSSI. A Frame Relay connection comes up automatically upon completion of the MSSI configuration.

Examine the AP defaults for MSSI. These default values should suffice for the majority of cases.

```
> show mssi.1 config

                                        MSSI.1 CONFIG

          MTU* = 4470 bytes
Up Down Trap* = enabled
        Alias* = ""
     Clocking* = external
     CRC Mode* = crc16
```

By default, the AP sets MSSI clocking to external. Clocking for the connection is performed by the attached DSU/CSU framing device.

NOTE *For MSSI connections, the clocking must be set to external.*

Verify that the clocking is set to external and the CRC mode is set to crc16. If these values are not set, enter the following commands as appropriate:

```
> config mssi.1 clocking external
> config mssi.1 crc-mode crc16
```

CONFIGURING T1 AND E1 LINKS

Access Point routers support Frame Relay, SMDS or PPP over T1 or E1 links. The AP supports quad T1/E1 cards, with four dedicated T1/E1 lines for each card. The AP supports Fractional T1/E1 connections as explained on Page 112.

ABOUT T1/E1 CONNECTIONS

While the AP supports T1 and E1 connections, T1 is the default link type. The AP handles T1 and E1 connections similarly, using some of the same defaults and requiring the same essential set-up procedure.

Once you establish a T1or E1 link, the AP receives a Data Link Connection Identifier (DLCI) from a Frame Relay switch on the WAN, and brings up a Frame Relay connection. The connection should come up shortly after the T1 or E1 link is active.

To show active DLCIs at any time, issue this command:

```
> show frame-relay.* dlcis summary
```

For more information on setting up and validating Frame Relay connections, see "Setting Up Frame Relay PVCs" on Page 164. As mentioned earlier, you can also run PPP or SMDS over T1/E1 connections. See "Configuring PPP" on Page 168 for information on configuring PPP. For information on configuring SMDS, see "Configuring Switched Multimegabit Data Service" on Page 175.

NOTE *Connection of any data communications device to the telephony network requires you to submit certain information to your network provider. A template for the information you need to submit is provided in the preface to your installation guide.*

You must add an IP address to an interface to bring up the link. See Page 178 for detailed instructions about adding IP addresses to an interface.

DISPLAYING SLOT COMPONENTS

To display the relationship between T1/E1 instances and Access Point slot/port combinations, issue a *status* command as shown in Figure 10.

Figure 10 Status Screen, Access Point T1/E1

SYSTEM STATUS

ROW			Layer Status	
1	slot.1.1	Tcom4 T1 Driver	enabled	down
2	slot.1.1	cbq.1	enabled	down
3	slot.1.1	t1.1	enabled	down
4	slot.1.1	frame-relay.1	enabled	down
5	slot.1.1	ip.1	enabled	down
6	slot.1.2	Tcom4 T1 Driver	enabled	down
7	slot.1.2	cbq.2	enabled	down
8	slot.1.2	t1.2	enabled	down
9	slot.1.2	frame-relay.2	enabled	down
10	slot.1.2	ip.2	enabled	down
11	slot.1.3	Tcom4 T1 Driver	enabled	down
12	slot.1.3	cbq.3	enabled	down
13	slot.1.3	t1.3	enabled	down
14	slot.1.3	frame-relay.3	enabled	down
15	slot.1.3	ip.3	enabled	down
16	slot.1.4	Tcom4 T1 Driver	enabled	down
17	slot.1.4	cbq.4	enabled	down
18	slot.1.4	t1.4	enabled	down
19	slot.1.4	frame-relay.4	enabled	down
20	slot.1.4	ip.4	enabled	down
21	slot.2.1	Tcom4 T1 Driver	enabled	down
22	slot.2.1	cbq.5	enabled	down
23	slot.2.1	t1.5	enabled	down
24	slot.2.1	frame-relay.5	enabled	down
25	slot.2.1	ip.5	enabled	down
26	slot.2.2	Tcom4 T1 Driver	enabled	down
27	slot.2.2	cbq.6	enabled	down
28	slot.2.2	t1.6	enabled	down
29	slot.2.2	frame-relay.6	enabled	down
30	slot.2.2	ip.6	enabled	down
31	slot.2.3	Tcom4 T1 Driver	enabled	down
32	slot.2.3	cbq.7	enabled	down
33	slot.2.3	t1.7	enabled	down
34	slot.2.3	frame-relay.7	enabled	down
35	slot.2.3	ip.7	enabled	down
36	slot.2.4	Tcom4 T1 Driver	enabled	down
37	slot.2.4	cbq.8	enabled	down
38	slot.2.4	t1.8	enabled	down
39	slot.2.4	frame-relay.8	enabled	down
40	slot.2.4	ip.8	enabled	down
41	slot.3.1	LCM5200 HSSI Driver	enabled	down
42	slot.3.1	cbq.9	enabled	down
43	slot.3.1	hssi.1	enabled	down
44	slot.3.1	frame-relay.9	enabled	down
45	slot.3.1	ip.9	enabled	down
46	slot.4.1	Ethernet Driver	enabled	down
47	slot.4.1	cbq.10	enabled	down
48	slot.4.1	eth.1	enabled	down
49	slot.4.1	ip.10	enabled	down
50	virtual	Virtual Tunnel Transport	enabled	down
51	virtual	cbq.11	enabled	down
52	virtual	ipsec.1	enabled	down
53	virtual	ip.11	enabled	down

NOTE *E1 links are shown as T1 (t1.<instance>, rather than e1.<instance>) when you issue a status command or show summary information for a specific E1 instance.*

SETTING UP T1 LINKS

To set up T1 links, you must:

- Determine which components are T1 links
- Examine the physical layer defaults
- Examine T1 defaults

DETERMINE T1 LINKS

To determine which components are T1 links, use this command:

```
> show t1-interfaces physical-layer config
```

Examine the display to find those entries that do *not* contain e1 as the interface type. These entries are the T1 links.

EXAMINE PHYSICAL LAYER DEFAULTS

To examine the physical layer defaults for each T1 line, use a command similar to the following and replace it with the appropriate instance.

```
> show t1.1 physical-layer config
```

```
                        T1.1 PHYSICAL-LAYER CONFIG

              Line Type* = t1ESF          Interface Type* = t1csu00db0
            Line Coding* = b8zs            Timeslot Mask* = 1-24
              Circuit ID* = ""               Invert HDLC* = disabled
Data Link Facilities* = dsx1Fdl-none          Idle Code* = ic7e
            Clock Source* = external          Equalizer* = disabled
```

These defaults should be sufficient for the majority of T1 connections. Check the following values, and adjust as needed.

LINE TYPE

ESF framing (t1ESF) is the default line type.

The line type dictates the framing pattern used by the port. This parameter must agree with the equipment on the other side of the line. If you want to use Super Framing, specify t1D4 as the line type.

LINE CODING

B8ZS (b8zs) is the default type of line coding.

The line coding dictates the protocol for ensuring ones density on the line. This parameter should agree with the equipment on the other side of the line. If the default value is not used, some other method must be used to maintain the ones density. For example, if you want to use an external device, such as a CSU, specify AMI (ami).

DATA LINK FACILITIES

None (dsx1Fdl-none) is the default value.

This parameter specifies the type of in-band management used when the ESF line type is specified. If you want unsolicited one-second performance reports to be sent to the remote end over the facility data link (FDL), specify T1.403 (dsx1Ansi-T1-403).

CLOCK SOURCE

External is the default clock source.

Only set clocking to internal if you are running the Access Point T1/E1 back-to-back with another AP, using a crossover cable.

INTERFACE TYPE

A CSU application with a line build out of 0 is the default value (t1csu00dbA).

This parameter specifies the interface options. The T1 value depends on the application (whether it is a CSU or simple dsx1) and the line build out value. If it is a CSU application, then a line build out between 0 and 22 db must be selected (t1csu00dbA, t1csu07dbB, t1csu15dbC, or t1csu22db). If it is a dsx1 application, then the distance (between 0 and 533 feet) from the network interface (CSU) must be selected (t1dsx000ft, t1dsx133ft, t1dsx266ft, t1dsx399ft, or t1dsx533ft). Configure the line build out value as designated by the service provider.

INVERT HDLC

Disabled is the default value.

HDLC inverting mode lets you use the HDLC bit stuffing rule in reverse when using AMI line coding to enhance the interoperability between the AP and the CSU.

IDLE CODE

ic7e (0x7e idle code pattern) is the default idle code.

The idle code is a fixed eight-bit word that contains at least two ones. It is used to maintain ones density on the line. This parameter should agree with the equipment on the other side of the line. You can specify icff (0xff) or iczero (0x0) idle code patterns.

EQUALIZER

The adaptive equalizer mode is enabled by default.

The adaptive equalizer mode ensures that the correct set of T1 line coefficients is loaded.

EXAMINE T1 DEFAULTS

Next, examine the Access Point T1/E1 defaults using this command:

```
> show t1.1 config
```

```
                                T1.1 CONFIG

     Clocking* = external
     CRC Mode* = crc16
          MTU* = 4470 bytes
Up Down Trap* = enabled
        Alias* = ""
```

Like the physical layer defaults, these should suffice for the majority of cases.

By default, the AP sets T1 clocking to external. Clocking for the connection is performed by the network. To show T1 physical layer parameters, issue this command:

```
> show t1.<instance> physical-layer config
```

This sample screen shows the physical layer parameters as they can be set for T1 links.

```
                        T1.1 PHYSICAL-LAYER CONFIG

                Line Type* = t1ESF          Interface Type* = t1csu00db0
              Line Coding* = b8zs            Timeslot Mask* = 1-24
                Circuit ID* = ""               Invert HDLC* = disabled
Data Link Facilities* = dsx1Fdl-none            Idle Code* = ic7e
             Clock Source* = external           Equalizer* = enabled
```

SETTING UP E1 LINKS

To set up E1 links, you must:

- Configure the E1 physical layer values
- Determine which components are E1 links
- Check E1 physical layer configuration

CONFIGURE E1 PHYSICAL LAYER

Physical layer values for E1 differ in several areas from those used for T1 connections. Specifically, you must set the parameters for line type, line coding, clock source, data link facilities, and interface type. Once you make the changes outlined here, the AP supports E1 (rather than T1) on each T1 link.

Note that E1 links continue to be shown as T1 (t1.<instance>, rather than e1.<instance>) when you issue a *status* command or show summary information for a specific E1 instance.

LINE TYPE

Issue this command to set the appropriate line type for E1 links:

```
> config t1.1 physical-layer line-type e1FAS
```

You can also use line-type e1FAScrc if you need to support CRCs on the line.

LINE CODING

Issue this command to set the appropriate line coding for E1 links:

```
> config t1.1 physical-layer line-coding hdb3
```

CLOCK SOURCE

Issue this command to set the appropriate clock source for E1 links:

```
> config t1.1 physical-layer clock-source external
```

You can also set the clock source to internal if you are running the AP T1/E1 back-to-back with another AP, using a crossover cable.

DATA LINK FACILITIES

Issue this command to set the data-link-facilities parameter appropriately:

```
> config t1.1 physical-layer data-link facilities dsx1Fdl-none
```

INTERFACE TYPE

Issue this command to set the interface type to E1:

```
> config t1.1 physical-layer interface-type e1
```

DETERMINE E1 LINKS

To determine which components are E1 links, use this command:

```
> show t1-interfaces physical-layer config
```

Examine the display to find those entries that contain e1 as the interface type. These entries are the E1 links.

DISPLAY THE E1 PHYSICAL LAYER CONFIGURATION

To view E1 physical-layer parameters, issue this command for the appropriate interface instance:

```
> show t1.<instance> physical-layer config
```

NOTE *You must specify T1 (rather than E1) as the link type.*

The following sample screen shows the parameters as they should be set for an E1 link:

```
                              T1.1 PHYSICAL-LAYER CONFIG

                  Line Type* = e1FAS          Interface Type* = e1
                Line Coding* = hdb3           Timeslot Mask* = 1-24
                  Circuit ID* = ""              Invert HDLC* = disabled
Data Link Facilities* = dsx1Fdl-none              Idle Code* = ic7e
              Clock Source* = external          Equalizer* = enabled
```

SETTING UP T1 AND E1 LINKS IN LOOPBACK MODE

The AP supports loopback tests to validate T1 (local and remote) and E1 (local only) connections. Four types of loopback are supported.

- Remote Payload Loopback

  When running remote payload loopback, signals received on the link are looped back through the AP's framing function for transmission.

- Remote Line Loopback

  In remote line loopback mode, signals received on the link do not pass through the device, but are instead looped back out on the T1 or E1 line. You can run remote line loopback at the same time as local framer loopback.

- Local Analog Loopback

  In local analog loopback mode, signals transmitted on the link are looped back through the AP's framing function for receipt. The transmitter outputs must connect to a cable termination or a transmit termination resistor for the local analog loopback to pass signals.

- Local Framer Loopback

  In local framer loopback mode, signals transmitted on the link do not pass through the device, but are internally routed. You can run local framer loopback at the same time as remote line loopback.

FORCING THE LOCAL T1 OR E1 INTERFACE INTO LOOPBACK MODE

You can force the local T1 or E1 interface into loopback mode by running loopback tests directly from the command line.

You can also force the local T1 interface into or out of loopback mode by having the remote T1 device send loopback codes to the AP. Note that this procedure is not supported for E1 links.

RUNNING LOOPBACK TESTS FROM THE COMMAND LINE

To run any of the loopback tests, specify the appropriate loopback test with this command:

```
> config t1.<instance> physical-layer config-test loopback-config
  < remoteLineLoop | remotePayloadLoop | localFramer | localAnalog >
```

where the parameters indicate these loopback tests:

remoteLineLoop	Remote line loopback
remotePayloadLoop	Remote payload loopback
localFramer	Local framer loopback
localAnalog	Local analog loopback

Sending Loopback Codes From the Remote T1 Device

The T1 interface on the AP can be forced into or out of loopback mode when the remote device sends the loop up/down codes specified in ANSI T1.403. Note that this procedure is not supported for E1 links.

The loop activation code is 000010000100001. The loop deactivation code is 001001001001. These codes replace all DS1 data, not framing data.

Forcing Remote T1 Systems into Loopback Mode

This section outlines the procedure for sending and receiving loopbacks to and from remote T1 systems. Note that this procedure is not supported for E1 links.

1. Force the remote T1 system to go into remote payload loopback or remote line loopback by sending one of the appropriate sendcodes as follows.

   > config t1.1 physical-layer config-test send-code dsx1SendPayload

   > config t1.1 physical-layer config-test send-code dsx1SendLineCode

   On a T1 line, issuing either of these commands causes the remote system to enter the remote loopback state.

2. Send out a send-nocode message to stop the local T1 system from sending the sendcode pattern as follows:

   > config t1.1 physical-layer config-test send-code dsx1SendNoCode

   The remote system remains in the loopback state.

NOTE *Steps 1 and 2 describe how to put the devices into a specific state using the sendcode pattern. To send a specific test pattern, send a test code as described in step 3.*

3. Send a test code and generate a number of bit errors by issuing the following commands.

   > config t1.1 physical-layer config-test test-code sendPRBS2E20fz

> config t1.1 physical-layer config-test insert-bit-errors <number>

Note that the test code shown in this step is the standard Quasi-Random Sequence Signal (QRSS) pattern used to test T1 connectivity in most cases. The QRSS pattern sends a Pseudo Random Bit Sequence (PRBS) of 2E20-1 that is framed and includes Zero Code Suppression (ZCS), thus the test-code value of sendPRBS2E20fz.

You can send other test patterns (2E11-1, 2E15-1, 2E20-1, or 2E23-1) that are framed (as indicated by the letter f) or include ZCS (as indicated by the letter z) by using a different test-code value.

4 Check to see that the test code is sent back and that you are receiving all the bit errors inserted on the line by issuing this command:

> show t1.1 physical-layer interval-current

5 Stop sending the test code after line tests are concluded with this command:

> config t1.1 physical-layer config-test test-code sendNone

6 Be sure to take the remote system out of loopback mode by issuing this command:

> config t1.1 physical-layer config-test send-code dsx1SendResetCode

CONFIGURING FRACTIONAL T1/E1

Many users need the capabilities of a T1/E1 network, but few need a full complement of 24 DS0 channels (or 32 DS0s for E1) running into their facility. For those users who need less than 1.544 Mbps of bandwidth, Fractional T1 (FT1) lets them select DS0 channels individually, rather than in groups of 24 (or 32, in the case of E1). As a result, users can meet all of their communication needs with FT1 service, instead of underutilizing a full T1 circuit. For example, users could lease four channels for video applications, two for voice, and two for data communications.

When implementing FT1 service, you need to do the following:

- Configure speed
- Configure bandwidth
- Configure contiguous or non-contiguous DS0s

The three configuration parameters used in configuring Fractional T1 have to

agree for two networking devices to communicate. For instance, if device A had configured the 1, 4, 6, 8, 10, 12, 14, 16, 18, 20, and 22 DS0 channels, then device B must have the same DS0 channels configured to allow a simple ping packet to go through.

CONFIGURING SPEED

You must select between Nx56Kbps or Nx64Kbps, depending on the line coding supported by your service provider. Nx56Kbps is typically offered if the service provider uses alternate mark inversion (AMI) line coding, in which only 56 Kbps of the 64 Kbps DS0 is used for customer data. The remaining bandwidth is filled by the forced insertion of ones to maintain ones density. This service reduces throughput, but allows fractional service on a T1 span using AMI.

Nx64Kbps service is offered if the service provider has incorporated bipolar eight zero substitution (B8ZS) line coding into their T1 network. B8ZS allows use of the entire FT1 bandwidth by eliminating the forced insertion of ones.

To configure the speed, use the appropriate line coding (ami for Nx56Kbps service, b8zs for Nx64Kbps service):

```
> config t1.1 physical-layer line-coding < ami | b8zs >
```

CONFIGURING BANDWIDTH

Next, select the DS0 channels (amount of bandwidth) needed. FT1 service typically offers DS0s in groups of 2, 4, 6, 8, and 12 at the speed rate of 56 or 64 Kbps per channel. A full T1 channel contains 24 DS0s and E1 contains 32 DS0s. The most common fractional data rates offered are:

	Nx56Kbps (AMI)	**Nx64Kbps (B8ZS)**
1x	56 Kbps	64 Kbps
2x	112 Kbps	128 Kbps
4x	224 Kbps	256 Kbps
6x	336 Kbps	384 Kbps
8x	448 Kbps	512 Kbps
12x	672 Kbps	768 Kbps

To configure DS0s (time slots), use the following command:

```
> config t1.1 physical-layer timeslot-mask <time slot>
```

The timeslot-mask parameter can have a value in the range of 1-24 for T1 and 1-32 for E1. The default value of the timeslot-mask parameter is 1-24. To set the full range for E1, specify 1-32 as the value for the E1 instance.

```
> config t1.1 physical-layer timeslot-mask 1-32
```

You can enter individual time slots separated by commas or ranges separated by a hyphen. For instance, the following command configures the time slot as a range from 1 to 7 and individual time slots 16, 18, 20, and 22.

```
> config t1.1 physical-layer timeslot-mask 1-7,16,18,20,22
```

You can use the *show* command to verify that a specific time slot has been configured.

```
> show t1.1 physical-layer config
```

```
                        T1.1 PHYSICAL-LAYER CONFIG

          Line Type* = t1ESF              Interface Type* = t1csu00dbA
        Line Coding* = b8zs             Timeslot Mask* = 1-7,16,18,20,22
          Circuit ID* = ""                Invert HDLC* = disabled
Data Link Facilities* = dsx1Fdl-none        Idle Code* = ic7e
        Clock Source* = external            Equalizer* = enabled
```

CONFIGURING CONTIGUOUS OR NON-CONTIGUOUS DS0S

You must specify either contiguous or non-contiguous DS0s time slot configuration.

Contiguous bandwidth is DS0s that are routed together end-to-end to support applications greater than 56 or 64 Kbps. Applications such as data and video often require contiguous bandwidth throughout the T1 network, eliminating any delay problems caused by different routing paths through a digital area cross-connect system (DACS). If the service user's application is data-oriented and requires Nx64Kbps service, the service providers' T1 network must support B8ZS line coding. If not, contiguous bandwidth cannot be used for the application.

If you have already selected Nx64Kbps with line coding of B8ZS ("Configuring Speed" on Page 113), you are set up for contiguous DS0s. To configure contiguous DS0s, specify B8ZS line coding for the T1 instance:

```
> config t1.1 physical-layer line-coding b8zs
```

Non-contiguous bandwidth refers to individual DS0s that are not adjacent, but

are routed together through a DACS. Note that non-contiguous DS0s are used primarily for multimedia applications that do not require contiguous bandwidth. If Nx64Kbps service must be used for voice applications and the service provider does not offer B8ZS line coding, non-contiguous service must be used. However, you must configure the appropriate idle code (as supplied by service provider) to maintain the T1 ones density requirement. To configure non-contiguous DS0s to support data applications, specify the idle code pattern using this command:

```
> config t1.1 physical-layer idle-code < ic7e | icff | iczero >
```

where ic7e is the 0x7e, icff is the 0xff, and iczero is the 0x0 idle code pattern. For example, use this command to specify the 0x7e idle code pattern:

```
> config t1.1 physical-layer idle-code ic7e
```

CONFIGURING T3 INTERFACES

The AP supports Frame Relay, SMDS or PPP over T3 links. Once you establish a T3 link, the AP receives a Data Link Connection Identifier (DLCI) from a Frame Relay switch on the WAN, and brings up a Frame Relay connection. The connection should come up shortly after the T3 link is active. For more information on setting up and validating Frame Relay connections, see "Setting Up Frame Relay PVCs" on Page 164.

As mentioned earlier, you can also run PPP or SMDS over T3 connections. See "Configuring PPP" on Page 168 for information about configuring PPP. For information on configuring SMDS, see "Configuring Switched Multimegabit Data Service" on Page 175.

You must add an IP address to an interface to bring up the link. See Page 178 for detailed instructions about adding IP addresses to an interface.

EXAMINING PHYSICAL LAYER DEFAULTS

T3 provides a physical connection supporting rates of 45 Mbps. Most of the configuration takes place at the physical layer. To view physical layer defaults, use this command:

> show t3.1 physical-layer config

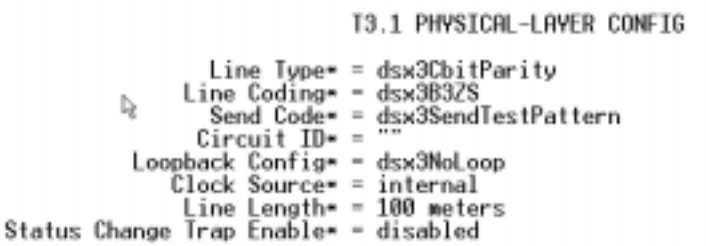


Table 6 shows the values that can be assigned to each physical-layer parameter.

Table 6 Configurable T3 Physical Layer Parameters and Values

T3 Attribute	Value
Line Coding	B3ZS only
Line Type	C Bit parity only
Line Length	Set to length in meters. 100 is the default. See "Setting Line Length" on Page 116.
Send Code	SendTestPattern only
Circuit ID	This variable contains the transmission vendor's circuit identifier, for use in troubleshooting operations.
Loopback Config	lineLoop - loopback everything on the wire
Transmit Clock Source	local timing (internal)

SETTING LINE LENGTH

Most T3 values do not need to be set. However, you should set line length to a value in meters by using this command:

> config t3.1 physical-layer line-length <length in meters>

The AP uses the value you specify to select between a short distance (less than or equal to 33 meters) or a long distance (greater than 33 meters) for T3 drive levels.

USING THE PAYLOAD SCRAMBLER

The AP supports a payload scrambler that reduces the pattern sensitivity of multiplexing equipment. The scrambler works by encoding a pseudo-random bit sequence into the data stream. The receiving end removes this code and passes on the original data to the user.

The AP scrambler supports both Cisco and ADC Kentrox. The same software must be running at both ends of the T3 circuit. The default value of scrambling mode is dsx3None, which disables the scrambler function.

To configure the payload scrambler, use this command:

```
> config t3.2 physical-layer scrambling-mode
```

Use this command to derive information on the scrambler's current mode.

```
> show t3.2 physical-layer config
```

CONFIGURING ATM

This section explains how ATM is implemented in the Access Point system (AP), and how to set up and manage ATM interfaces. A brief overview has been provided for those unfamiliar with ATM. This section contains the following topics.

- ATM Overview
- ATM Implementation in the Access Point Systems
- Setting Up ATM Interfaces
- Setting Up ATM Subinterfaces
- Selecting Specific VCs for Traffic Classes

ATM OVERVIEW

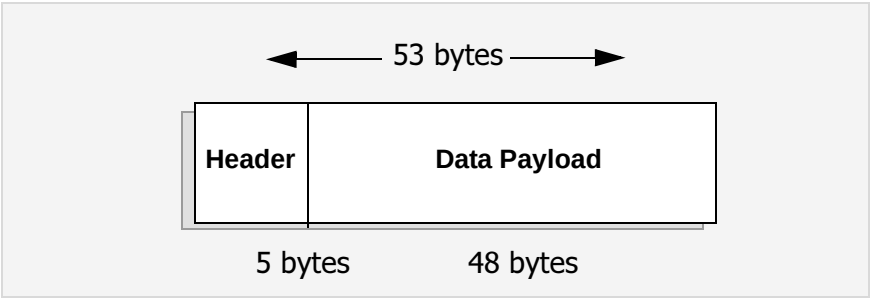
Asynchronous Transfer Mode (ATM) technology is specifically designed for the high-speed transfer of voice, video, and data through private and across public networks. ATM differs from synchronous transfer mode technologies in that it does not use time-division multiplexing techniques to reserve time slots for users. Instead, ATM uses statistical multiplexing techniques to make the network available upon demand, rather than by reservation. This results in

more efficient bandwidth utilization, since slots are only used when needed.

Unlike IP networks, ATM networks are connection-oriented: data traversing the network flows through a set of intervening ATM switches on a fixed path. This path is known as a *virtual connection* (VC). VCs supported by ATM are of two types: switched (SVCs) and permanent (PVCs). ATM segments data into small, fixed-length (53 byte) packets called cells. Cells are transferred through the network via ATM switches. These read the header information in received cells, and switch them to an output interface that connects to the next switch en route to the cell's destination.

From the point of view of the ATM network, the AP is an ATM end station that supports PVCs only.

Figure 11 ATM Cell Structure



THE ATM ADAPTATION LAYER

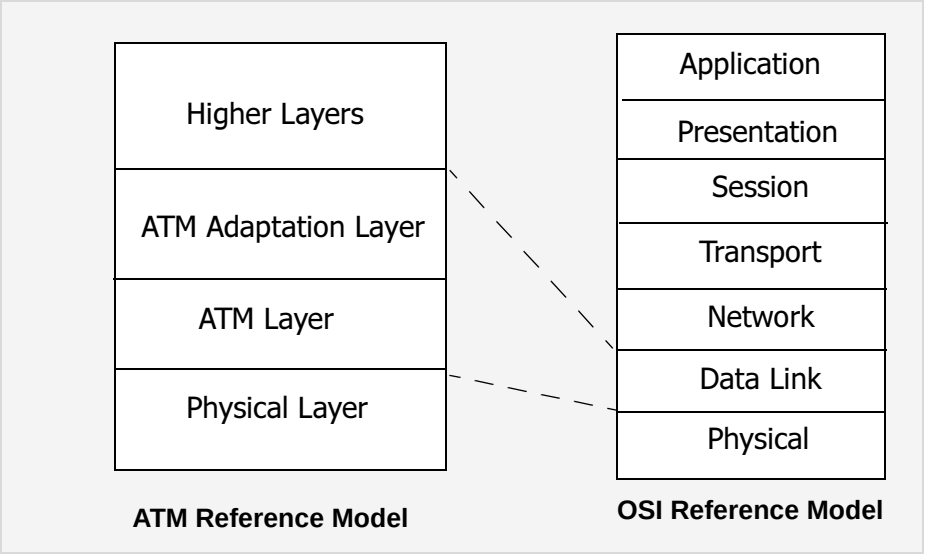
The ATM Adaptation Layer (AAL) translates data from upper-level protocols (like HTTP and Telnet) into the 48-byte packets that form the data area of each cell. Several distinct layer types exist: AAL1, AAL3/4, and AAL5.

The AP ATM supports AAL5, which is normally used for data communications.

ATM AND THE OSI REFERENCE MODEL

In developing the ATM reference model, the ATM Forum adopted an overlay model that does not correspond with the 7 layer model of the OSI network. As a result, ATM is sometimes thought of as a data link protocol. In fact, ATM possesses many of the characteristics of a layer 3 protocol such as IP, including a hierarchical address space and a complex routing protocol.

Figure 12 ATM and OSI Reference Models



How ATM Switching Works

ATM switches map cells received on their input ports to appropriate output ports. This mapping is based on the *virtual path identifier* (VPI) and *virtual circuit identifier* (VCI) contained in each cell's header. Taken together, each VPI/VCI pairing comprises a virtual channel link (VCL). You can view VCLs as separate links in the chain that makes up a virtual circuit. As cells pass from one switch to the next, the switch that receives them maps them to its output ports, sending them on through the network until they reach their destination.

The switch relies on a switching table to alter the VPI/VCI combinations of the cells it receives, and to direct them out through a specific port. For example, if a switch receives a cell with a VPI/VCI of 1/45 on port 1, its table may indicate that this combination should be changed to 2/37 and directed to output port 2.

ATM QUALITY OF SERVICE

ATM implements Quality of Service (QoS) parameters to describe and enforce the traffic flow on virtual circuits. When an ATM end-station connects to the network, it enters into a *contract* based on QoS parameters. The contract specifies the following traffic characteristics.

- Service category (CBR, VBR, UBR)
- Peak bandwidth
- Average bandwidth
- Burst size

ATM devices are designed to adhere to the contract using *traffic shaping* techniques. Essentially, this involves the use of queues to constrain bursty data, limit peak data rate, and smooth jitter. ATM switches can use *traffic policing* to enforce the contract. When traffic exceeds contracted boundaries, a switch can mark cells as discard eligible. When so marked, the cells may be dropped by this or any other switch during periods of congestion.

ATM IMPLEMENTATION IN THE ACCESS POINT SYSTEMS

Figure 13 depicts the relationships between the layers in the ATM stack, and indicates which MIBs have been implemented.

THE CBQ LAYER

Class-Based Queuing (CBQ) is responsible for traffic classification and queuing.

THE ATM-DS3 LAYER

This is the physical layer. The DS-3 MIB (RFC 1407) describes how this layer is implemented.

THE ATM-SONET LAYER

This is the physical layer. The SONET MIB (RFC 1595) describes how this layer is implemented.

THE ATM LAYER

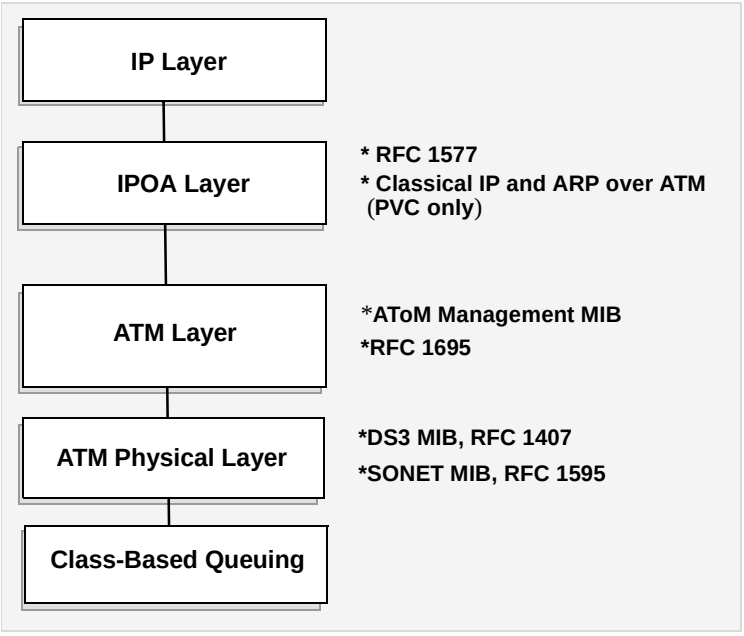
The ATM layer is responsible for the multiplexing of VCs and dispatching traffic to and from various higher layers. The AToM MIB (RFC 1695)

describes how this layer is implemented.

THE IPOA LAYER

This layer implements classical IP over ATM, as described in RFC 1577, including the portion of the MIB that treats PVCs. Also implemented is LLC encapsulation (RFC 1483), and most of the changes defined by the Classic 2 draft RFC and the associated MIB draft RFC.

Figure 13 Layers in the Access Point ATM Implementation



SETTING UP ATM INTERFACES

Setting up ATM on the AP requires you to perform a few steps at each layer of the protocol stack. These steps include:

- Configuring the ATM Physical Interface
- Configuring the ATM Data Link Layer
- Configuring the IPOA Layer

Note that you can perform these steps in any order you like. The order shown here represents a bottom-up approach, beginning at the physical layer (bottom) and progressing up the protocol stack.

RUN STATUS ON THE ACCESS POINT SYSTEM

Before you begin configuring anything, issue the *status* command to find out which components are in the AP's slots.

SYSTEM STACK

ROW		Layer	Status
1	slot.2.1 ATM Driver	enabled	up
2	slot.2.1 cbq.1	enabled	up
3	slot.2.1 atm-ds3.1	enabled	up
4	slot.2.1 atm.1	enabled	up
5	slot.2.1 ipoa.1	enabled	up
6	slot.2.1 ip.1	enabled	up
7	slot.3.1 Ethernet Driver	enabled	down
8	slot.3.1 cbq.2	enabled	down
9	slot.3.1 eth.1	enabled	down
10	slot.3.1 ip.2	disabled	down

Notice that in this partial sample display, only two of the AP's four slots are populated. Slot 2 contains the ATM interface card. The instructions that follow show how to configure the ATM-specific objects (atm-ds3.1, atm-sonet.1, atm.1, and ipoa.1) associated with the interface. You will also have to assign an IP address to the IP instance (ip.1) associated with the interface.

CONFIGURING THE ATM PHYSICAL INTERFACE

To configure the ATM physical interface, perform one of the following tasks:

- Configuring the DS3 Layer
- Configuring the SONET Interface

CONFIGURING THE DS3 LAYER

The DS3 layer shown here is the physical layer of the device. To show characteristics of the ATM DS3 layer, issue this command:

```
> show atm-ds3.1 summary
```

```
                                ATM-DS3.1 SUMMARY

        Description  = ATM DS3 PHY slot: 2 port: 1
               Type  = ds3
              Speed  = 44184000 bits per second
        MAC Address  = 00:00:00:00:00:00
               Mode* = up
              State  = up
  Last State Change  = 1488 (0 hours 0 min 14 secs)
```

In addition to standard interface layer information shown here, you can find more information in the physical-layer subcategory. This subcategory contains information specific to the DS3 MIB (RFC 1407).

SHOWING DS3 PHYSICAL LAYER INFORMATION

To show DS3 physical-layer information, issue this command:

```
> show atm-ds3.1 physical-layer config
```

```
                        ATM-DS3.1 PHYSICAL-LAYER CONFIG

              Line Type* = dsx3CbitParity
            Line Coding* = dsx3B3ZS
              Send Code* = dsx3SendNoCode
             Circuit ID* = Xedia ATM DS3 Interface
        Loopback Config* = dsx3NoLoop
           Clock Source* = internal
           Line Length* = 137 meters
  Status Change Trap Enable* = enabled
```

Table 7 specifies supported DS3 attributes and values.

Table 7 DS3 Attributes and Values

DS3 Attribute	Value
Line Coding	B3ZS only
Line Type	C Bit parity only
Line Length	Set to length in meters. See “Setting Line Length” on Page 124.
Send Code	Not supported
Circuit ID	An administrative read-write string
Loopback Config	The following are supported: • loopback everything on the wire (dsx3LineLoop) • loopback everything sent (dsx3InwardLoop) • loopback disabled (dsx3NoLoop)
Transmit Clock Source	The following are supported: • local timing (internal) • loop timing (external)

SETTING LINE LENGTH

Most DS3 values do not need to be set. However, you should set line length to a value in meters by using this command:

```
> config atm-ds3.1 physical-layer line-length <length in meters>
```

NOTE *Some ATM equipment will not function properly if the actual line length differs greatly from the assigned value. This makes it important to set the line length reasonably accurately. Set the value within 20% of the actual value.*

The AP uses the value you specify to select between a short distance and a long distance (under or over 68 meters, up to a maximum of 137) for DS3 drive levels. Distances that exceed 137 meters are rejected.

The sections that follow show how to access additional information on other DS3 attributes.

To Show ATM DS3 Status

You can use the following command to derive information on the DS3 configuration.

```
> show atm-ds3.1 physical-layer status
```

Table 8 DS3 Status Attributes and Values

DS3 Status Attribute	Value
line status	A bitstring that gives any current alarm indications, or noAlarm.
loopback status	A bitstring that represents the loopback mode.
valid and invalid intervals	Statistics are accumulated every 15 minutes (e.g., each interval). There are 96 intervals in 24 hours. • The current 15 minute interval is given by the "seconds in current." • Valid intervals gives the number of prior intervals accumulated, up to 96. • Invalid intervals = 96 - number of valid intervals.

To Show the Current Interval

```
> show atm-ds3.1 physical-layer interval-current
```

The current interval includes a set of statistics for the current 15 minute interval. The DS3 MIB describes these statistics in detail.

When the current interval hits 15 minutes, it becomes interval-current.1 and all other prior intervals get shifted down accordingly.

To Show Prior Intervals

```
> show atm-ds3.1 physical-layer interval-history.1
```

Use this command to get statistics for prior intervals. The largest specifiable interval instance is interval-history.96, which equates to 24 hours ago. The interval-history.1 instance is the most recent 15 minute interval. Every 15 minutes a new interval is added, and the prior intervals are shifted down accordingly.

TO SHOW THE SUM OF PRIOR INTERVALS

```
> show atm-ds3.1 physical-layer interval-totals
```

The total interval is the sum of all the prior intervals. This command provides a convenient way to see statistics on all events over the last 24 hours (or since the AP has been on-line). The local line status is a bitwise "OR" of all alarm indications over all prior intervals.

CONFIGURING THE SONET INTERFACE

The SONET layer shown here is the physical layer of the device. The AP ATM supports OC-3. Note that either multimode fiber (MMF) or single-mode fiber (SMF) cabling is supported with different models of the ATM card.

To show characteristics of the ATM SONET interface, issue this command:

```
> show atm-sonet.1 summary
```

```
                              ATM-SONET.1 SUMMARY

      Description  = ATM SONET PHY slot: 1 port: 1
             Type  = sonet
            Speed  = 149760000 bits per second
      MAC Address  = 00:00:00:00:00:00
             Mode* = up
            State  = up
Last State Change  = 1667400 (4 hours 37 min 54 secs)
```

In addition to standard interface layer information shown here, you can find more information in the physical-layer subcategory. This subcategory contains information specific to the SONET MIB (RFC 1595).

SHOWING SONET PHYSICAL LAYER INFORMATION

To show physical-layer subcategory information, issue this command:

```
> show atm-sonet.1 physical-layer config
```

```
                          ATM-SONET.1 PHYSICAL-LAYER CONFIG

    Medium Type* = sonet
      Circuit ID* = Xedia ATM SONET Interface
   Clock Source* = internal
Loopback Config* = sonetNoLoop
```

Table 9 specifies supported SONET attributes and values.

Table 9 SONET Attributes and Values

SONET Attribute	Value
Circuit ID	An administrative read-write string
Transmit Clock Source	The following are supported: • local timing (internal) • loop timing (external)
Loopback Config	The following are supported: • loopback everything on the wire (sonetLineLoop) • loopback everything sent (sonetInwardLoop) • loopback disabled (sonetNoLoop)
Medium Type	The following are supported: • sonet • sdh

The sections that follow show how to access additional information on other SONET attributes.

To Show ATM SONET Status

You can use the following command to derive information on the SONET configuration.

```
> show atm-sonet.1 physical-layer status
```

Table 10 SONET Status Attributes and Values

SONET Status Attribute	Value
section, line, or path status	A bitstring that gives any current alarm indications, or noAlarm.
loopback status	A bitstring that represents the loopback mode.
valid and invalid intervals	Statistics are accumulated every 15 minutes (e.g., each interval). There are 96 intervals in 24 hours. • The current 15 minute interval is given by the “seconds in current.” • Valid intervals gives the number of prior intervals accumulated, up to 96. • Invalid intervals = 96 - number of valid intervals.

TO SHOW THE CURRENT INTERVAL

```
> show atm-sonet.1 physical-layer interval-line-current
> show atm-sonet.1 physical-layer interval-path-current
> show atm-sonet.1 physical-layer interval-section-current
```

The current interval includes a set of statistics for the current 15 minute interval for the line, the path, or the section. The SONET MIB describes these statistics in detail.

When the current interval hits 15 minutes, it becomes interval-*-current.1 (where * is line, path, or section) and all other prior intervals get shifted down accordingly.

TO SHOW PRIOR INTERVALS

```
> show atm-sonet.1 physical-layer interval-line-history.1
> show atm-sonet.1 physical-layer interval-path-history.1
> show atm-sonet.1 physical-layer interval-section-history.1
```

Use this command to get statistics for prior intervals for the line, the path, or the section. The largest specifiable interval instance is interval-*-history.96 (where * is line, path, or section), which equates to 24 hours ago. The interval-*-history.1 instance is the most recent 15 minute interval. Every 15 minutes a new interval is added, and the prior intervals are shifted down accordingly.

CONFIGURING THE ATM DATA LINK LAYER

Configuring the data link layer involves the following actions:

- Adding and configuring virtual channel links (VCLs) for use by each ATM instance
- Adding traffic descriptors
- Configuring the ATM interface

After configuring VCLs and adding traffic descriptors, configure the ATM interface as explained in “Configuring the ATM Interface” on Page 136.

ADDING AND CONFIGURING VCLS

Each VC has a VPI (virtual path identifier) and VCI (virtual channel identifier) associated with it. The AP supports up to 4 VPIs in the range 0-3. For each VPI, you can add up to 1023 VCIs.

Each VPI/VCI is a distinct virtual channel, designed to carry a particular type of service. In the aggregate, each ATM card supports approximately 4,000 virtual channel links (4x1023). Each of these is associated with a locally unique VPI/VCI pair called a VCL or *virtual channel link*.

RESERVED VCIS

VCIs in the range 0-31 are reserved for special uses, and cannot be assigned. Hence in the sample commands that follow, the lowest VCI used is 32.

ADDING VCLS

These commands add four virtual channel links (VCLs) to the ATM card in slot 1 (atm.1). Notice that two of VCLs use VPI 0, and two use VPI 1.

```
> add atm.1 vcl vpi.0 vci.32
> add atm.1 vcl vpi.0 vci.40
> add atm.1 vcl vpi.1 vci.35
> add atm.1 vcl vpi.1 vci.36
```



Tip: Adding Virtual Circuits

Rather than adding VCs directly as shown here, you can also add them automatically during configuration of the IPOA layer. This may save you time in the configuration process. For details, see "Adding VCLs at the IPOA Layer" on Page 138.

After adding VCs, you can issue this command to show summary information:

```
> show atm.1 vcl all-vcls summary
```

```
                                ATM.1 VCL ALL-VCLS SUMMARY
      INTERFACE|  VPI|   VCI|              Mode|             State|  Last Change|
               |     |      | Rx Traffic Descr| Tx Traffic Descr|   Row Status|
---------------|-----|------|-----------------|-----------------|-------------|
          atm.1|    0|   100|               up|               up|500 (0 hours>|
               |     |      |                1|                1|       active|
```

ADDING TRAFFIC DESCRIPTORS TO VIRTUAL CHANNEL LINKS

Assign a traffic descriptor to each VCL you have added. Traffic descriptor 1 (UBR) is the default, and is assigned to each PVC as it is created. To change this, use the *config* command as shown here.

```
> config atm.<instance> vcl vpi.<instance> vci.<instance> tx-traffic-descr 5/
  rx-traffic-descr 5
```

Notice that the tx (transmit) and rx (receive) descriptor values given here are the same (5). While this need not be the case, the tx and rx values chosen must specify the same QoS values (see “Defining Traffic Descriptors” on Page 132).

The sections that follow provide general information on traffic descriptors and ATM service categories, and specific information on the predefined traffic descriptors provided by the AP. Review this information before assigning descriptors to VCLs.

ABOUT TRAFFIC DESCRIPTORS AND SERVICE CATEGORIES

Traffic descriptors are stored descriptions that determine the nature of traffic on the PVCs to which they are assigned. You can either define your own descriptors, or use one of the predefined descriptors as explained below (see “Using Predefined Traffic Descriptors” on Page 132).

If you elect to define your own descriptors, you must do this before referencing them in VCL entries. Each VC you create requires that a traffic descriptor be associated with it. If you do not add a descriptor, the AP assigns the PVC the default descriptor (1).

The *service categories* supported by ATM offer different service guarantees. Each service category is more or less suitable, depending on the type of appli-

cation in use. Basically, users of the network (ATM end systems) get what they pay for: a guaranteed level of service requires the end system to conform to an agreed traffic contract. However, a “best effort” service requires no conformity on the end system’s part.

The AP supports QoS for three service categories. These are:

- UBR - Unspecified Bit Rate
- VBR - Variable Bit Rate
- CBR - Constant Bit Rate

UBR

UBR is a “best effort” service. The network does not guarantee that it will deliver the traffic offered. The network also does not require the sender to conform to a precise traffic pattern.

UBR best matches the classic IP traffic model of “best effort” service. It comes in two forms: unspecified (anything goes up to the physical line rate) and limited (peak rate limited to Param 1 cells per second).

CBR AND VBR

CBR and VBR are both rate-limited classes that offer service guarantees. The network requires end systems, such as the AP, to conform to agreed traffic patterns (rates). In exchange, the network commits to delivering the traffic offered.

CBR is intended for constant streams of data having a well-defined constant rate. The traffic descriptor gives that rate, in cells per second, in “Param 1.”

VBR circuits allow an average and a peak bandwidth, plus some amount of burst bandwidth per second. *VBR* is for variable data streams (for example, compressed video) where there is a well-defined average and a defined peak rate. Param 1 is the peak rate, Param 2 is the average rate (both in cells per second) and Param 3 is the maximum burst size (in cells).

USING PREDEFINED TRAFFIC DESCRIPTORS

The AP comes with four preconfigured traffic descriptors. To display these descriptors, use the *show atm traffic-descriptors* command:

ATM TRAFFIC-DESCRIPTORS

INDEX	QOS Param 4	Type Description	Param 1 Row Status	Param 2	Param 3
1	ubr 0	atmNoTrafficDescriptor UBR, no rate limit	0 active	0	0
2	ubr 0	atmNoClpNoScr UBR: max 10Mbps	26042 active	0	0
3	cbr 0	atmNoClpNoScr CBR: max 20Mbps	52084 active	0	0
4	vbrConn> 0	atmNoClpScr VBR: sust 5Mbps peak 1>	26042 active	13021	200

You can assign any one of these traffic descriptors to a PVC, or create one or more new descriptors. To assign one of the predefined descriptors, use the *config* command as shown here.

```
> config atm.<instance> vcl vpi.<instance> vci.<instance> tx-traffic-descr
  <index> rx-traffic-descr <index>
```

For example, use the following command to assign the predefined VBR traffic descriptor:

```
> config atm.1 vcl vpi.1 vci.38 tx-traffic-descr 4 rx-traffic-descr 4
```

Keep in mind that the AP automatically assigns each VCL a traffic descriptor of 1 as they are created unless you specify otherwise.

DEFINING TRAFFIC DESCRIPTORS

Defining a traffic descriptor requires you to supply a range of information, including:

- A specific instance number for the descriptor
- The QoS (Quality of Service) type. This is usually called the Service Category, but the AToM MIB calls the parameter QoS.
- Average, peak, and burst rates (in the case of VBR services)

An example of a command that specifies a new traffic descriptor (traffic-descriptor.5) is shown here.

```
> config atm traffic-descriptor.5 qos vbrAudiovideo param-1 10000 \
  param-2 2500 param-3 300 row-status createAndWait
```

See the following two sections for information about traffic parameters and circuit characteristics.

SERVICE CATEGORIES AND DESCRIPTOR TYPES

ATM supports a number of traffic descriptor types. These can be associated with the service category (CBR, VBR or UBR) and the specific characteristics of service instances when defining traffic descriptors.

In determining which traffic descriptor type to use, rely on the circuit-specific information provided by the network operator or service provider.

Table 11 Traffic Descriptors and Service Categories

Service Category	Characteristics	Parameters
VBR.1	• Does not matter how the CLP bit is set • Traffic sent is subject to specified PCR, SCR, and MBS limits	type = atmNoClpScr qos = vbrAudioVideo, vbrConnData, or vbrConnlessData param-1 = peak cell rate param-2 = sustained cell rate param-3 = max burst size param-4 = CDVT
VBR.2	• Traffic sent with the CLP bit not set is subject to specified SCR and MBS limits • Traffic sent with the CLP bit set is subject to specified PCR limit only	type = atmClpNoTaggingScr qos=vbrAudioVideo, vbrConnData, or vbrConnlessData param-1 = peak cell rate param-2 = sustained cell rate param-3 = max burst size param-4 = CDVT

Service Category	Characteristics	Parameters
VBR.3	All traffic sent is subject to specified PCR limit only **NOTE:** VBR.3 guarantees low-loss service when traffic is sent at the rate defined by PCR, SCR, and MBS. However, traffic can be sent at faster rates. In this case, the network (usually the ATM switch) sets the CLP bit to 1 for the excess traffic. Unlike VBR.1 and VBR.2, the excess traffic is not discarded immediately so VBR.3 provides the benefit of service at PCR for this excess traffic if the network is lightly loaded.	type = atmClpTaggingScr qos = vbrAudioVideo, vbrConnData, or vbrConnlessData param-1 = peak cell rate param-2 = sustained cell rate param-3 = max burst size param-4 = CDVT
UBR	• No rate parameters in use • No rate limits • Default descriptor for AP	type=atmNoTrafficDescriptor qos=ubr
UBR	• Limited rate	type=atmNoClpNoScr qos=ubr param-1=peak cell rate param-2 = CDVT
CBR	• A constant flow of data at the rate set by param-1 (PCR)	type=atmNoClpNoScr qos=cbr param-1=peak cell rate param-2 = CDVT

NOTES ON TRAFFIC PARAMETERS

The AP performs traffic scheduling based on minimum (1/8510 of line rate) and maximum (line rate) cell rates. As a result, you can specify any rate, but it will be rounded DOWN to the next lower actual cell rate value (an 8510th fraction of the line rate).

There is an exception to this rule. If you specify a cell rate that is less than the minimum cell rate (1/8510 of line rate), the AP rounds this rate UP to that value, rather than down to zero.

- The sustained cell rate (SCR) specified for a VC must be less than the peak cell rate (PCR).

- For any traffic descriptor, the rates must be between the minimum and the maximum supported cell rate for that line. For DS3, the minimum cell rate is 13 cells per second and line rate is 105,509 cps. For OC-3, the minimum cell rate is 42 cps and line rate is 353,207 cps.

CELL DELAY VARIATION TOLERANCE

Cell delay variation tolerance (CDVT) is one of the cell scheduling parameters used for cell rate enforcement (Usage Parameter Control or UPC). By default, the value for CDVT is 1000 (that is, 100 microseconds) or one cell time, whichever value is larger. Cell time is the time between cells transmitted at the peak cell rate. For example, if PCR is 1000000, then one cell time is 1 microsecond.

You can specify CDVT along with the familiar PCR, SCR, and MBS parameters. All these parameters are described in Table 11 for particular service categories. For example, *param-2* is used to specify the CDVT for UBR (with limited rate) or CBR. For VBR, *param-4* is the CDVT.

In all cases, the AP specifies CDVT in units of 100 nanoseconds (tenth of a microsecond).

NOTE

Some ATM switch implementations might specify CDVT in different units. For example, some switches might use units of 1 microsecond instead of 100 nanoseconds. If you have such a switch, make sure that you make the proper conversion when you specify CDVT on the AP and the switch. For example, 50 microseconds would match a value of 500 on the AP.

CLP BIT SETTINGS

The setting of the CLP bit is under control of CBQ, and is not related to any service category. You can set the CLP bit for specific traffic classes to mark these classes for discard during periods of congestion. The setting of the CLP bit is important when the type parameter for the VBR service categories is atmClpNoTaggingScr or atmClpTaggingScr.

For this mechanism to work as intended, you must configure CBQ properly in relation to the IPOA layer. For information on use of CBQ to control VC selection, see “Selecting Specific VCs for Traffic Classes” on Page 141. See “Setting the CLP Bit” on Page 142 for information on setting the CLP bit for a traffic class.

CONFIGURING THE ATM INTERFACE

The AP functions as an ATM end-station, or User-Network Interface (UNI), instead of as an ATM switch. Its settings must match those of the ATM switch for a working connection to exist.

SHOW INTERFACE CHARACTERISTICS

Use the *show* command to examine characteristics of the interface. Parameters marked with an asterisk are settable.

```
> show atm.1 config

                                    ATM.1 CONFIG

          MTU* = 8000 bytes                  VPI Bits* = 2
  Up Down Trap* = enabled                    VCI Bits* = 10
        Alias* = ""                 Remote IP Address* = 0.0.0.0
     Max VPCs* = 0              Remote Interface Name* = unknown
     Max VCCs* = 4096                    Empty Cells* = idle
         VPCs  = 0                        Cell Align* = hec
         VCCs  = 0
```

Use this command to configure the interface:

```
> config atm.1 cell-align hec empty-cells idle
```

These are the standard settings for the interface. The mode of the interface defaults to up. Basic information on settable parameters is given below.

BASIC PARAMETERS

The only configuration parameters you need to be concerned with are mode (set to up), cell-align (normally set to hec) and empty-cells (normally set to idle).

EMPTY-CELLS

Specifies the type of cells the interface sends as a filler when there is no data to send.

CELL-ALIGN

Specifies the cell alignment mechanism used by the DS3 framer. In HEC mode, the framer uses the HEC alignment method. 53-octet cells are expected. In PLCP mode, PLCP framing overhead is used, resulting in 57-octet cells.

You may need to alter these settings, depending on what ATM equipment exists on the other end of the physical link. For example, many switches

default cell-align to PLCP rather than HEC, however they generally recommend HEC alignment.

CONFIGURING THE IPOA LAYER

Begin by entering the *show* summary command for the IPOA layer as shown here.

```
> show ipoa.1 summary
```

```
                                IPOA.1 SUMMARY

      Description  = IP over ATM slot: 1 port: 1
             Type  = other
            Speed  = 0 bits per second
      MAC Address  = 00:00:00:00:00:00
             Mode* = up
            State  = down
Last State Change  = 323 (0 hours 0 min 3 secs)
```

Use these commands to bind the PVCs you have configured with the IPOA layer in its slot.

```
> add ipoa.1 configured-pvc vpi.0 vci.32
```

```
> add ipoa.1 configured-pvc vpi.0 vci.40
```

```
> add ipoa.1 configured-pvc vpi.1 vci.35
```

```
> add ipoa.1 configured-pvc vpi.1 vci.36
```

When you finish binding the PVCs, issue the following command to check the status of the IPOA layer:

```
> show ipoa.1 configured-pvc all-pvcs summary
```

```
                    IPOA.1 CONFIGURED-PVC ALL-PVCS SUMMARY

      INTERFACE| VPI|   VCI| Row Status|
---------------|----|------|-----------|
         ipoa.1|   0|   100|     active|
```

REQUESTING VCS FOR EACH NEIGHBOR

You should request that the network administrator of the ATM switch set up no more than one VC for each neighboring ATM router. Setting up multiple VCs for a router is not useful because only one connection will be used. The rest will only consume system resources.

ADDING VCLS AT THE IPOA LAYER

You can add VCLs while setting up the IPOA layer, rather than adding them at the ATM layer as explained earlier (see "Adding and Configuring VCLs" on Page 129) by specifying the configured-pvc parameter with the *add* command. When you add a VCL, it will automatically be added with a traffic descriptor of 1 (UBR).

You can then modify this VCL using the *config* command. However, removing an IPOA instance does not remove the VCL associated with it. To remove the IPOA instance and the associated VCL, you must use two *remove* commands as follows:

```
>remove ipoa.1 configured-pvc vpi.0 vci.45

>remove atm.1 vcl vpi.0 vpi.45
```

CONFIGURING THE ARP TABLE SIZE

You can configure the size of the ARP table for ATM (IPOA). When you change the ARP table size, it takes effect immediately. All ARP entries are removed, but Inverse ARP is restarted to re-learn the entries. You should not make the ARP table size smaller than the number of neighbors or you will not be able to communicate with some of them. The allowed range for the ARP table size is 1 to 5000. The default is 1000.

Changing the ARP table size can save memory. For example, typically you'll have far fewer than 1000 ATM neighbors on an interface. If you want to use a significant number of ATM subinterfaces, each one of these will have a separate IPOA instance above it. By default, each of these has a separate 1000 entry ARP table. Usually, only a few entries are needed in each of these tables. You can therefore save a lot of AP memory by using a command such as:

```
> config ipoa.1 arp-config arp-table-size 10
```

VALIDATING THE IPOA CONNECTION

After you set up IPOA, the IPOA layer uses the Inverse ARP protocol to validate that it can communicate with its counterpart on the other end of the PVC. Issue this command to see the results of this validation process:

```
> show ipoa.<instance> vcs summary
```

If communication has been established, you will see an entry listing the VPI/VCI and the remote IP address.

ADDING IP ADDRESSES TO EACH ATM INTERFACE

Add an address to the IP instance associated with each slot as follows:

```
> add ip.<instance> address.<address in dotted quad notation>
```

SETTING UP ATM SUBINTERFACES

Access Point systems support the creation of subinterfaces for ATM data links. Subinterfaces provide discrete instances of the connectivity provided by the main or *integral* interface type. Integral interfaces and subinterfaces provide identical functionality. You can use them together to create independent instances of the connectivity provided by the data link. This provides the following benefits:

- On Non-broadcast Multiple Access (NBMA) networks like ATM, subnets that are not fully meshed can be broken down into fully-meshed subinterfaces. This aids in configuring bridging.
- In a routed configuration, you can assign ATM circuits to specific IP address forwarders within the AP so you can run different routing protocols on each subinterface.
- Assigning different ATM circuits to specific subinterfaces also allows you to enable and use split horizon management. This is not ordinarily possible on non-fully-meshed NBMA networks.

While integral interfaces have one index, subinterfaces have two indices. For instance, atm.1 is an integral interface, but atm.1.1 and atm.1.2 are subinterfaces. Each subinterface uses a separate IP forwarder. In the case of ATM, a separate instance of IPOA and a separate IP forwarder are required.

CONFIGURING ATM SUBINTERFACES

You can add ATM subinterfaces at the virtual control layer of the ATM stack. The goal is to separate certain circuits for handling by different subinterfaces (or the integral interface). Like the integral interface, each ATM subinterface requires an instance of IPOA and an IP instance stacked above it. To add an ATM subinterface, proceed as in this example:

```
> stack slot.1.1 cbq.1 atm-ds3.1 atm.1 atm.1.1 ipoa.2 ip.8
```

The following *status* command display shows the result of adding this sub-

interface, together with IPOA and IP instances.

SYSTEM STACK

ROW			Layer	Status
1	slot.1.1	ATM Driver	enabled	up
2	slot.1.1	cbq.1	enabled	up
3	slot.1.1	atm-ds3.1	enabled	up
4	slot.1.1	atm.1	enabled	up
5	slot.1.1	ipoa.1	enabled	up
6	slot.1.1	ip.1	enabled	up
7	slot.1.1	atm.1.1	enabled	up
8	slot.1.1	ipoa.2	enabled	testing
9	slot.1.1	ip.8	enabled	down
10	slot.2.1	Ethernet Driver	enabled	up
11	slot.2.1	cbq.2	enabled	up
12	slot.2.1	eth.1	enabled	up
13	slot.2.1	ip.2	enabled	up
14	slot.3.1	Ethernet Driver	enabled	up
15	slot.3.1	cbq.3	enabled	up
16	slot.3.1	eth.2	enabled	up
17	slot.3.1	ip.3	enabled	up
18	virtual	Virtual Tunnel Transport	enabled	down
19	virtual	cbq.4	enabled	down
20	virtual	ipsec.1	enabled	down
21	virtual	ip.6	enabled	down

ADDING CIRCUITS TO SUBINTERFACES

Once you have added subinterfaces, you can add circuits to them just as you would to the integral interface. There are two ways to add circuits:

- You can add circuits directly to the subinterface.
- You can add circuits implicitly, by adding a configured PVC to the IPOA layer stacked above the subinterface.

For example, this command adds circuit VPI 1 VCI 50 to subinterface atm.1.1:

```
> add atm.1.1 vcl vpi.1 vci.50
```

The following command adds a circuit implicitly to the subinterface. The circuit is added at the IPOA layer:

```
> add ipoa.2 configured-pvc vpi.2 vci.35
```

Adding circuits in this manner causes the specified IPOA layer to send a message to the virtual circuit subinterface stacked below it. This adds the circuit to the appropriate subinterface.

SELECTING SPECIFIC VCS FOR TRAFFIC CLASSES

You can use CBQ to map traffic classes to specific ATM VCs. Doing so lets you map CBQ's traffic shaping capabilities with the QoS capabilities inherent in ATM networks.

This section explains how to map CBQ classes to specific VCs. Doing this only makes sense, of course, if more than one VC — with different QoS characteristics — exists between the AP end station and an ATM neighbor. To make use of this capability, be sure to request the network administrator of the neighboring ATM switch to set up as many VCs as you need. Then follow the procedure outlined here.

VC mapping capability works in the same way for ATM integral interfaces and subinterfaces. However, VC selection can only be performed on transmit. Do not implement VC selection on traffic classes that reside within CBQ input trees.

ADD A CBQ INSTANCE OVER THE IPOA LAYER

First, use the *stack* command to add an instance of CBQ between each IPOA interface and the IP forwarder as shown here.

```
> stack slot.2.1 atm-sonet.1 atm.1 ipoa.1 cbq.new ip.1
```

Adding a CBQ instance in this location (as an uplink from IPOA) enables CBQ to classify packets before sending them out on a VC.

CONFIGURING CBQ TO SELECT A VC

As an ATM end station, the AP uses traffic descriptors to specify the nature of the traffic (such as service category and cell rate) on VCs.

To select a VC for transmission of a class, map the class to a specific ATM traffic descriptor by configuring the class with a *datalink-traffic-index* parameter. In this command, the class named bulkmail is configured to map to a VC with a traffic descriptor of 6.

```
> config cbq.1 traffic-class.bulkmail parent root-output-tree \
  bandwidth-allocation 10000000 datalink-traffic-index 6
```

If a class definition does not contain a *datalink-traffic-index* or there is no CBQ instance above IPOA in the stack, the AP forwards packets on the VC with the lowest traffic descriptor index.

IF A MATCHING VC DOES NOT EXIST...

In some cases, the *datalink-traffic-index* value incorporated in a class definition does not map to any VC. In this event, the AP checks for another parameter called *datalink-traffic-flags* to see whether to discard the packet, or forward it on another VC.

DISCARDING THE PACKET

To discard packets when the *datalink-traffic-index* value does not match any VC, set the *datalink-traffic-flags* value to 2. This example configures the AP to forward packets matching bulkmail on the VC with a traffic descriptor of 6. If a VC with this descriptor is not found, the AP drops the packet.

```
> config cbq.5 traffic-class.bulkmail datalink-traffic-index 6 \
  datalink-traffic-flags 2
```

If you do not specifically mark a class for discard, the AP forwards packets in this class on the VC with the lowest traffic descriptor index (usually, on a VC with an index of 1).

SETTING THE CLP BIT

You can also use the *datalink-traffic-flags* parameter to set the CLP (Cell Loss Priority) bit for specific traffic classes. If the CLP bit is set, the ATM network considers this data as eligible for discard in case of congestion. For this reason, it makes sense to set the CLP bit for low priority (hence, best effort) classes.

To configure the *datalink-traffic-flags* parameter to set the CLP bit, use a value of 1. Use a value of 0 to clear the CLP bit. The default setting is 0 (that is, traffic is not eligible for discard).

This example makes the class bulkmail eligible for discard.

```
> config cbq.5 traffic-class.bulkmail datalink-traffic-index 6 \
  datalink-traffic-flags 1
```

SETTING THE DATALINK-TRAFFIC-FLAGS PARAMETER

As described in preceding sections, the *datalink-traffic-flags* parameter is used to determine whether to discard packets in certain situations. This table summarizes the available values.

Value	Meaning
0	Clear CLP bit.
1	Set CLP bit.
2	Discard packets when no matching VC is found for the traffic descriptor index specified by the *datalink-traffic-index* parameter.
3	Set CLP bit and discard packets when no matching VC is found.

A SAMPLE OF VC SELECTION USING DATALINK PARAMETERS

In this example, you want to make sure that some low-priority traffic does not use up all the bandwidth. In this case, you want to set the CLP bit so that the ATM network discards this data when there is congestion and you want to discard packets when no matching VC is found.

To do this, you would set up a traffic descriptor that specifies a limited UBR service. Then, you would define a traffic class and specify this traffic descriptor index as the *datalink-traffic-index* value and 3 as the *datalink-traffic-flags* value using this command.

```
> config cbq.5 traffic-class.junkmail datalink-traffic-index 6 \
  datalink-traffic-flags 3
```

GETTING INFORMATION ON CBQ DATALINK PARAMETERS

To display the CBQ datalink parameter configuration, issue this command:

```
> show cbq.3 traffic-class.foo datalink
```

```
                              CBQ.3 TRAFFIC-CLASS.FOO DATALINK

Datalink Traffic Index* = 4
Datalink Traffic Flags* = 0x00
```

CONFIGURING INTEGRATED SERVICES DIGITAL NETWORK

Integrated Services Digital Network (ISDN) is a set of communications standards that allows a single physical link to carry different types of data (for example, voice, digital network services, and video). You can set up the Access Point system (AP) for permanently connected, demand-based, and backup dial service configurations. The following ISDN BRI configurations are supported:

- Two B channels plus one D channel: 2 x 64 Kbps (precompressed)
- Interoperable Switched 56: 2 x 56 Kbps (precompressed)
- Single and multipoint configurations
- Compatible with data or voice B-channel ISDN switch types
- ISDN dialup and leased line at 64 and 128 Kbps

Connect the AP to the ISDN network with a cable containing #26 gauge (or larger) wire. Use the yellow-coded cable (Part Number 308-0006-001) provided with the AP.

The ISDN U interface module can only be used in the United States and Canada. The ISDN S/T interface module is used everywhere else.

CONFIGURING THE ISDN INTERFACE

To configure ISDN interfaces, you would configure the ISDN integral interface (local interface) and ISDN subinterface (remote interface). You must configure a subinterface for each configured remote number that can be connected to the IP forwarder.

SETTING UP THE ISDN INTERFACE

You configure the ISDN interface (local interface) by providing the directory number, the subaddress, and the switch type.

NOTE *If you are using the ISDN U interface, then you must specify the directory number, the Service Profile Identifier (SPID), and any secondary directory number or SPID as provided by your telephone company.*

You must also specify the switch type.

ADDING A DIRECTORY NUMBER AND SUBADDRESS

You can specify the directory number (calling number) and subaddress (extension) identifying this signaling channel for outgoing messages. They can also be used to specify a specific number for an incoming call to match.

When using the ISDN U interface, you must specify the directory number (and secondary directory number, if any) provided by your telephone company when your service is activated.

To specify the calling number (usually 10 digits):

```
> config isdn.<instance> directory-number <calling number>
```

To specify the secondary directory number:

```
> config isdn.<instance> secondary-directory-number
  <secondary directory number>
```

To specify the subaddress (a string of up to 20 ASCII characters):

```
> config isdn.<instance> subaddress <subaddress>
```

For example, use the following command to indicate that 978-555-1234 x567 is the directory number and extension you are calling from for the isdn.2 instance:

```
> config isdn.2 directory-number 9785551234 subaddress 567
```

For example, use the following command to indicate that 978-555-5678 is the secondary directory number for the isdn.2 instance:

```
> config isdn.2 secondary-directory-number 9785555678
```

CONFIGURING THE SWITCH TYPE

The switch type is the particular switch supported by the ISDN network to which this signaling channel is connected. The switch type indicates these supported switches:

National ISDN-1	ni1
INS, HSD 64, and 128 (Japan)	ins64
Euro-ISDN (CTR3)	etsi
Lucent 5ESS	ess5
Nortel DMS-100	dms100

The default switch type is Euro-ISDN (etsi). To specify another switch type:

```
> config isdn.<instance> switch-type < dms100 | ess5 | etsi | ins64 | ni1 >
```

When using the ISDN U interface, you must specify the switch type. For example, use this command to indicate that you are using a National ISDN-1 switch for the isdn.2 instance:

```
> config isdn.2 switch-type ni1
```

ADDING A SERVICE PROFILE IDENTIFIER

When using the ISDN U interface, you must specify the Service Profile Identifier (SPID) information for this terminal endpoint. Your telephone company provides you with the SPID when your service is activated. To specify the SPID:

```
> config isdn.<instance> spid <SPID>
```

If your telephone company provides you with a secondary SPID, you must also specify that value as follows:

```
> config isdn.<instance> secondary-spid <secondary SPID>
```

In addition to this information, you must specify the directory number (and secondary directory number, if any) provided by your telephone company.

DISPLAYING THE ISDN CONFIGURATION

To display the ISDN configuration:

```
> show isdn.1 config
```

```
                                ISDN.1 CONFIG

            MTU* = 1500 bytes Secondary Directory Number* = 5553000
   Up Down Trap* = enabled                          Spid* = 0555100001
          Alias* = ""                  Secondary Spid* = 0555300001
    Switch Type* = ni1                     Subaddress* = ""
Directory Number* = 5551000                  Loopback* = disabled
```

DISPLAYING THE ISDN VERSION

To display the ISDN subsystem software version:

```
> show isdn-global summary
```

```
                                ISDN-GLOBAL SUMMARY

Version  = 1.0.0.0
```

CONFIGURING ISDN SUBINTERFACES

To configure an ISDN subinterface (remote interface), you must:

- Add a subinterface for each remote directory number that you want to call (peer).
- Configure the IP forwarder.
- Configure the peer to support your particular dial service.

ADDING ISDN SUBINTERFACES

You add an ISDN subinterface and its PPP and IP instances using the *stack* command. For example, the following command adds an ISDN subinterface (isdn.1.1) and new PPP and IP instances over the isdn.1 interface in slot 4.1.

```
> stack slot.4.1 isdn.1 isdn.1.1 ppp.new ip.new
```

If you want to use B-channel aggregation for two B-channels on the same ISDN interface to achieve an aggregate maximum of 128 Kbps, you would add instances of Multilink PPP (PPP MP) using the *stack* command. For example, the first command adds an ISDN subinterface (isdn.1.1) and PPP instance (ppp.1) over the isdn.1interface in slot 4.1 and the second command adds a new ISDN subinterface and PPP and PPP MP instances over the same inter-

face in slot 4.1.

```
> stack slot.4.1 isdn.1 isdn.1.1 ppp.1 ppp-mp.1.new ip.new
```

```
> stack slot.4.1 isdn.1 isdn.1.new ppp.new ppp-mp.1.new
```

Note that you do not need to specify an IP instance for the second PPP MP instance because the AP automatically adds the IP instance included with the first PPP MP instance as all PPP MP instances use the same IP instance. Refer to "Configuring PPP" on Page 168 for more information about PPP and Multi-link PPP.

CONFIGURING THE IP FORWARDER

To configure the IP forwarder, you must add an IP address to the IP instance above the PPP instance or configure the IP instance to ask for an address from the PPP peer.

Add or configure the IP address (includes unnumbered interface) using these commands:

```
> add ip.<instance> address.<IP address> net-mask <network mask>
```

```
> config ip.<instance> address.<IP address> net-mask <network mask>
```

You can also specify that the peer assigns the IP address through IPCP once the connection is up by not specifying the IP address with the address parameter and enabling negotiated addressing with the following command.

```
> config ip.<instance> negotiate-addr enabled
```

Note that the peer device must be able to handle address assignment using IPCP and cannot be another AP. You should also configure a router address on another IP interface. You should also configure a static route to the peer device.

To display whether negotiated addressing is enabled:

```
> show ip.2 config
```

```
                                    IP.2 CONFIG

            MTU* = 1500 bytes
  Up Down Trap* = enabled
          Alias* = ""
Negotiate Addr* = enabled
```

To display how the address has been assigned:

```
> show ip.* address-table advanced

                        IP.2 ADDRESS-TABLE ADVANCED

       INTERFACE|         ADDRESS| Broadcast Address| Router Address| Source|
----------------|-----------------|------------------|---------------|-------|
             ip.2|         0.0.0.0|                 0|          false|intern>|

One or more values were truncated (last char '>').
Use 'show' with complete row identifier for "address.<inst>"
```

If you are negotiating addresses, then the Address field contains an unnumbered address and the Source field indicates internal as shown in the example.

If the peer has assigned the address, the Address field contains the assigned address and the Source field indicates peer-assigned.

CONFIGURING PEERS

You configure the peers to support the appropriate dial service for each ISDN subinterface by specifying parameters that provide such information as directory numbers for the called and calling numbers and timers. The following table lists the parameters and their function. Refer to "Configuring Dial Services" on Page 151 for more information about how to configure specific dial services.

answer-dn	Identifies the peer for an incoming call. If this directory number is unknown or the same as the originate-dn parameter, this value is a null string.
control	Connects (connect) or disconnects (disconnect) the connection immediately.
cug	Identifies the Closed User Group at which the peer is called. If the Closed User Group is undefined for the given media or unused, this value is a null string.
dial-backup	Specifies parameters for the dial backup service, including the interface that should be backed up and the failover and failback times.
failure-delay	Specifies the time in seconds after which call attempts are placed again after a peer has been unreachable after the specified number of retries. A value of zero means a peer is not called again.
idle-timer	Specifies the time in seconds after which the connection is automatically disconnected if it has been inactive. A value of zero means the connection is not automatically taken down due to inactivity.
info	Specifies the Information Transfer Capability (non-data or data mode connections) used when calling this peer. By default, unrestrictedDigital is used.

max-duration	Specifies the maximum call duration in seconds. A value of zero means there is no limit.
min-duration	Specifies the minimum duration of a call in seconds, starting from when the call is connected until it is disconnected.
originate-dn	Identifies the directory number at which the peer is called. It is the 'phone number' included in a D channel call request. This value is switch-specific.
peer-subaddress	Specifies the subaddress at which the peer is called. If the subaddress is undefined for the given media or unused, this value is a null string.
permission	Specifies applicable permissions for calling. Values are: originate, answer, both (default), callback, or none. If callback is used, it rejects the call and then calls back. Note that the originate-dn and answer-dn values can be different numbers.
retries	Specifies the number of calls that can be made to a non-responding directory number. A value of zero means there is no limit.
retry-delay	Specifies the time in seconds between call retries. A value of zero means that call retries can be done immediately.
speed	Specifies the information transfer speed in bits/second when calling this peer. Values are rate56 (56 Kbps) and rate64 (64 Kbps). If different transfer speeds are used, specify the lower speed.

DISPLAYING THE ISDN SUBINTERFACE CONFIGURATION

To display the ISDN subinterface configuration:

```
> show isdn.<instance>.<instance> config
```

```
                              ISDN.1.1 CONFIG

                        MTU* = 1500 bytes Peer Subaddress* = ""
                Up Down Trap* = enabled                Cug* = ""
                       Alias* = ""                   Speed* = default
                 Switch Type* = ni1                   Info* = unrestrictedDigital
           Directory Number* = 5551000          Permission* = both
Secondary Directory Number* = 5553000          Idle Timer* = 60
                        Spid* = 0555100001    Min Duration* = 10
              Secondary Spid* = 0555300001    Max Duration* = 0
                  Subaddress* = ""                 Retries* = 3
                    Loopback* = disabled       Retry Delay* = 5
                Originate Dn* = 5554000      Failure Delay* = 60
                   Answer Dn* = ""                 Control* = noOp
```

DISPLAYING INFORMATION ABOUT THE CURRENT CALL

To display information about the current call specified by the subinterface:

```
> show isdn.<instance>.<instance> current
```

```
                                ISDN.1.1 CURRENT

    Remote Address  = ""                                        Info  = 0
 Remote Subaddress  = ""                                  Out Packets  = 0
              Time  = 0 (0 hours 0 min 0 secs)             Out Bytes  = 0
          Duration  = 121144 (0 hours 20 min 11 secs)    In Packets  = 0
             State  = 0                                     In Bytes  = 0
            Origin  = 0
```

CHANGING THE CALL STATE

To connect immediately to the peer specified by the subinterface:

```
> config isdn.<instance>.<instance> control connect
```

To disconnect the call immediately:

```
> config isdn.<instance>.<instance> control disconnect
```

CONFIGURING DIAL SERVICES

The following dial services are supported for the ISDN subinterfaces:

- Permanently Connected
- Dial on Demand
- Dial Back
- Dial Backup
- Nailed-Up B-Channel

CONFIGURING THE PERMANENTLY CONNECTED SERVICE

You can configure the ISDN interface to initiate a call to a configured called number when the AP boots. This call stays connected as long as the AP has power and the configuration remains the same. In the event of a connection failure, call establishment is attempted automatically.

To configure the peer for permanently connected service:

```
> config isdn.<instance>.<instance> originate-dn <called number> idle-timer
  0 retries 0
```

For example, to set up a permanent connection to the directory number 5552000:

```
> config isdn.2.1 originate-dn 5552000 idle-timer 0 retries 0
```

Remember to configure the IP forwarder associated with this subinterface. (See "Configuring the IP Forwarder" on Page 148.)

CONFIGURING THE DIAL ON DEMAND SERVICE

To configure the ISDN interface for dial on demand, you configure a called number used when the AP forwards traffic. All locally-generated traffic (that is, traffic generated by the AP) except for NTP traffic brings up a connection. All forwarded traffic brings up a connection, unless you configure the AP to not bring up a connection by placing a CBQ instance between the PPP and IP instances. Note that you cannot use SNMP traps to allow or disallow connection establishment when CBQ is classifying traffic.

When the AP receives qualifying traffic, it tries to call and buffers the data until the call is established and all data is allowed to pass over the ISDN connection. The dial on demand service also accepts calls to establish the connection. The dial on demand circuit remains active as long as forwarded traffic matching the call initiation criteria is sent within the configured keepalive time (idle-timer parameter). Note that received traffic does not reset the idle timer.

To support the dial on demand service:

1. Configure the ISDN subinterface with a CBQ instance to specify selection criteria for forwarded traffic that does not bring up a connection. You can add or move the CBQ instance with the *stack* command. Note that the CBQ instance must be placed between the PPP and IP instances.

   ```
   > stack slot.<instance>.<instance> isdn.<instance>
     isdn.<instance>.<instance> ppp.<instance> cbq.new ip.new
   ```

   For example:

   ```
   > stack slot.1.1 isdn.2 isdn.2.2 ppp.1 cbq.1 ip.1
   ```

   Use the *status* command to check that the CBQ instance is in the correct location.

2. Define traffic classes to classify traffic so that forwarded traffic does and

does not bring up the connection using the *datalink-traffic-index* parameter. Note that the traffic class parent has to be root-output-tree because the dial on demand service only works with output classification.

To configure the *datalink-traffic-index* parameter for CBQ:

```
> config cbq.<instance> traffic-class.<name> \
  datalink-traffic-index <number>
```

where <instance> is the CBQ instance over the PPP instance, <name> is the CBQ traffic class name containing the criteria used to classify the ISDN traffic, and <number> is a value from 0 - 65534 (indicating that the connection should not be brought up) or 65535 (indicating that the connection should be brought up).

The following command examples define traffic classes classifying traffic based on destination ports. One traffic class (all-other-default) lets traffic on all destination ports bring up a connection and the other traffic class (test1) does not let any traffic for the NTP destination port bring up a connection.

```
> config cbq.1 traffic-class.all-other-default parent root-output-tree \
  datalink-traffic-index 65535 dest-ports any
```

```
> config cbq.1 traffic-class.test1 parent root-output-tree \
  datalink-traffic-index 77 dest-ports ntp
```

To display the CBQ datalink parameter configuration:

```
> show cbq.<instance> traffic-class.<name> datalink
```

```
                    CBQ.1 TRAFFIC-CLASS.TEST1 DATALINK

Datalink Traffic Index* = 77
Datalink Traffic Flags* = 0x00
```

To display the CBQ flow definition (for example, the configured destination port):

```
> show cbq.<instance> traffic-class.<name> flow-def
```

```
                     CBQ.1 TRAFFIC-CLASS.TEST1 FLOW-DEF

 Src Ip Addresses* = ""          Classification Tos* = ""
Dest Ip Addresses* = ""     Classification TOS Mask* = 0x00
        Src Ports* = ""            Src Domain Names* = ""
       Dest Ports* = ntp       Dest Domain Names* = ""
        Protocols* = ""                    Operator* = and
    Applications* = ""
```

3 Make sure the root-output-tree is active with this command:

```
> config cbq.1 traffic-class.root-output-tree row-status active
```

Check your traffic classes to verify that root-output-tree is the parent and that all these traffic classes are active.

```
> show cbq.1 traffic-classes summary
```

```
                                CBQ.1 TRAFFIC-CLASSES SUMMARY

          INTERFACE|                  CLASS NAME|                 Parent|
                   |                            | Bandwidth Allocation|
                   |                            |                Bounded|
                   |                            |          Max Bandwidth|
                   |                            |             Row Status|
-------------------|----------------------------|-----------------------|
              cbq.1|            all-other-default|      root-output-tree|
                   |                            |                      0|
                   |                            |                  false|
                   |                            |                      0|
                   |                            |                 active|
              cbq.1|              root-input-tree|                  input|
                   |                            |              100000000|
                   |                            |                  false|
                   |                            |                      0|
                   |                            |          notInService|
              cbq.1|             root-output-tree|                 output|
                   |                            |              100000000|
                   |                            |                  false|
                   |                            |                      0|
                   |                            |                 active|
              cbq.1|                       test1|      root-output-tree|
                   |                            |                      0|
                   |                            |                  false|
                   |                            |                      0|
                   |                            |                 active|
```

4 Configure the IP address. (See “Configuring the IP Forwarder” on Page 148.)

5 Configure the static default route for the ISDN subinterface so that all traffic without a defined route in the routing table is sent to the specified next-hop address.

For example, to set up your static default route for a numbered address where the next-hop address is 19.20.21.22:

```
> add ip static-route.0.0.0.0 mask.0.0.0.0 tos.0 next-hop.19.20.21.22
```

To set up your static default route for an unnumbered address (such as the negotiated address) where the next-hop address is the ip.1 interface:

```
> add ip static-route.0.0.0.0 mask.0.0.0.0 tos.0 next-hop.ip.1
```

6 Configure the ISDN subinterface with the called number and its keepalive time:

```
> config isdn.<instance>.<instance> originate-dn <called number>
  idle-timer <seconds>
```

For example, to set up a dial on demand connection to call 5551999 and disconnect it after 45 seconds of inactivity:

```
> config isdn.2.2 originate-dn 5551999 idle-timer 45
```

CONFIGURING THE DIAL BACK SERVICE

You can configure the ISDN interface to dial a particular number when a received call matches as a valid calling number. After authentication, the incoming call is dropped and the configured called number is called if the incoming call matched a valid calling number.

To configure the peer for dial back service:

```
> config isdn.<instance>.<instance> answer-dn <calling number>
  originate-dn <called number> permission callback
```

For example, to set up a connection to call back directory number 5552000 when directory number 5551234 calls:

```
> config isdn.2.3 answer-dn 5551234 originate-dn 5552000 \
  permission callback
```

Remember to configure the IP forwarder associated with this subinterface. (See “Configuring the IP Forwarder” on Page 148.)

CONFIGURING THE DIAL BACKUP SERVICE

You can configure the ISDN interface to back up a traditional WAN or LAN interface. When the primary interface goes down, the backup circuit is established after a period of time elapses. Once established, the routing parameters set up for failover lets traffic flow over the dial backup connection. If the primary interface comes back up, the backup ISDN call is terminated after a period of time elapses and traffic flows back over the primary circuit.

To support the dial backup service:

1 Configure the ISDN subinterface with an IP address. (See “Configuring the IP Forwarder” on Page 148.)

2 Configure the backup parameters for the subinterface and enable the dial

backup mode with this command.

```
> config isdn.<instance>.<instance> dial-backup \
  primary-interface <interface> failover-time <seconds> \
  failback-time <seconds> mode enabled
```

The value for primary-interface can be any interface. If the specified interface is up, the ISDN subinterface is brought down. You can check whether the interface is up with the *status* command. The failover and failback times are specified in seconds.

For example, to set up a dial backup connection for the ip.2 interface that fails over and fails back after 5 seconds, use this command.

```
> config isdn.2.2 dial-backup primary-interface ip.2 failover-time 5 \
  failback-time 5 mode enabled
```

To display the ISDN subinterface backup configuration:

```
> show isdn.<instance>.<instance> dial-backup
```

```
                                    ISDN.2.1 DIAL-BACKUP

Primary Interface* = none
    Failover Time* = 0
    Failback Time* = 0
             Mode* = disabled
```

3 Configure routing to use the ISDN interface for failover based on your network routing protocols. For example, you can use static default routes for the IP instances associated with the interface that you want to back up and the appropriate ISDN subinterface. Set the static default route for the ISDN subinterface with a higher cost value so that it is used for failover.

For example, if your physical interface uses an unnumbered address for its IP instance (ip.1) and the ISDN subinterface has the ip.2 instance, you would use the following commands to set up your static default routes.

```
> add ip static-route.0.0.0.0 mask.0.0.0.0 tos.0 next-hop.ip.1 cost 1
```

```
> add ip static-route.0.0.0.0 mask.0.0.0.0 tos.0 next-hop.ip.2 cost 20
```

CONFIGURING THE NAILED-UP B-CHANNEL SERVICE

You can configure the ISDN interface to support nailed-up B-channels. This call brings up the connection on the specified B-channel (B1, B2, or none). No B-channel is selected as the default.

To configure the peer as a nailed-up B-channel:

```
> config isdn.<instance>.<instance> nailed-up < b1 | b2 | none >
```

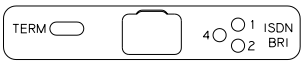
For example, to set up a nailed-up B-channel on the B1 channel:

```
> config isdn.2.1 nailed-up b1
```

Remember to configure the IP forwarder associated with this subinterface.

VERIFYING PROPER OPERATION FOR THE S/T INTERFACE

If your AP and your ISDN S/T interface module are connected correctly, you will see LED indications on the interface module LEDs shown here.



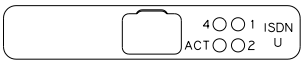
LED 4 is the leftmost LED, LED 1 is the top right LED, and LED 2 is the bottom right LED.

Indication			Meaning
LED 4	**LED 2**	**LED 1**	
Off	Off	On	Inactive
Off	On	Off	Sensing
Off	On	On	Deactivated
On	Off	Off	Awaiting signal
On	Off	On	Identifying input
On	On	Off	Synchronized
On	On	On	Activated
Off	Off	Off	Lost synchronization

The ISDN S/T interface module has a terminating switch. If the AP must be terminated, make sure the TERM switch is toggled to the left.

VERIFYING PROPER OPERATION FOR THE U INTERFACE

If your AP and your ISDN U interface module are connected correctly, you will see LED indications on the interface module LEDs shown here.



LED 4 is the top left LED, ACT is the bottom left LED, LED 1 is the top right LED, and LED 2 is the bottom right LED.

LEDs 1, 2, and 4 indicate the condition of the internal S/T bus as defined in "Verifying Proper Operation for the S/T Interface" on Page 157. The ACT LED has the following indications for the ISDN U interface:

Indication	Meaning
Off	The U interface is inactive if the other LEDs indicate deactivated.
Fast blinking	The U interface is unsynchronized.
Slow blinking	The U interface is synchronized, but the internal S/T bus is unsynchronized.
Solid	All interfaces are fully synchronized if the other LEDs are solid yellow or solid green.

CONFIGURING PACKET OVER SONET

The AP supports Packet over SONET (PoS) connections. PoS sends IP packets, using Frame Relay (default) or PPP, over Synchronous Optical Network (SONET) or Synchronous Digital Hierarchy (SDH) circuits. The AP supports OC-3 with the following transceiver types: multimode fiber (MMF) or single-mode fiber (SMF), intermediate or long range. Using a single connection to a neighboring router, PoS enables you to obtain Frame Relay or PPP service at rates up to 155 Mbps.

CONFIGURING THE SONET PHYSICAL LAYER

The SONET layer shown here is the physical layer of the device. The AP supports OC-3. Note that either multimode fiber (MMF) or single-mode fiber (SMF) cabling is supported. To show characteristics of the SONET interface, use this command:

```
> show sonet.1 summary
```

```
                                SONET.1 SUMMARY

      Description  = SONET PHY media type OC-3 Packet over Sonet single mode in\
                     termediate range, etch 0, eco 2, Cn8236 rev 1, PHY rev 0 s\
                     lot: 1 port: 1
             Type  = sonet
            Speed  = 149760000 bits per second
      MAC Address  = 00:00:00:00:00:00
             Mode* = up
            State  = up
Last State Change  = 563 (0 hours 0 min 5 secs)
```

Use the following command to look at the SONET physical layer configuration:

```
> show sonet.1 physical-layer config
```

```
                            SONET.1 PHYSICAL-LAYER CONFIG

    Medium Type* = sdh
      Circuit ID* = Xedia PoS SONET Interface
    Clock Source* = internal
Loopback Config* = sonetNoLoop
```

Table 12 specifies supported SONET attributes and values.

Table 12 SONET Attributes and Values

SONET Attribute	Value
Medium Type	The following are supported: • sonet • sdh
Circuit ID	An administrative read-write string
Clock Source	The following are supported: • local timing (internal) • loop timing (external)
Loopback Config	The following are supported: • loopback everything on the wire (sonetLineLoop) • loopback everything sent (sonetInwardLoop) • loopback disabled (sonetNoLoop)

The following conditions table provides rules that will help you determine whether to specify internal or external for *Clock Source*.

If:	Set Clock Source to:
Loopback Config is sonetLineLoop	external
Loopback Config is sonetInwardLoop	internal
Data being looped back to the port externally (with loopback connector or a loop in network)	internal
There is no loopback (internal recommended)	internal
Another network recommends or requires use of network-based clocks (transmit clock derived from incoming SONET data stream)	external

NOTE *If you use Clock Source external when the rules indicate that internal must be used, PoS operates improperly. Problems ranging from high error rates to a failure of the link to become operational can occur.*

In a direct connection between two PoS devices (direct fiber connection without the use of a SONET network infrastructure) at least one end of the connection must have *Clock Source* set to internal. That end then provides the master clock if the other end has *Clock Source* set to external.

CONFIGURING THE PACKET OVER SONET INTERFACE

Examine the AP defaults for the PoS interface with the following command. These values should suffice in most instances.

```
> show sonet.<instance> config
```

```
                                    SONET.1 CONFIG

          MTU* = 4470 bytes
 Up Down Trap* = enabled
        Alias* = ""
    CRC Mode* = crc32
```

CRC Mode should be set to crc32, which is the default. An alternate value is crc16. The *CRC Mode* must match at both ends of the PoS connection. If you are using PPP, *CRC Mode* should also match the PPP *CRC Size* setting.

After configuring the physical layer, make sure the data link, Frame Relay, and IP addresses are configured properly for your interface. Use the *status* command to show an appropriately configured stack for PoS:

```
> status
```

```
                                SYSTEM STACK

   ROW|                                                          Layer Status|
------|-------------------------------------------------------------------------|
     1|             slot.1.1 PoS Driver                    enabled      up |
     2|             slot.1.1   cbq.1                       enabled      up |
     3|             slot.1.1    sonet.1                    enabled      up |
     4|             slot.1.1     frame-relay.1             enabled      up |
     5|             slot.1.1      ip.1                     enabled      up |
```

If you want to use PPP instead of the default Frame Relay, use the *stack* command to replace Frame Relay:

```
> stack slot.1.1 cbq.1 sonet.1 ppp.new ip.1
```

```
> status
```

```
                                SYSTEM STACK

   ROW|                                                          Layer Status|
------|-------------------------------------------------------------------------|
     1|             slot.1.1 PoS Driver                    enabled      up |
     2|             slot.1.1   cbq.1                       enabled      up |
     3|             slot.1.1    sonet.1                    enabled      up |
     4|             slot.1.1     ppp.1                     enabled      up |
     5|             slot.1.1      ip.1                     enabled      up |
```

For additional information, see “Configuring PPP” on Page 168.

SONET PATH LABELS

PoS uses SONET path labels to indicate what sort of data is carried on the SONET connection. The PoS standard (RFC 2615) describes two modes: scrambled (required) and non-scrambled (optional) data. The AP supports only scrambled data. PoS uses two values to represent this data:

- 16 hex (scrambled data)
- CF hex (non-scrambled data)

Some PoS implementations use the path label CF even when the data is scrambled. To interoperate with these implementations, the AP accepts the path label CF when carrying scrambled data. The AP does not accept any non-scrambled data.

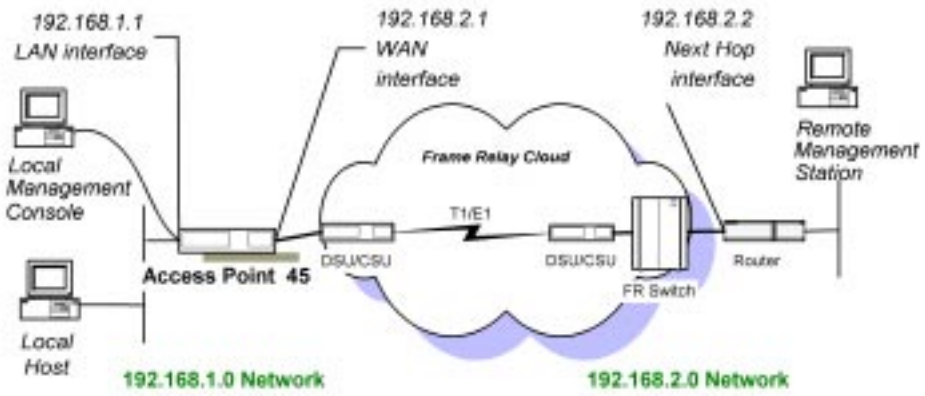
Other path labels indicate data that is not PoS. For example, ATM cell traffic is marked by path label 13 hex. If you plug an ATM device into the PoS interface, you receive the error message:

```
# ADRV  (42) wrong-label                x11/18/1999       13:56:30:001    Warn

      slot.2.1  -  unexpected SONET Path label 0x13
```

CONFIGURING FRAME RELAY

The following section explains how to configure the AP to run Frame Relay over different physical layer connections. Figure 14 shows an Access Point 45 implemented in a Frame Relay network. Refer to this diagram in the instructions that follow.

Figure 14 Access Point 45 in a Frame Relay Network



FRAME RELAY OVER PHYSICAL LAYER CONNECTIONS

The AP supports Frame Relay over these physical layer connections:

- HSSI or MSSI interfaces

  See "Configuring HSSI and MSSI Interfaces" on Page 101 for instructions on setting up HSSI or MSSI interfaces.

- T1 or E1 interfaces

  See "Configuring T1 and E1 Links" on Page 103 for instructions on setting up T1 or E1 interfaces.

- T3 interfaces

  See "Configuring T3 Interfaces" on Page 115 for instructions on setting up T3 interfaces.

ABOUT FRAME RELAY AND PPP SUPPORT

The AP supports both Frame Relay and PPP for wide area connections. Frame Relay networking is the default. A Frame Relay connection comes up automatically upon completion of the physical layer configuration.

Configuring PPP entails replacing the default frame relay connection with an instance of PPP. See Page 168 for detailed instructions on setting up PPP.

Follow the instructions given in the sections below to set up the Frame Relay interface.

SETTING UP THE FRAME RELAY INTERFACE

When setting up Frame Relay, begin by checking the default values to see that they meet your needs. These values should suffice for the majority of cases.

```
> show frame-relay.1 config

                                FRAME-RELAY.1 CONFIG

                      MTU* = 4470 bytes
              Up Down Trap* = enabled
                     Alias* = ""
Connection Mgmt Scheme* = ansiT1-617-D
        Address Format* = q922
        Address Length* = two-octets
      Polling Interval* = 10 seconds
 Full Enquiry Interval* = 6 polling cycles
       Error Threshold* = 3 unanswered Status Enquiries
      Monitored Events* = 4 status polling intervals
     Max Supported VCs* = 32
```

Notice that the default connection management scheme is ANSI T1-617-D (ANNEX-D). You can change any aspect of the Frame Relay configuration using the *config* command.

SETTING UP FRAME RELAY PVCS

After the Frame Relay physical layer comes up, the AP exchanges status enquiry polls with the cooperating Frame Relay switch. The switch responds, and is polled again by the AP. Eventually, the switch sends back a full status enquiry response.

This response contains a listing of Data Link Connection Identifiers (DLCIs); the AP associates a DLCI with its end of the PVC. All of these actions are transparent to the Access Point administrator.

CHECKING CONFIGURED PVCS

You can check to see the PVCs that have been set up by entering this command.

```
> show frame-relay.* dlcis summary
```

The wild card (*) indicates all of the device's Frame Relay interfaces.

If your connection management scheme is set to none, you can configure static DLCIs with the following command:

```
> add frame-relay.1 dlci.<dlci number>
```

VALIDATING FRAME RELAY CONNECTIVITY

From the Access Point console, ping the router on the far end of the Frame Relay cloud (see Figure 14).

```
> ping 192.168.2.2
```

If the ping fails, check the IP address and net mask. Then check that the DLCI in use is correct and active.

SETTING UP FRAME RELAY SUBINTERFACES

Access Point systems support the creation of subinterfaces for Frame Relay data links. Subinterfaces provide discrete instances of the connectivity provided by the main or *integral* interface type. Integral interfaces and subinterfaces provide identical functionality. You can use them together to create independent instances of the connectivity provided by the data link. This provides the following benefits:

- On Non-broadcast Multiple Access (NBMA) networks like Frame Relay, subnets that are not fully meshed can be broken down into fully-meshed subinterfaces. This aids in configuring bridging.
- In a routed configuration, you can assign FR DLCIs to specific IP address forwarders within the AP so you can run different routing protocols on each subinterface.
- Assigning different DLCIs to specific subinterfaces also allows you to enable and use split horizon management. This is not ordinarily possible on non-fully-meshed NBMA networks.

While integral interfaces have one index, subinterfaces have two indices. For instance, frame-relay.1 is an integral interface, but frame-relay.1.1 and frame-relay.1.2 are subinterfaces. Each subinterface uses a separate IP forwarder.

CONFIGURING FRAME RELAY SUBINTERFACES

To create a subinterface, use the *stack* command to place it over the data link. For instance, you can create a frame-relay subinterface beneath the integral frame-relay.1 instance using this command:

```
> stack slot.1.1 cbq.2 hssi.1 frame-relay.1 frame-relay.1.new
```

The subinterface created with frame-relay.1.new is stacked above frame-relay.1. As mentioned, this subinterface requires an additional IP forwarder, which can be either numbered or unnumbered. To add the forwarder, proceed as follows:

```
> stack slot.1.1 cbq.2 hssi.1 frame-relay.1 frame-relay.1.new ip.new
```

The keyword “new” causes the AP to supply the next available instance of the object added with *stack*.

ASSIGNING DLCIS TO SUBINTERFACES

The integral frame relay interface and subinterfaces are differentiated by their support of different DLCIs. Separating DLCIs in this manner makes it possible to manage each DLCI independently.

For example, you can set up the integral interface frame-relay.1 to support only DLCIs 100 through 110 while the subinterface frame-relay.1.1 can support DLCIs 111, 112, and so on.

You can add DLCIs to each subinterface in this manner:

```
> add frame-relay.1.1 dlci.111
```

```
> add frame-relay.1.2 dlci.112
```

Use the same syntax to add DLCIs to the integral frame-relay interface:

```
> add frame-relay.1 dlci.110
```

Note that each DLCI number is unique, and each one is associated with a particular instance of Frame Relay. Packets received by the AP on a DLCI configured on a subinterface will be forwarded to the IP instance stacked above it.

SELECTING SPECIFIC DLCIS FOR TRAFFIC CLASSES

You can use CBQ to map traffic classes to specific Frame Relay DLCIs. This provides you with enhanced control over Frame Relay packets. This mapping capability works in the same way for Frame Relay integral interfaces and sub interfaces. However, DLCI selection can only be performed on transmit. Do not implement DLCI selection on traffic classes that reside within CBQ input trees.

ADD A CBQ INSTANCE OVER THE FR LAYER

First, use the *stack* command to add an instance of CBQ between each FR interface and the IP forwarder as shown here.

```
> stack slot.3.1 hssi.1 frame-relay.1 cbq.3 ip.3
```

Adding a CBQ instance in this location (as an uplink from Frame Relay) enables CBQ to classify packets before sending them out on a DLCI.

CONFIGURING CBQ TO SELECT A DLCI

To select a particular DLCI for transmission of a class, map the class to the DLCI by configuring a *datalink-traffic-index* parameter. Assign a datalink-traffic-index that is the same as the DLCI you want to select for transmission of the class. For instance, in this command, the class named ABC is configured to map to DLCI 678.

```
> config cbq.1 traffic-class.abc parent root-output-tree \
  bandwidth-allocation 10000000 datalink-traffic-index 678
```

CONFIGURING DLCIS

The foregoing command, which specifies a traffic index for a traffic class, only works if a DLCI exists that matches this index. In this case, only DLCI 678 will be used for packets belonging to class ABC. If this DLCI does not exist, the AP simply drops packets belonging to class ABC.

To add a DLCI, proceed as follows:

```
> add frame-relay.1 dlci.678
```

In conjunction with the definition for traffic class ABC, this definition causes the AP to forward ABC's packets into DLCI 678.

You can show all DLCIs that exist on the AP using this command:

```
> show frame-relay.1 dlcis summary

                                FRAME-RELAY.1 DLCIS SUMMARY

        INTERFACE|  DLCI|      State|        Creation Time|     Last Time Change|
                 |      | Row Status|            Dlci Type|
-----------------|------|-----------|---------------------|---------------------|
    frame-relay.1|   678|     active|7546206 (20:57:42.06)|7546206 (20:57:42.06)|
                 |      |     active|               static|
    frame-relay.1|   777|     active|9039586 (1 01:06:35.>|9039586 (1 01:06:35.>|
                 |      |     active|               static|
```

CONFIGURING PPP

The AP supports both Frame Relay and PPP for wide area connections. Frame Relay networking is the default. A Frame Relay connection comes up automatically upon completion of the physical interface configuration (for example, HSSI, MSSI, T1/E1, or T3).

To configure PPP, configure the physical layer connection and replace a Frame Relay instance with an instance of PPP using a *stack* command like the following.

```
> stack slot.<instance> cbq.<instance> <interface>.<instance> ppp.new
```

Issuing this command creates and uplinks a new PPP instance above the physical interface instance specified. An example of this command with the HSSI interface is the following:

```
> stack slot.3.1 cbq.2 hssi.1 ppp.new
```

You must issue this command for each Frame Relay instance that you want to replace with PPP, changing instance numbers as needed. Issue a *status* command when you have finished to verify the change.

SHOWING PPP DEFAULTS

PPP end stations automatically negotiate a common set of working parameters. As a result, the PPP default settings shown below will work for the majority of installations.

To show the PPP defaults, issue the following commands:

> show ppp.1 config

```
                                    PPP.1 CONFIG

          MTU* = 1500 bytes
Up Down Trap* = enabled
        Alias* = ""
```

> show ppp.1 link-config

```
                                PPP.1 LINK-CONFIG

Initial Max Receive Units* = 1500
      Receive Control Map* = 0x00000000
     Transmit Control Map* = 0x00000000
 Magic Number Negotiation* = true
                 CRC Size* = 16
            Echo Interval* = 3
```

The following list provides more information about the magic number negotiation and echo interval parameters:

- Magic Number Negotiation

  During a PPP session, the local node tries to perform magic number negotiation with the remote node if this parameter is set to true. The magic number negotiation can be used to determine whether the PPP session is in loopback mode. If the remote node returns the same magic number that the local node sent for a consecutive number of retries, then the AP thinks that the PPP session is in loopback mode. By default, this negotiation is enabled (the parameter is set to true). This parameter will be in effect when the link is next restarted.

  To disable magic number negotiation for PPP:

  > config ppp.<instance> magic-number-negotiation false

- Echo Interval

  The echo interval, functions as a keepalive mechanism for PPP sessions. Settings range from 0 - 3600 seconds. The default value is 3. In effect, setting the echo interval to 0 turns this mechanism off. In this case, PPP sessions would stay up indefinitely, whether or not connectivity with a peer exists. To configure the echo interval, use the following command:

  > config ppp.<instance> echo-interval 10

Use this command to get summary information on each PPP interface.

```
> show ppp-interfaces summary
```

```
                                PPP-INTERFACES SUMMARY

        INTERFACE|          Description| Type| Speed|
                 |          MAC Address| Mode| State|
                 |    Last State Change|
-----------------|---------------------|-----|------|
            ppp.1|  PPP slot: 3 port: 1|other|     0|
                 |    00:00:00:00:00:00|   up|testi>|
                 |24186699 (67 hours 1>|
```

VALIDATING PPP CONNECTIVITY

From the Access Point console, ping the router on the far end of the PPP cloud (see Figure 14).

```
> ping 192.168.2.2
```

If the ping fails, make sure the IP address and net mask are accurate.

CONFIGURING MULTILINK PPP

PPP MP (Multilink PPP) provides a cost-effective way to scale beyond available T1 bandwidth, without incurring the costs associated with a T3 connection. The AP supports Multilink PPP connections over HSSI and T1 connections.

Basically, Multilink PPP makes it possible to combine multiple T1 links into the same PPP session. This allows a number of T1 links to function together as a high-bandwidth physical connection for PPP.

REPLACE THE FRAME RELAY LINK WITH PPP

Frame Relay is the default data link for both HSSI and T1. Once a link is established with a remote peer, replace the frame relay instance with an instance of PPP, as follows:

```
> stack slot.1.1 cbq.1 t1.1 ppp.new
```

With PPP added in place of frame relay, the screen will look as follows:

SYSTEM STACK

ROW		Layer	Status
1	slot.1.1 Tcom4 T1 Driver	enabled	up
2	slot.1.1 cbq.1	enabled	up
3	slot.1.1 t1.1	enabled	up
4	slot.1.1 ppp.1	enabled	up
5	slot.1.1 ip.1	enabled	down
6	slot.1.2 Tcom4 T1 Driver	enabled	up
7	slot.1.2 cbq.2	enabled	up
8	slot.1.2 t1.2	enabled	up
9	slot.1.2 ppp.2	enabled	up
10	slot.1.2 ip.2	enabled	down

ADD THE PPP MP INSTANCES

Use the *stack* command to add an instance of PPP MP, as follows:

```
> stack slot.1.1 cbq.1 t1.1 ppp.1 ppp-mp.1.new ip.new
```

It is only necessary to include an IP instance when adding the first instance of PPP MP. The AP automatically stacks the same instance above subsequent PPP MP instances as you add them. For example, the following command adds another PPP MP instance to slot 1.2:

```
> stack slot.1.2 cbq.2 t1.2 ppp.2 ppp-mp.1.2
```

Notice that no IP instance is specified. The AP knows that all PPP MP instances must uplink to the same IP instance, and adds this instance for you. The screen that follows shows the result of these commands. Note that all layers are in an Up state.

SYSTEM STACK

ROW		Layer	Status
1	slot.1.1 Tcom4 T1 Driver	enabled	up
2	slot.1.1 cbq.1	enabled	up
3	slot.1.1 t1.1	enabled	up
4	slot.1.1 ppp.1	enabled	up
5	slot.1.1 ppp-mp.1.1	enabled	up
6	slot.1.1 ip.1	enabled	up
7	slot.1.2 Tcom4 T1 Driver	enabled	up
8	slot.1.2 cbq.2	enabled	up
9	slot.1.2 t1.2	enabled	up
10	slot.1.2 ppp.2	enabled	up
11	slot.1.2 ppp-mp.1.2	enabled	up
12	slot.1.2 ip.1	enabled	up

MANAGING BANDWIDTH ON A PPP MP LINK

In the foregoing example, two CBQ instances (CBQ.1 and CBQ.2) exist below the PPP MP instances in the protocols stack. While this is a valid PPP MP configuration, it is *not* one that allows management of aggregate PPP bandwidth on the PPP MP link.

To accomplish this goal, you need to use the *stack* command to remove each CBQ instance, and then insert a single CBQ instance above PPP-MP. To remove CBQ instances, simply omit them while building the stack. A sample command that removes a CBQ instance and adds a new instance above PPP-MP follows:

```
> stack slot.1.1 t1.1 ppp.1 ppp-mp.1.new cbq.new ip.new
```

A FINAL TIP...

Don't forget to add an IP address to the IP forwarder used by PPP MP. This address can be numbered or unnumbered.

In this case, the forwarder is IP.1. To add an IP address to IP.1, use this command:

```
> add ip.1 address.<IP address>
```

If you use an unnumbered interface, you must specify a reachable IP address as the source address.

CONFIGURING PPP AUTHENTICATION

PPP sessions over a physical link (router-to-router connections) do not require authentication. To configure the AP to authenticate physical PPP connections, follow the instructions given below.

See Page 173 for instructions on configuring the AP to be authenticated by a remote router.

CONFIGURING THE AP TO AUTHENTICATE CONNECTIONS FROM A REMOTE ROUTER

To authenticate the remote router, you must configure the following:

- Security preference protocols

  Security preferences are the authentication protocol types that the AP proposes to a remote router. The AP supports Challenge-Handshake Authenti-

cation Protocol (CHAP) or Password Authentication Protocol (PAP). The AP uses the chosen protocol to authenticate the remote router.

- RADIUS (Remote Authentication Dial-In User Service) or internal user list authentication

  Responses received from the remote peer include a user name. The AP authenticates this name either via RADIUS, or through comparison with an internal user name list.

To Allow authentication via CHAP only:

```
> config ppp.1 security-preference.1 protocol chap
```

To Allow Authentication via CHAP or PAP:

```
> config ppp.1 security-preference.1 protocol chap
> config ppp.1 security-preference.2 protocol pap
```

Where security-preference.1 is the first authentication protocol proposed to the remote peer. If the peer refuses preference 1 (in this case, CHAP), we will then propose preference 2 (e.g., PAP). If the peer refuses preference 2, we will not be able to authenticate them and the link will be dropped.

To show the configured PPP security preferences, use this command:

```
> show ppp.1 security-preference-table
```

PPP.1 SECURITY-PREFERENCE-TABLE

INTERFACE	PPPSECURITYCONFIGPREFERENCE	Protocol Status
ppp.1	1	chap valid
ppp.1	2	pap valid

Configuring the AP To Be Authenticated By a Remote Router

If the remote peer requires CHAP or PAP authentication of the AP, you must configure the Security Secrets that the AP will use to identify itself to the remote peer. Security Secrets are username/password pairs (or CHAP

name/secret pairs) used during the authentication process.

To configure the AP to be authenticated via CHAP:

```
> config ppp.1 security-secret.1 identity beavis protocol chap secret buthead
```

To configure the AP to be authenticated via CHAP or PAP:

```
> config ppp.1 security-secret.1 identity johnnybgoode protocol pap \
  secret hello
```

```
> config ppp.1 security-secret.2 identity johnnybgoode protocol chap \
  secret hello
```

It is a good idea make sure you have configured Security Secrets as intended by using this command:

```
> show ppp.1 security-secret-table
```

PPP.* SECURITY-SECRET-TABLE

INTERFACE	PPPSECURITYSECRETSIDINDEX	Direction Protocol Identity Secret Status
ppp.1	1	local-to-remote pap johnnybgoode hello - - - - d
ppp.1	2	local-to-remote chap johnnybgoode hello - - - - d
ppp.1	3	local-to-remote chap beevis buthead - - - - d

CONFIGURING SWITCHED MULTIMEGABIT DATA SERVICE

Switched Multimegabit Data Service (SMDS) is a cost-effective, connectionless service that provides fast WAN-LAN interconnection for low bandwidth connections.

An SMDS configuration typically contains SMDS switches and a router, gateway, or terminal connected to each Channel Service Unit/Data Service Unit (CSU/DSU). The Access Point system (AP) runs as a DTE.

Configuring SMDS on the AP involves modifying parameters used by SMDS Interface Protocol (SIP) Level 3 and Data eXchange Interface (DXI). DXI is used for communication between the AP and the CSU/DSU. The CSU/DSU must implement the full SIP protocol, while the AP only requires SIP Level 3 capability. DXI transmits encapsulated SIP Level 3 frames over a High Speed Serial Interface (HSSI) link.

Configuring SMDS involves performing the following tasks on routers at each end of the connection:

- Adding an SMDS interface between the physical layer (for example, HSSI or T1/E1) and the IP interface layer.
- Configuring the subnet for the SMDS interface.

ADDING SMDS INTERFACE

You add an SMDS interface on both routers using the *stack* command. For example, the following command adds an SMDS instance (smds.1) between the HSSI interface (hssi.1) and the IP interface (ip.3) for the slot.3.1 configuration.

```
> stack slot.3.1 cbq.3 hssi.1 smds.1 ip.3
```

You would assign the IP address for each router using this command:

```
> add ip.3 address.<ip address>
```

Each router has its own IP address, but they should be on the same subnet.

CONFIGURING THE SUBNET

To configure the subnet on both routers, use this command:

```
> config smds.1 subnet.<ip address>
```

For example, if you assigned 10.0.0.1 as the IP address for Router A, you would use this command:

```
> config smds.1 subnet.10.0.0.1
```

When configuring the subnet, you must specify these SMDS addresses:

- Group address
- Individual address
- ARP request address

The ARP request address and the group address are multicast addresses that can be the same on both routers and should match the address of the SMDS switch. The individual address should be different on each router. SMDS messages contain a source address and a destination address. The addresses are E.164 addresses (resembling North American telephone numbers).

CONFIGURING THE GROUP ADDRESS

The group address is a multicast SMDS address. Messages using group addresses allow a single source to send identical messages to many destinations because one address represents a group of systems. Use this command to configure the group address:

```
> config smds.<instance> group-address.e1<E.164 address>
```

The E.164 addresses are 10 digits long. There are four additional digits following the 10 digits. These digits are filled in with ffff by default.

CONFIGURING THE INDIVIDUAL ADDRESS

The individual address is a unicast SMDS address. Messages using individual addresses are single source, single destination messages.

```
> config smds.<instance> smds-address.c1<E.164 address>
```

The E.164 addresses are 10 digits long. There are four additional digits following the 10 digits. These digits are filled in with ffff by default.

CONFIGURING THE ARP REQUEST ADDRESS

The ARP request address is a multicast address used for address mapping. This address is usually the same as the group address.

```
> config smds.<instance> arp-request-address.e1<E.164 address>
```

The E.164 addresses are 10 digits long. There are four additional digits following the 10 digits. These digits are filled in with ffff by default.

CONFIGURING CRC

To configure the cyclic redundancy check (CRC) for error checking, you would configure the value at the HSSI interface level using this command:

```
> config hssi.<instance> crc-mode <mode>
```

where mode is crc16 (the default), crc32, or crcnone.

DISPLAYING THE SMDS CONFIGURATION

To display the SMDS configuration, use this *show* command:

```
> show smds-interfaces summary
```

To display a particular SMDS instance, use this command:

```
> show smds.<instance>
```

CONFIGURING STATIC SMDS ADDRESSES

To configure static SMDS addresses, use this command:

```
> config smds.<instance> arp-cache-entry.<IP address> entry-type static\
  mac-address <MAC address>
```

This command adds a static SMDS address (the MAC address) for a particular SMDS instance and IP address. The MAC address is 16 digits and includes c1at the beginning, the E.164 address in the middle, and the additional four digits (usually ffff) at the end of the address. For example, c11234567890ffff.

CONFIGURING THE ARP TABLE SIZE

You can configure the size of the ARP table for SMDS. When you change the ARP table size, it takes effect immediately. Any static ARP entries are preserved, but dynamic entries are removed and learned again as needed. You should not make the ARP table size equal or less than the number of defined static entries because that leaves no room for dynamic entries. The allowed range for the ARP table size is 1 to 5000. The default is 1000.

You can change the ARP table size as follows:

```
> config smds.1 arp-config arp-table-size 100
```

DISPLAYING THE SMDS AND MAC ADDRESS MAPPING

To display the mapping between the SMDS and MAC addresses:

```
> show smds-interfaces arp-cache summary
```

This command displays the SMDS interface names, the network address (the IP address), the MAC address, and the entry type.

CONFIGURING IP

Access Point interfaces do not come up with a default IP address. Initially, you must connect to the AP from a PC or terminal via a serial line and establish LAN side connectivity as part of the installation procedure.

You must assign an IP address to each interface to establish IP connectivity. This address must be added to the IP instance that resides in the physical layer interface you are configuring. Issue a *status* command if you are unsure of which IP instance is associated with each interface.

ADDING AN IP ADDRESS TO EACH INTERFACE

You must assign an IP address to each interface by using this command:

```
> add ip.<instance> address.<address> net-mask <network mask>
```

The IP instance is the one associated with the particular interface (Ethernet, HSSI, MSSI, T1, E1, T3, or ATM). If you do not supply a network mask, the default network mask is 255.0.0.0.

Once you have added an IP address to an interface, it can be used as the IP forwarder for that interface. You can connect to the device over Ethernet for the LAN or over the other physical layer connections for the WAN. You must add an IP address on the WAN side of the AP to bring up the link. The following example shows the addition of an IP address to the IP instance (ip.2) for the WAN interface shown in Figure 14:

```
> add ip.2 address.192.168.2.1
```

To check that IP addresses have been added successfully, display the IP address table using this command.

```
> show ip.* address-table summary
```

USING UNNUMBERED INTERFACES

You can use unnumbered interfaces when you want to conserve real IP addresses. Unnumbered interfaces function in most of the same ways as ordinary (numbered) IP addresses. They differ from numbered interfaces in the following ways:

- Only one unnumbered interface can exist on any IP instance. In contrast to this, many numeric IP addresses can exist on an IP instance.
- Numbered and unnumbered IP addresses cannot coexist on the same IP interface.

ADDING AN UNNUMBERED INTERFACE

To add an unnumbered address to an IP interface, proceed as follows:

```
> add ip.11 address.0.0.0.0 src-addr 6.7.8.9
```

If you assign an unnumbered address to an IP instance without providing a source address, the AP uses the router address as a source address. If no router address exists, the AP uses 0.0.0.0 as the source address.

NOTE *If the interface associated with the source address is down, then the unnumbered interface is still up. However, if the source address is deleted or does not exist, then the unnumbered interface goes down.*

To check that an unnumbered interface has been added successfully, display the IP address table using the *show ip.* address-table summary* command.

USING THE LOOPBACK INTERFACE ADDRESS AS THE SOURCE ADDRESS

In the preceding sample command, the source address chosen (6.7.8.9) is associated with the loopback interface, ip.loopback. Using the loopback interface address as a source address ensures that the unnumbered address will remain active at all times because the loopback interface is always up and active.

The loopback address must be configured before you can use it as a source address. See "Using the Loopback Interface Address" on Page 180 for detailed information on configuring ip.loopback.

DELETING AN UNNUMBERED INTERFACE

To delete an unnumbered interface, proceed as follows:

```
> remove ip.11 address.0.0.0.0 src-addr 6.7.8.9
```

To check that an unnumbered interface has been removed successfully, display the IP address table using this command.

```
> show ip.* address-table summary
```

USING THE LOOPBACK INTERFACE ADDRESS

The loopback interface (ip.loopback) provides a source address that is always active. It is useful as the source address for unnumbered addresses or other IP addresses (such as the router address) because once you assign an IP address to ip.loopback, the address is always up.

To assign an IP address to the ip.loopback interface, proceed as follows:

```
> add ip.loopback address.<ip address> net-mask <network mask>
```

For example, assign 6.7.8.9 to the ip.loopback interface with this command:

```
> add ip.loopback address.6.7.8.9
```

The loopback address has a default network mask of 255.0.0.0. The address's status is always active.

USING THE IP.SINK ADDRESS

The AP also supports a special IP interface called ip.sink. Use of the ip.sink interface is important in the IPSec context. You do not have to create the ip.sink interface. It is always present, and always up (Active). Its purpose is to discard any packets forwarded to it. This ensures that these packets will not be sent to an intended destination over an insecure path.

When IPSec tunnels are created, the AP automatically creates a static route to the remote tunnel endpoint. Should this remote endpoint be unreachable, the packets are forwarded to ip.sink (and discarded) rather than being routed to the endpoint by some other path.

DISPLAYING IP ADDRESSES

To display information on addresses associated with IP instances and other interfaces (such as ip.loopback and ip.sink), use this command:

```
> show ip.* address-table summary
```

Following is a sample display of the output of this command:

IP.* ADDRESS-TABLE SUMMARY

INTERFACE	ADDRESS	Net Mask Configured Protocols Src Addr	Broadcast Address Row Status Router Address
ip.8	0.0.0.0	255.255.255.255 ""	0 active
		208.218.164.6	false
ip.sink	0.0.0.0	255.255.255.255 ""	0 active
		208.218.164.6	false
ip.10	0.0.0.0	255.255.255.255 ""	0 active
		208.218.164.6	false
ip.11	2.3.4.5	255.0.0.0 ""	1 active
		0.0.0.0	false
ip.loopback	6.7.8.9	255.0.0.0 ""	1 active
		0.0.0.0	false

5

CONFIGURING ROUTING

The sections that follow explain how to configure routing for the Access Point system (AP). The sections cover these areas:

Configuring IP Interface Parameters *184*
Configuring General IP Parameters *187*
Supported IP Protocols *190*
Configuring Static Routes *191*
Configuring RIP *192*
Configuring OSPF *196*
Configuring ISIS *211*
Configuring BGP *218*
Route Policy *227*
Creating and Selecting a Forwarding Policy *249*
Configuring VRRP *253*

CONFIGURING IP INTERFACE PARAMETERS

Adding an IP address is a basic requirement for establishing initial IP connectivity. To add an IP address, use this command.

```
> add ip.<instance> address.<ip address> net-mask <ip address>
```

A net mask need not be specified if the natural mask is correct for the address. This section describes additional configuration capabilities for AP IP interfaces.

IP MULTINETTING

You can add many IP addresses to each AP interface. This allows the AP to participate in many different subnets at the same time. All IP forwarding and routing protocol operations function in the same way as when a single IP address assigned to an interface.

USING UNNUMBERED IP ADDRESSES

In point-to-point WAN topologies, it is often unnecessary and undesirable to use a new subnet for every WAN interface. In these situations, the link can operate as an *unnumbered IP interface*. When configured as an unnumbered interface, no IP address is configured and IP packets originated from the router over this interface use the configured *source address* for the interface as the source address in the outgoing IP packets.

To configure an interface unnumbered, issue the following command. The *source-address* used must be an active IP address on the AP.

```
> add ip.1 address.0.0.0.0 src-addr 10.2.3.4
```

USING THE ROUTER ADDRESS

The router address is primarily intended as an identifying address for VPN based capabilities. However, it is useful in non-VPN configurations as the default source address for unnumbered interfaces.

```
> config ip.1 address.1.2.3.4 net-mask.255.0.0.0 router-address true
```

VERIFYING THE SOURCE OF A PACKET

Reverse path verification allows the AP to verify the source address of incoming packets. The source address of the incoming packet is looked up in the route table, and if the next hop interface does not match that of the incoming packet, the AP drops the packet. Reverse path verification does not work in asymmetric routes, but it is useful in preventing spoofed IP packets from moving beyond the router.

By default, reverse path verification is disabled. To enable it, use the following command:

```
> config ip.2 verify-reverse-path enabled
```

To see if reverse path verification is enabled, enter the following command:

```
> show ip.2 config
```

```
                IP.2 CONFIG

                 MTU* = 1500 bytes
       Up Down Trap* = enabled
              Alias* = ""
       Primary Addr* = 1.1.1.1
     Negotiate Addr* = enabled
Verify Reverse Path* = enabled
```

USING PROXY ARP

Proxy ARP allows the AP to handle address resolution requests for hosts not physically located on a local subnet.That is, the AP can reply to ARP requests on behalf of remote or virtual hosts. The AP supports Proxy ARP for:

- VPN or L2TP hosts given addresses from a locally attached subnet
- The entire subnet

PROXY ARP SUPPORT FOR VPN OR L2TP HOSTS

The AP uses Proxy ARP when dial-in VPN or L2TP clients receive an address from RADIUS or a local pool for a subnet that is configured on one of the router's LAN interfaces. The dial-in host is “virtually” on the local subnet. Because other hosts on that subnet will issue an ARP for that address when they need to send it packets, Proxy ARP is supported to allow the router to answer the ARP with its own address, so it can route packets to the appropriate tunnel.

When a dial-in client receives a host address, it automatically installs a static route for the host address in the static route table so packets destined for the host know which tunnel to enter. Static route processing also checks the local subnets to see if there is a locally configured supernet of the static route. If there is, static route processing enables Proxy ARP on that interface for the network (or host) defined by the newly added static route.

The following displays help explain this process. Use the address table below as a baseline:

```
> show ip.* address-table summary
```

```
                                IP.* ADDRESS-TABLE SUMMARY

      INTERFACE|         ADDRESS|        Net Mask| Configured Protocols|
               |                 |      Row Status|             Src Addr|
---------------|-----------------|----------------|---------------------|
           ip.1|         20.1.1.1|     255.255.0.0|                   ""|
               |                 |          active|              0.0.0.0|
           ip.2|         20.2.2.1|     255.255.0.0|                   ""|
               |                 |          active|              0.0.0.0|
           ip.3|        10.1.3.96|   255.255.255.0|                   ""|
               |                 |          active|              0.0.0.0|
           ip.4|       192.1.1.1 |   255.255.255.0|                   ""|
               |                 |          active|              0.0.0.0|
       ip.sink|         0.0.0.0 | 255.255.255.255|                   ""|
               |                 |       notReady |              0.0.0.0|
```

Now, assume a static route has been added for 20.1.1.3, corresponding to a new tunnel host address assignment:

```
> show ip static-route-table summary
```

```
                                IP STATIC-ROUTE-TABLE SUMMARY

 DESTINATION|            MASK| TOS|         NEXT HOP|        Cost|   Status|
            |                |    |                 |  Row Status|   Source|
            |                |    |                 |   Proxy Arp| Volatile|
------------|----------------|----|-----------------|------------|---------|
   20.1.1.3 | 255.255.255.255|   0|             ip.4|           1|   active|
            |                |    |                 |      active|  netmgmt|
            |                |    |                 |        true|    false|
   20.3.1.3 | 255.255.255.255|   0|             ip.4|           1|   active|
            |                |    |                 |      active|  netmgmt|
            |                |    |                 |       false|    false|
```

Notice that the Proxy Arp field indicates true because 20.1.1.3/32 is a subnet of a locally attached 20.1.1.1/16, but 20.3.1.3/32 does not indicate that Proxy ARP is active.

NOTE *Because Proxy ARP occurs as a result of a new static route's relationship to locally attached subnets, you can enable Proxy ARP, for network applications if necessary, by adding static routes directly into the static route table. You can do this for*

static routes for networks with less than 32 bit masks. As long as the statically defined route is for a network that is a subnet of a locally attached interface, Proxy ARP will be active for the whole subnet defined for the static route.

PROXY ARP SUPPORT FOR THE ENTIRE SUBNET

You can control whether the AP responds with its own address for *every* address on a particular subnet by enabling Proxy ARP for the entire subnet with the *subnet-proxy-arp* parameter. The following command enables Proxy ARP on subnet 36.3.4.10:

```
> config ip.3 address.36.3.4.10 subnet-proxy-arp enabled
```

Proxy ARP is disabled by default, and you should only enable it after thoroughly determining its need in your network.

CONFIGURING GENERAL IP PARAMETERS

Several parameters are used to control various aspects of the basic IP router. These range from high level control of forwarding to the memory allowances for the different routing pools. These parameters can be displayed by issuing the following commands:

```
> show ip summary
```

```
> show ip memory
```

```
> show ip counters
```

SUPPORT FOR ICMP REDIRECT

ICMP redirect is disabled by default. To enable it, proceed as follows:

```
> config ip icmp-redirect enabled
```

FORWARDING OF DIRECTED BROADCASTS

The AP supports forwarding for directed broadcasts. This feature is disabled by default because directed broadcasts can be used as a denial-of-service attack.

To enable the forwarding of directed broadcasts, use this command.

```
> config ip forward-directed-broadcasts enabled
```

SUPPORT FOR A DEFAULT NETWORK

You can configure a default network as the next hop for static routes. Doing so enables dynamic routing to determine the real next hop to use rather than statically defining it.

To configure a default network as the next hop in a routing path, use the following commands. Note that the next-hop network should not be locally connected to the AP.

```
> config ip default-network 192.32.23.0

> add ip static-route.0.0.0.0 mask.0.0.0.0 tos.0 next-hop.192.32.23.0
```

These commands cause the AP to calculate the next hop for the default route based on the next hop to the 192.32.23.0 network. Until there is a route (and therefore a next hop) to the default network, the route to this network will not appear in the routing table.

DISPLAYING A LIST OF ROUTES FOR A SPECIFIC DESTINATION

You can display a list of routes for a specific destination in the order of preference (most-to-least specific). The destination can be either a host or a network address. In the following example for target destination 2.2.2.1, the AP returns three routes:

- 2.2.2.0/24
- 2.0.0.0/8
- Default route

```
> show ip destination.2.2.2.1 summary

                    IP DESTINATION.2.2.2.1 SUMMARY

     DESTINATION| ROUTE|          Prefix|                  Mask|
                |      |        Next Hop|              Interface|
                |      |      Route Type|               Protocol|
                |      |             Age|                   Cost|
                |      |        Distance|                       |
----------------|------|----------------|-----------------------|
          2.2.2.1|     1|         2.2.2.0|    255.255.255.0(/24)|
                 |      |         3.3.3.2|           ip.2/hssi.1|
                 |      |          remote|                   rip|
                 |      |          36 sec|                     2|
                 |      |             120|
          2.2.2.1|     2|         2.0.0.0|         255.0.0.0(/8)|
                 |      |         1.1.1.1|            ip.1/eth.1|
                 |      |           local|               netmgmt|
                 |      |          37 sec|                     1|
                 |      |              10|
          2.2.2.1|     3|         0.0.0.0|           0.0.0.0(/0)|
                 |      |         3.3.3.1|            ip.2/hssi.1|
                 |      |           local|               netmgmt|
                 |      |          37 sec|                     1|
                 |      |              10|
```

ADVANCED IP MEMORY CONFIGURATION

By default, the AP allocates approximately 2 Mb of route memory to IP. In certain circumstances, you might need to change the amount of memory allocated to IP routing. After making any such changes, you must reboot the AP for the new memory allocations to take effect.

CHANGING IP MEMORY ALLOCATION

To change the memory used by IP, use this command:

```
> config ip configured-route-memory <number of bytes>
```

The number of bytes must fall within the range of 131072 and 16777216. If you are decreasing the memory allocated to IP, bear in mind that this allocation includes memory allocated to BGP. If you do not allocate sufficient memory, errors can occur when the AP uses BGP routing.

CHANGING ROUTE ALLOCATION

By default, the AP allocates approximately 80,000 routing table entries. To change this allocation, use this command:

```
> config ip configured-max-routes <number of routes>
```

The minimum and maximum allowed values are 0 and 81920. The amount of memory you allocate to IP route memory and maximum number of routes specified must correlate because each route takes a small amount of memory.

CHANGING BGP MEMORY ALLOCATION

You can also change the memory allocated to BGP for route and path information. By default, the AP allocates around 80,000 routes to BGP. To change this allocation, use this command:

```
> config ip bgp configured-max-routes <number of routes>
```

The minimum and maximum allowed values are 0 and 81920.

You can also change the memory allocated to BGP paths. To do so, use this command:

```
> config ip bgp configured-max-paths <number of paths>
```

The number of paths allocated must be between 0 and 15360. Note that changes to BGP memory allocation should be related to the overall IP memory configuration.

SUPPORTED IP PROTOCOLS

The AP supports routing via static route configuration and dynamic routing using the following protocols:

- RIP
- OSPF
- ISIS
- BGP4

The routing protocols share a common route policy mechanism which provides sophisticated route filtering, parameter modification, and import/export among the protocols.

In addition, the AP supports the Virtual Router Redundancy Protocol (VRRP), providing IP failover capability.

These capabilities and their interaction are described in the following sections.

CONFIGURING STATIC ROUTES

A static route defines a path between a routing interface and another routing interface that serves as the next-hop for traffic destined to a desired network.

A default route is a special case of a static route. When a default route is configured, all traffic that does not have a defined route in the routing table is sent to this next-hop address.

ADDING A STATIC ROUTE

Add a static route to assure connectivity between the AP and the router that is the next hop for traffic to the desired network. To add a static route, use this command:

```
> add ip static-route.192.32.1.0 mask.255.255.255.0 tos.0 \
  next-hop.192.168.2.2
```

This command assures that traffic destined for the network and mask specified is routed to the next-hop address, 192.168.2.2.

You can also add a static route over an interface. To do this, configure the next hop to be the interface (numbered or unnumbered), as follows:

```
> add ip static-route.192.32.1.0 mask.255.255.255.0 tos.0 next-hop.ip.2
```

ADDING A DEFAULT ROUTE

You add a default route in the same manner as a static route. The only difference is that, instead of a specific network and mask, you provide 0.0.0.0 and 0.0.0.0 respectively.

The sample command shown here defines a default route with the same next-hop network as the preceding example.

```
> add ip static-route.0.0.0.0 mask.0.0.0.0 tos.0 next-hop.192.168.2.2
```

SHOWING STATIC ROUTES

Use this command to verify that a static route has been added.

```
> show ip static-route-table summary
```

```
                                IP STATIC-ROUTE-TABLE SUMMARY

    DESTINATION|            MASK|   TOS|        NEXT HOP|       Cost| Status|
               |                |      |                 | Row Status| Source|
               |                |      |                 |  Proxy Arp|       |
---------------|----------------|------|-----------------|-----------|-------|
         0.0.0.0|         0.0.0.0|     0|  208.218.164.18|          1| active|
               |                |      |                 |     active|netmgmt|
               |                |      |                 |      false|       |
        10.0.0.0|       255.0.0.0|     0|   208.218.164.6|          1|inacti>|
               |                |      |                 |     active|netmgmt|
               |                |      |                 |      false|       |
      38.1.31.32| 255.255.255.252|     0|  208.218.164.20|          1| active|
               |                |      |                 |     active|netmgmt|
               |                |      |                 |      false|       |
```

CONFIGURING RIP

The AP supports versions 1 and 2 of the Routing Information Protocol (RIP). When adding RIP to an interface, you can configure the AP to send either RIP1 or RIP2 updates, and to receive either or both versions of the protocol.

The AP supports two levels of policies for RIP. You can think of these policies as *implicit* or *explicit*. Implicit RIP policies come into effect whenever you add RIP to an interface. You can alter the default values for the implicit RIP policies to suit your needs. However, it is important to understand that adding RIP implements some type of (implicit) policy configuration for send and receive.

In contrast to implicit RIP policies, the AP supports explicit policy routing for RIP send policies. Explicit send policies provide a way of matching to specific criteria to determine whether to execute an action. In this respect, explicit policies provide more fine-grained control for influencing the advertisement or reception of routes than is possible with the more generalized (implicit) policy routing that comes with RIP. See “Configuring RIP Route Policies” on Page 195 for more information on configuring RIP route policies.

The rest of this section covers the following topics:

- Activating and deactivating RIP
- Accepting or changing the default implicit RIP policies
- Configuring RIP route policies

ACTIVATING AND DEACTIVATING RIP

To activate RIP on an interface, issue this command for each IP address desired on the interface.

```
> config ip.1 address.<ip address> configured-protocols rip
```

The default value for send policies is ripVersion2. The default for receive policies is rip1OrRip2. The defaults come into play if you activate RIP and do not explicitly specify send or receive values. To change the default values, see the "Changing Default Implicit Policies" on Page 193.

To deactivate RIP, use the bang symbol (!) as follows:

```
> config ip.1 address.<ip address> configured-protocols!rip
```

EQUAL-COST MULTIPATH LOAD BALANCING FOR RIP

If multiple AP interfaces are running RIP with multiple, equal-cost paths to the same destination, all the equal-cost paths are added to the routing table. This potentially provides several next hops to a destination.

CHANGING DEFAULT IMPLICIT POLICIES

To change the default values for implicit send and receive policies, use the following command:

```
> config ip rip.<ip address> [send <action>] [receive <action>]
```

For example, to change the values for the send and receive policies of addresses 126.3.4.5 and 124.3.4.5 to doNotSend and rip1OrRip2 respectively, issue the following commands:

```
> config ip rip.126.3.4.5 send doNotSend receive rip1OrRip2
```

```
> config ip rip.124.3.4.5 send doNotSend receive rip1OrRip2
```

Table 13 provides completion values for implicit RIP interface send and receive actions. Note that these values differ from those used with explicit RIP route policies, as explained on Page 195.

Table 13 RIP Completion Values

Action Type	*Completions*
Send	• doNotSend • ripVersion1 • rip1Compatible • ripVersion2 (*default)
Receive	• rip1 • rip2 • rip1OrRip2 (*default) • doNotReceive

SHOWING RIP INFORMATION

You can determine whether RIP is learning routes by checking the routing table. Do this by issuing this command:

```
> show ip route-table summary
```

Look for routes that identify RIP as the routing protocol in use.

```
                                IP ROUTE-TABLE SUMMARY

DESTINATION|                MASK| TOS|          NEXT HOP|  Interface|
           |                    |    |                  |Route Type|
           |                    |    |                  |  Protocol|
           |                    |    |                  |       Age|
           |                    |    |                  |      Cost|
-----------|--------------------|----|------------------|----------|
   0.0.0.0|         0.0.0.0(/0)|   0|     208.218.164.5|ip.1/eth.1|
           |                    |    |                  |     local|
           |                    |    |                  |   netmgmt|
           |                    |    |                  |   1215952|
           |                    |    |                  |         1|
  10.0.0.0|   255.255.0.0(/16)|   0|       10.0.16.3|ip.2/eth.2|
           |                    |    |                  |     local|
           |                    |    |                  |   netmgmt|
           |                    |    |                  |   1215951|
           |                    |    |                  |         1|
```

CONFIGURING RIP ROUTE POLICIES

The AP implements implicit RIP send and receive configuration on a per-interface basis. You can also use explicit route policies for RIP to fine-tune the advertisement and reception of routes on specific networks, based on configurable match criteria.

Explicit RIP send policies conform to the general route policy mechanism (see "Route Policy" on Page 227) that governs other routing protocols. A single interface can have both implicit and explicit RIP policies. If no explicit send policy matches, the implicit RIP behavior determines the action to take for a route.

DEFAULT RIP BEHAVIOR

An explicit RIP route policy need not exist for RIP advertisement to occur. In fact, having no explicit policy is equivalent to a "send on all circuits" policy because the default implicit RIP send value (ripVersion2) comes into play. Likewise, by default, all routes advertised by a RIP neighbor are received by RIP and entered into the route selection process.

Because RIP has the concept of implicit and explicit route policy configuration, it is important to understand how these policy areas interact if both are configured. The specifics of explicit RIP route policy configuration are described in detail in "RIP-Specific Capabilities" on Page 233.

CONFIGURING OSPF

OVERVIEW

OSPF is a link-state protocol that uses an SPF routing algorithm to calculate the least-cost path for delivery of IP packets. The protocol works by periodically sending Link State Advertisements (LSAs) across a backbone area to other OSPF areas within an autonomous system (AS). Information gathered from LSAs is collected in a link-state database, and shared among OSPF areas. Sharing this information gives these routers (called area border routers) a common view of the network. As a result, OSPF does not need to send portions of its routing table to update its neighbors. Instead, LSAs are used to maintain a consistent topological view among participating routers.

OSPF COMPONENTS

OSPF components operate within a single *autonomous system* (AS) comprised of *OSPF areas*. Each AS contains the following components:

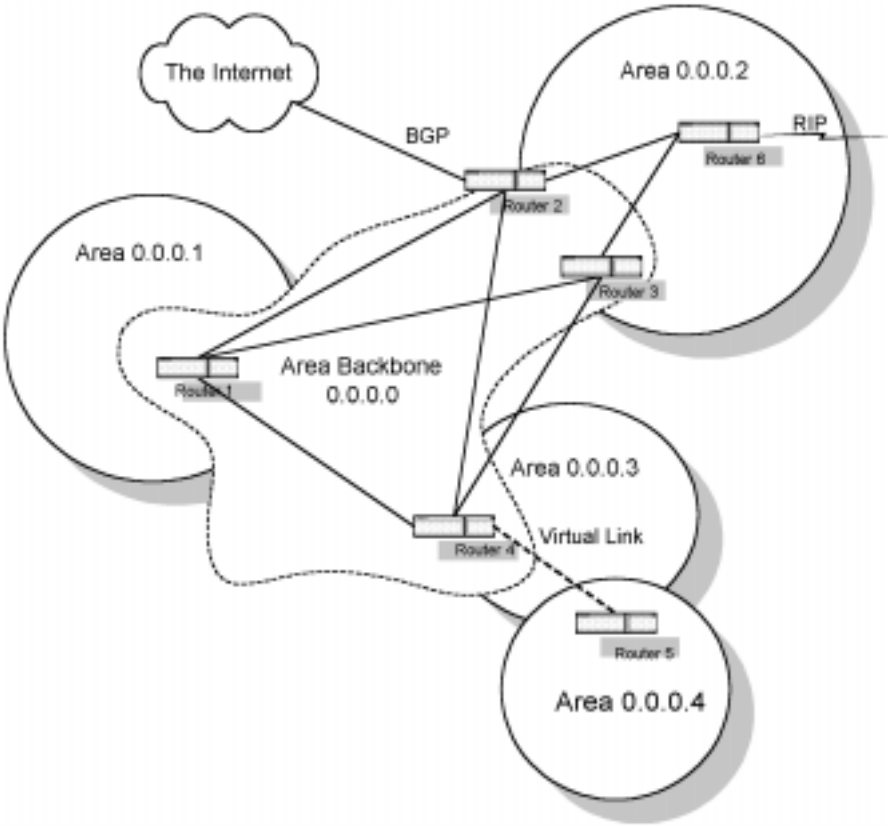
- OSPF areas
- Border and boundary routers
- Internal routers
- An OSPF backbone area

An autonomous system (AS) is a group of hosts or networks under a common administration, or domain. An *OSPF area* is a collection of contiguous networks and attached hosts. Some routers participate in multiple areas. Appropriately called *area border routers*, these routers maintain separate link-state databases for each OSPF area to which they are connected. *AS boundary routers* are routers running an external gateway protocol, such as BGP, that provides a link to networks external to the autonomous system. An area border router may also function as an AS boundary router. Internal routers operate internal to each OSPF area, periodically receiving LSAs from the area border routers to which they are connected. Itself an OSPF area, an *OSPF backbone* distributes routing information between attached OSPF areas. The backbone comprises all area border routers.

Figure 15 shows each of these components as they are related within a sample autonomous system. This system consists of four OSPF areas. Routers 1, 2, 3 and 4 comprise the backbone. Router 2 is an AS boundary router, capable of

importing route information from outside networks via BGP. Routers 1, 2, 3, and 4 are border routers. These routers comprise the OSPF backbone area. Notice that Router 2 is both a border and a boundary router.

Figure 15 OSPF Areas and Backbone



Routers that are physically connected to the OSPF backbone (area border routers) share link-state information between OSPF areas. However, other routers can also share link-state updates and information. Making this possible requires the establishment of virtual links between these routers and a router on the backbone. In Figure 15, Area 0.0.0.4 is not physically connected to the backbone area. To share link-state information, Router 5 requires a virtual link to Router 4. “Configuring OSPF Virtual Links” on Page 206 explains how to establish virtual links.

PLANNING YOUR OSPF NETWORK

Before configuring OSPF, doing some upfront planning can save you trouble, time, and expense. OSPF supports three area types: normal, stub, and not-so-stubby. You should understand the implications of these three area types (See "Configuring Stub and Not-So-Stubby Areas" on Page 207 for details about these areas.).

Setting up an OSPF network with the type of areas you need, or thoughtfully adding a new area to an existing network is far less disruptive than attempting to convert from one area type to another. Careful planning can eliminate the need for converting from one area to another.

SUMMARY OF KEY POINTS

Following are some general comments on OSPF. These comments are intended as a summary of key points. For more complete information on the protocol, see one of the OSPF sources listed in the preface.

- *You must enable OSPF on each routing interface in order for the protocol to function.*
- *To delete or change an OSPF area, OSPF must be disabled.*
- *The OSPF router ID is automatically set to the first IP address that is configured on the interface running OSPF.*

  If the interface designated as the router ID goes down or is deleted, the AP resets the OSPF router ID to the next IP address. In this event, routing information must be recalculated and sent out. To avoid this, you can set the router ID to a specific value, as follows:

  ```
  > config ip ospf router-id <IP address>
  ```

- *If a send policy has been configured to import routes into OSPF, the is-as-border-router parameter must be set to TRUE, as follows:*

  ```
  > config ip ospf is-as-border-router true
  ```

  This is the default state for the is-as-border-router parameter.

CONFIGURING OSPF INTERFACES

Configuring OSPF interfaces involves the following few basic steps:

1. Showing OSPF defaults
2. Configuring OSPF areas
3. Enabling OSPF
4. Configuring passive OSPF interfaces
5. Configuring OSPF interface types

Depending on your specific needs, these two other steps might be necessary.

6. Creating virtual links
7. Designating stub and not-so-stubby areas

The brief sections that follow explain each of these steps.

SHOWING OSPF DEFAULTS

By default, OSPF is enabled on the AP. However, you might wish to verify the protocol's state. Issue the following command to view all OSPF default values. Notice that the default status of OSPF is enabled.

```
> show ip ospf summary
```

```
                              IP OSPF SUMMARY

                  Router ID* = 157.130.6.38   Supports Type Of Service* = false
                     Status* = enabled                LSAs Originated  = 3
                    Version  = version2               New LSAs Received  = 0
    Is Area Border Router    = false             AS External LSA Limit  = -1
      Is AS Border Router*   = true            IP Multicast Forwarding  = 0
             External LSA Count  = 0          Seconds In Overflow State  = 0
    External LSA Checksum Sum    = 0            Demand Routing Support  = false
```

Notice also that the default value of *Is Area Border Router* is false, while *Is AS Border Router* is true. These similarly-named MIB values apply to different characteristics: to the functions of area border router and AS boundary router, respectively.

There is no need to specifically configure a router as a border router. The OSPF protocol determines border router status on its own, based on the information you provide on the OSPF backbone area, interfaces, and AS areas.

CONFIGURING OSPF AREAS

The AP creates the OSPF backbone area (area ID 0.0.0.0) by default. OSPF areas are specified in IP address format. To specify an OSPF area, use this command:

```
> add ip ospf area.<area>
```

Following is a sample area declaration:

```
> add ip ospf area.2.2.2.2
```

To view all areas configured on the AP, issue this command:

```
> show ip ospf areas summary
```

Notice that two OSPF areas exist: one is area 0.0.0.0, the default backbone area. Another is the area just added: area 2.2.2.2

```
                        IP OSPF AREAS SUMMARY

            AREA ID|  Import As Extern|     Area Border Routers|
                   | AS Border Routers|        Num LSDB Entries|
                   |  LSA Checksum Sum| Summary LSA Propagation|
                   |            Status|
----------------- -|------------------|------------------------|
                0.0.0.0|    importExternal|                       0|
                   |                 0|                       0|
                   |                 0|         sendAreaSummary|
                   |            active|
                2.2.2.2|    importExternal|                       0|
                   |                 0|                       0|
                   |                 0|           noAreaSummary|
                   |            active|
```

ENABLING OSPF

Before enabling OSPF on an interface, decide which area you want the interface assigned to. By default, interfaces are added to the backbone area (0.0.0.0), so no additional configuration is necessary. After deciding this, assign each interface to an OSPF area if necessary:

```
> config ip ospf.<interface address> area-id <area ID>
```

Following is a sample declaration that adds interface 1.2.3.4 to area 0.0.0.1.

```
> config ip ospf.1.2.3.4 area-id 0.0.0.1
```

If you are configuring an unnumbered interface, use the IP instance instead of the IP address, as in this command:

```
> config ip ospf.ip.3 area-id 0.0.0.1
```

After adding an interface to an area, enable OSPF on the interface with the following command:

```
> config ip ospf.<interface address> admin-status enabled
```

Another way to enable OSPF on an interface follows:

```
> config ip.<instance> address.<IP address> configured-protocols ospf
```

To disable OSPF, use the bang (!) qualifier as follows:

```
> config ip.<instance> address.<IP address> configured-protocols !ospf
```

CONFIGURING PASSIVE OSPF INTERFACES

You can configure an OSPF interface to operate in passive mode. In this mode, the network is advertised (as a stub network), but it does not establish any adjacencies. Passive mode is useful in those situations where you want to include the interface address in router LSAs, but you do not want to establish adjacencies with other routers (for example, customer routers). To cause an interface to operate in passive mode, set the hello timer to 0 using this command:

```
> config ip ospf.<interface address> hello-interval 0
```

CONFIGURING OSPF INTERFACE TYPES

OSPF supports the following interface types:

- Broadcast
- Point-to-point
- Non-broadcast multiaccess (NBMA)
- Point-to-multipoint

ABOUT DEFAULT TYPES

LAN interfaces are always broadcast type and unnumbered interfaces are always point-to-point type. By default, WAN interfaces are point-to-point type, but WAN interfaces can be any of the supported types.

Use the point-to-point interface type for a PPP or single PVC connections over Frame Relay or ATM and the point-to-multipoint interface type for Frame Relay or ATM connections with multiple PVCs.

If you are using the default interface types, further configuration to support the interface type is unnecessary. However, for numbered WAN interfaces, remember that both sides of the link must be configured with the same interface type for proper operation.

NOTE *If you are using an AP running Version 2.2 software over a WAN connection to another AP that is running a previous version of the Access Point software, you must set the Version 2.2 interface type to broadcast so the types match.*

ABOUT NEIGHBOR CONFIGURATION

If you are using the NBMA or point-to-multipoint interface types, then you must configure neighbors as described in "Configuring NBMA Neighbors" on Page 203 or "Configuring Point-To-Multipoint Neighbors" on Page 205.

SETTING OSPF INTERFACE TYPES

If you want to change the default OSPF interface type, perform these steps:

1 Disable the OSPF interface with this command:

```
> config ip ospf.<interface address> admin-status disabled
```

You should disable the interface you are changing so dynamic neighbor

discovery does not cause unexpected neighbors to be found before both sides have been set to the proper type.

2 Change the interface type with this command:

> config ip ospf.<interface address> type <interface type>

where <interface type> can be broadcast, pointToPoint, nbma, or pointToMultipoint. If you change to the point-to-multipoint or NBMA type, you must configure neighbors.

3 Reenable the OSPF interface with this command:

> config ip ospf.<interface address> admin-status enabled

Before changing the type, you can check the current type with this command:

> show ip ospf-interfaces status

IP OSPF-INTERFACES STATUS

INTERFACE ID	Area ID	Type	Admin Status
	Router Priority	State	Designated Router
	Backup Designated Router	Events	Authentication Type
ip.1	0.0.0.0	pointT>	disabled
	1	down	0.0.0.0
	0.0.0.0	0	none
ip.sink	0.0.0.0	pointT>	disabled
	1	down	0.0.0.0
	0.0.0.0	0	none

CONFIGURING NBMA NEIGHBORS

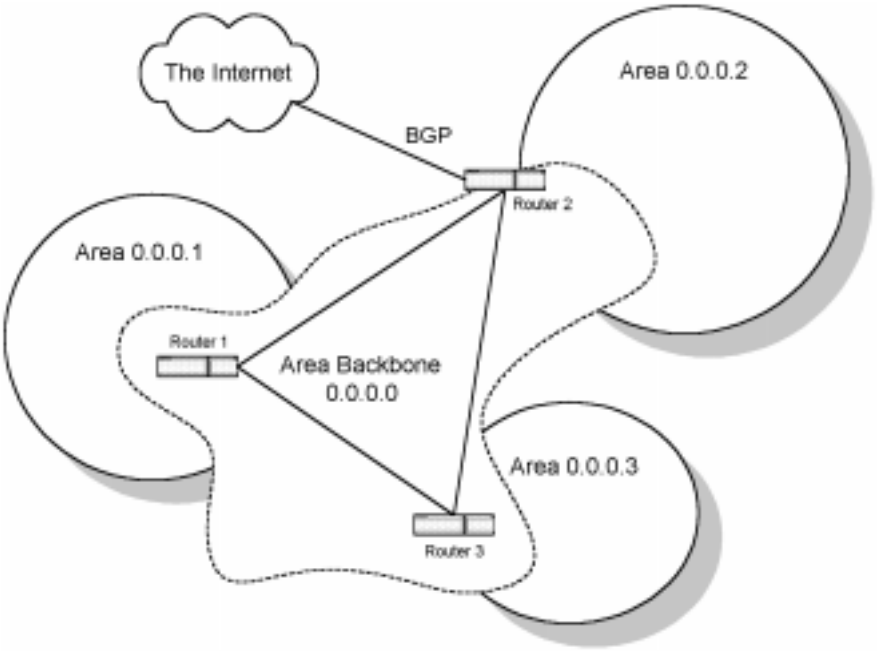
In fully-meshed Frame Relay or ATM networks, you can achieve the most optimal routing by using the NBMA interface type. However, NBMA configuration is more complicated and NBMA router problems are harder to determine. Consequently, point-to-multipoint configuration is preferable in many cases.

For NBMA configuration, you must configure neighbors on all routers eligible to become designated routers. Other routers do not need to be configured with neighbors, but their router priorities should be set to zero to ensure proper operation.

Figure 16 shows a simple network with three routers. In this example, only Router 2 can become designated router, so it is the only router that must be configured with neighbors. Note that when you add neighbors, you must include the unnumbered-interface.0 parameter. As both neighbors are not des-

ignated routers, they must be configured with a priority of 0 (the default is 1). The sample configuration shows the commands that would be entered for each router.

Figure 16 Sample OSPF Neighbor Configuration



Router 1:

```
> config ip ospf.192.1.1.1 router-priority 0
> config ip ospf.192.1.1.1 admin-status disabled
> config ip ospf.192.1.1.1 type nbma
> config ip ospf.192.1.1.1 admin-status enabled
```

Router 2:

```
> config ip ospf.192.1.1.2 admin-status disabled
> config ip ospf.192.1.1.2 type nbma
> add ip ospf neighbor.192.1.1.1 unnumbered-interface.0 priority 0
> add ip ospf neighbor.192.1.1.3 unnumbered-interface.0 priority 0
> config ip ospf.192.1.1.2 admin-status enabled
```

Router 3:

```
> config ip ospf.192.1.1.3 router-priority 0
> config ip ospf.192.1.1.3 admin-status disabled
> config ip ospf.192.1.1.3 type nbma
```

```
> config ip ospf.192.1.1.3 admin-status enabled
```

Note: Although this example shows the OSPF configuration, understanding the Frame Relay or ATM configuration is very important. The designated router on NBMA networks will assume full data link layer connectivity among all of the routers with which it has adjacencies. If this is not the case, black holes can develop.

CONFIGURING POINT-TO-MULTIPOINT NEIGHBORS

In a point-to-multipoint configuration, every router configures the routers with which it has a PVC connection as a neighbor.

The following sample point-to-multipoint configuration is for the network shown in Figure 16. In this example, the route between Router 1 and Router 3 goes through Router 2, even if a PVC between Routers 1 and 3 exists. If a PVC is provisioned later between Routers 1 and 3, simple additional neighbor configuration as shown in this example would allow optimal routing between the two routers. Note that when you add neighbors, you must include the unnumbered-interface.0 parameter.

Router 1:

```
> config ip ospf.192.1.1.1 admin-status disabled
> config ip ospf.192.1.1.1 type pointToMultipoint
> add ip ospf neighbor.192.1.1.2 unnumbered-interface.0
> config ip ospf.192.1.1.1 admin-status enabled
```

Router 2:

```
> config ip ospf.192.1.1.2 admin-status disabled
> config ip ospf.192.1.1.2 type pointToMultipoint
> add ip ospf neighbor.192.1.1.1 unnumbered-interface.0
> add ip ospf neighbor.192.1.1.3 unnumbered-interface.0
> config ip ospf.192.1.1.2 admin-status enabled
```

Router 3:

```
> config ip ospf.192.1.1.3 admin-status disabled
> config ip ospf.192.1.1.3 type pointToMultipoint
> add ip ospf neighbor.192.1.1.2 unnumbered-interface.0
> config ip ospf.192.1.1.3 admin-status enabled
```

REASSIGNING INTERFACES TO DIFFERENT OSPF AREAS

You can change which OSPF area an interface is assigned to by reassigning the area-id for the interface:

```
> config ip ospf.<interface address> area-id <area ID>
```

CONFIGURING OSPF VIRTUAL LINKS

OSPF requires that all areas attach directly to the backbone area. If the attachment is not a physical attachment, the area can be attached via a virtual link. The following scenario is based on the illustration in Figure 15.

ROUTER 4 INTERFACES:

- Interface 1.2.3.4 is connected to backbone area 0.0.0.0.
- Interface 2.3.4.5 is connected to area 0.0.0.3

ROUTER 5 INTERFACES:

- Interface 3.4.5.6 is connected to area 0.0.0.3.
- Interface 4.5.6.7 is connected to area 0.0.0.4

To create a virtual link between router 5 and router 4, use these commands. First, add a virtual interface to router 4 on the interface connected to area 0.0.0.3.

```
> add ip ospf area.0.0.0.3 virtual-interface.2.3.4.5
```

Next, add a virtual interface to router 5 on the interface, specifying connection to OSPF area 0.0.0.3:

```
> add ip ospf area.0.0.0.3 virtual-interface.3.4.5.6
```

These commands create a virtual link from area 0.0.0.4 to backbone area 0.0.0.0. After creating links, examine the virtual neighbors table to see that the links have been established. To view this table, issue this command.

```
> show ip ospf virtual-neighbors summary
```

CONFIGURING STUB AND NOT-SO-STUBBY AREAS

The AP's OSPF implementation supports *normal*, *stub* and *not-so-stubby* areas. These area designations and the parameter settings associated with them provide a way to control importation of externally-learned routes, summary LSAs, and Type-7 LSAs.

The default value for all OSPF areas is normal. The default for the import-as-extern parameter is *importExternal*. You do not need to alter these settings, unless you need to prevent or modify the importation of externally-learned route or summary information.

NOTE *If you want to change the area type, you must disable OSPF first. Therefore, it's a good idea to set the area type at creation time.*

ABOUT THE IMPORT-AS- EXTERN PARAMETER

The import-as-extern parameter is a key parameter that controls whether an OSPF area is a normal area, a stub area, or a not-so-stubby-area.

If *import-as-extern* is set to *importExternal*, the area is a normal area. If it is set to *importNoExternal*, the area is a stub area. If the parameter is set to *importNssa*, it is a not-so-stubby area.

ABOUT NORMAL AREAS

Normal OSPF areas import routing information learned external to OSPF, provided the parameter is-as-border-router is set to true. The following command specifies that an area is normal:

```
> add ip ospf area.<area ID> import-as-extern importExternal
```

ABOUT STUB AREAS

A stub area is an area that does not import externally learned routing information. For these areas, the import-as-extern parameter must be set to *importNoExternal*, as follows:

```
> add ip ospf area.<area ID> import-as-extern importNoExternal
```

ABOUT NOT-SO-STUBBY-AREAS

A not-so-stubby-area (NSSA) can import externally-learned routing information, and does allow such information to propagate within the area. For these areas, the import-as-extern parameter must be set to *importNssa*, as shown

here:

```
> config ip ospf area.<area ID> import-as-extern importNssa
```

This allows externally-learned route information to transit the area in Type-7 LSAs.

NOTE *Not-So-Stubby-Areas are not supported for releases prior to Version 2.2, and Type-7 to Type-5 translation is not supported in Version 2.2.*

CONFIGURING STUB AREA AND NSSA METRICS

If you have more than one area border router for a stub area or NSSA, you can control which exit point is used by changing the metric area border routers place in default summary LSAs sent into stub areas. The default metric (or cost) is 1 for all links, and you must include the type-of-service.0 (only TOS 0 is currently supported). Use the following command:

```
> config ip ospf stub-area.0.0.2.3 type-of-service.0 metric 4
```

You can show the metrics for all of the stub areas with the command:

```
> show ip ospf stub-areas summary
```

NOTE *For this command to succeed, the area must already be created and configured as a stub area.*

BLOCKING IMPORT OF SUMMARY LSAS

If you do not want a stub area or NSSA to import summary LSAs, you must verify that the *summary-lsa-propagation* parameter is set to *noAreaSummary*; otherwise, you should set it as shown here:

```
> add ip ospf area.<area ID> summary-lsa-propagation noAreaSummary
```

CONFIGURING AREA AGGREGATES

You can configure OSPF to allow subnet routing information to be aggregated into summary-LSAs (used by normal and stub areas) or type-7-LSAs (used by not-so-stubby-areas). The range of addresses in an area aggregate will be advertised to other areas in a single LSA.

You can use the effect parameter to hide IP address ranges from other OSPF areas. A sample command is shown below.

```
> add ip ospf area.0.0.0.1 area-aggregates type.summaryLink \
  network.1.2.0.0 mask.255.255.0.0 effect doNotAdvertiseMatching
```

SETTING INTERFACE COST

Each OSPF interface has a cost associated with it. This cost is the metric used in the Shortest Path First calculation. By default, this cost is set to 1 for all links. You can modify this cost value to control the topology through the metrics table for each interface (only ToS 0 is currently supported).

To view the current value, use this command:

```
> show ip ospf.198.202.232.155 metrics summary
```

```
                    IP OSPF.198.202.232.155 METRICS SUMMARY
   INTERFACE ID| TYPE OF SERVICE| Value| Status|
---------------|----------------|------|-------|
198.202.232.155|               0|     1| active|
```

To modify this value, use this command:

```
> config ip ospf.198.202.232.155 metric.0 value 20
```

```
                    IP OSPF.198.202.232.155 METRICS SUMMARY
   INTERFACE ID| TYPE OF SERVICE| Value| Status|
---------------|----------------|------|-------|
198.202.232.155|               0|    20| active|
```

The value 20 will now be used as the cost of this OSPF interface. Keep in mind that a higher cost indicates that a link is less (rather than more) preferred.

DEBUGGING OSPF CONNECTIVITY PROBLEMS

Use this command to show connected IP OSPF interfaces and their status.

```
> show ip ospf neighbors summary
```

This command shows the OSPF link state database for all areas.

```
> show ip ospf areas link-state-db summary
```

Issue this command to show routes that have been imported from outside of OSPF.

```
> show ip ospf external-link-state-db summary
```

CONFIGURING OSPF AUTHENTICATION

You can configure OSPF to authenticate routing updates (LSAs) so only trusted routers within the AS share routing information. By default, routing updates are not authenticated. Two methods of authentication are available:

- simplePassword
- md5

To determine if authentication is configured for OSPF, enter the following command:

```
> show ip ospf.<inst> config
```

```
                                IP OSPF.* CONFIG

   INTERFACE ID|                  Area ID|                      Type|
               |              Admin Status|           Router Priority|
               |    Authentication Type|         Authentication Key|
               |  Multicast Forwarding| Demand Procedure Support|
---------------|----------------------|--------------------------|
           ip.1|               0.0.0.0|              pointToPoint|
               |              disabled|                         1|
               |                  none|                        ""|
               |             multicast|                     false|
```

USING SIMPLE PASSWORD AUTHENTICATION

Using a simple password, which is sent in the clear, you can add a measure of security that can prevent routers from accidentally joining the routing domain. Be aware, however, that simple password authentication is vulnerable to attack. To configure authentication using a password, enter the *authentication-type*, *simplePassword*; then enter the *authentication-key*, a clear password up to 8-characters long:

```
> config ip ospf.1 authentication-type simplePassword \
  authentication-key 8charpwd
```

USING MD5 AUTHENTICATION

For a higher level of security, you can specify MD5 authentication. Receiving routers authenticate update messages from other routers using the MD5 hash algorithm. In order for MD5 authentication to work, both the receiving and transmitting routers must have the same MD5 key (16 bytes) defined. The keys are defined for each OSPF interface, including any virtual interfaces. If MD5 authentication is enabled on an interface, the interface only accepts routing

updates if MD5 authentication succeeded, otherwise they are rejected.

NOTE *The Key ID field for authentication-type md5 is hardcoded to 1. If you are using equipment from other vendors, you must configure the Key ID field of headers from those systems to 1 to interoperate with the AP. The Key ID field identifies the algorithm and secret key used to create the message digest appended to the OSPF packet.*

To configure authentication using MD5, enter the *authentication-type*, *md5*, and the *authentication-key*, the 16-byte key:

```
> config ip ospf.1 authentication-type md5 authentication-key 16bytekey
```

CONFIGURING ISIS

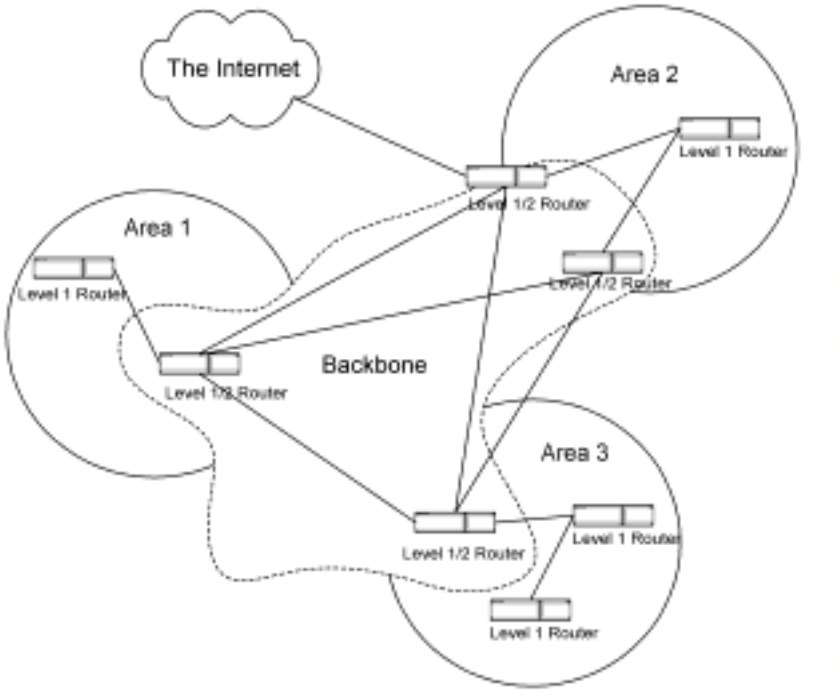
OVERVIEW

Intermediate system-to-intermediate system (ISIS) is a link-state hierarchical protocol similar to OSPF and is an OSI protocol. In this implementation, the Access Point system functions as an IP-only ISIS router. An intermediate system (IS) or router forwards link-state update messages to all neighboring ISs about changes in the network. Sharing this information gives ISs a common view of the network, allowing them to calculate the shortest path to a destination. As a result, ISIS need not send portions of its routing table to update its neighbors. Instead, link-state information is used to maintain a consistent topological view among participating routers.

An area is a collection of contiguous networks and attached hosts. A domain is a group of interconnected areas. Routing within an area is called level 1 routing. Routing between level 1 areas is called level 2 routing; this forms the domain backbone. Level 1 routers know the topology in their area. Level 1 routers do not know about destinations outside their area; they only know about the nearest level 2 router. A level 1 router forwards traffic for destinations outside of its area to a level 2 router in its area. Level 2 routers know the level 2 topology; they do not know about level 1 topology unless it is also participating as a level 1 router.

Figure 17 show components in a sample domain. There are three areas interconnected by a backbone of level 2 routers.

Figure 17 ISIS Domain



SUMMARY OF KEY POINTS

Following is a list of some key points to remember about ISIS:

- *The AP functions as an IP-only ISIS router. ISIS uses the IP forwarder and is configured like other IP protocols.*
- *You must assign a system ID.*
- *You must configure the area the system belongs to in the manual-area-address table.*
- *You must enable ISIS on the system and on every interface over which it will operate.*

CONFIGURING ISIS

To configure ISIS follow the steps listed below:

1. Assign a 6 octet system ID
2. Configure area addresses
3. Configure the routing level (level 1 or level 2)
4. Summarize addresses (if your AP is an area border router)
5. Enable ISIS
6. Enable ISIS on the IP interface

ASSIGNING A SYSTEM ID

The system ID is six octets long and must be unique. You can only configure a system ID when ISIS is disabled. Assign the system ID as follows:

```
> config ip isis system-id cafecafe0077
```

To see that you configured the system ID correctly, enter the following command:

```
> show ip isis config
```

```
                                IP ISIS CONFIG

                                  Mode* = enabled
                                 State  = up
                              System ID* = cafe.cafe.0077
                         Max Path Splits* = 2
           Minimum LSP Transmit Interval* = 5
         Maximum LSP Generation Interval* = 900
Minimum Broadcast LSP Transmit Interval* = 33
          Complete SNP Transmit Interval* = 10
           Partial SNP Transmit Interval* = 2
                  Maximum area addresses* = 3
         Minimum LSP Generation Interval* = 10
                           ES Hello Rate* = 50
                              Wait Time*  = 60
                         DR Hello Timer*  = 1
                         Max Area Check*  = false
                  Level 1 Database State  = enabled
                  Level 2 Database State  = enabled
                 Level 1 LSP Buffer Size* = 1492
                 Level 2 LSP Buffer Size* = 1492
```

CONFIGURING AN AREA ADDRESS

You need to configure each area to which an AP belongs in the manual area address table. Area addresses are represented as a hexadecimal octet string. In general, an AP requires only one area address. Additional area addresses are

required when merging two or more areas. The AP supports up to three area addresses. Add an area address as follows:

```
> add ip isis manual-area-address.490002
```

You can display all configured area addresses with the show command:

```
> show ip isis manual-area-addresses summary
```

```
                    IP ISIS MANUAL-AREA-ADDRESSES SUMMARY

     AREA ADDRESS| Row Status|
-----------------|-----------|
           490002|     active|
```

CONFIGURING THE ROUTING LEVEL

ISIS defaults to level-1-2 operation. If you want the system to operate as a level 1 or level 2 router only, use the *system-type* parameter as follows:

```
> config ip isis system-type l1
```

SUMMARIZING ADDRESSES

You can configure the level-1-2 (area border) routers to summarize level-1 routes into the backbone. Summarization allows multiple addresses and sub-masks to correspond to a single entry in the level2-summary-address table. The following command summarizes address 10.0.0.0 and sets its cost to 20.

```
> add ip isis level2-summary-address.10.0.0.0 mask.255.0.0.0 default-metric
  20
```

To view the configured level-2 summary addresses, use the *show* command.

```
> show ip isis level2-summary-addresses summary
```

ENABLING ISIS

ISIS is disabled by default. To enable ISIS, use the following command:

```
> config ip isis mode enabled
```

NOTE *ISIS will not become enabled until the system ID and at least one area address have been assigned.*

ENABLING ISIS ON THE IP INTERFACE

ISIS is disabled on the IP instance by default. You must enable ISIS on every interface over which ISIS operates. A single ISIS interface is created regardless of the number of IP addresses assigned to the IP instance. By default, ISIS interfaces operate in level-1-2 mode and allows level-1 and/or level-2 adjacencies to be established. If neccessary, configure an interface to allow only level-1 or level-2 adjacencies by changing the value of the *circuit-type* parameter. If *circuit-type* and *system-type* are not consistent, no adjacencies are established on the interface. (See “Configuring the Routing Level” on Page 214 for information about the *system-type* parameter.)

```
> config ip.1 isis circuit-type l1
```

Before enabling ISIS, you must configure the interface with a numbered or unnumbered IP address. To enable ISIS on the IP instance, use the following command:

```
> config ip.1 isis mode enabled
```

Use the following command to see that the interface is enabled:

```
> show ip.1 isis summary
```

```
                                IP.1 ISIS SUMMARY

                    Mode* = enabled
                   State  = up
           Circuit Type* = 12
     Subnetwork Type  = ptToPt
   Level1 Circuit Id  = cafe.cafe.2524.00
Level1 Designated Is  = cafe.cafe.2524
   Level2 Circuit Id  = cafe.cafe.0077.01
Level2 Designated Is  = cafe.cafe.0077
   Point To Point Id  = cafe.cafe.2524.00
```

DISPLAYING THE ADJACENCY TABLE

The adjacency table lists every adjacency established by the system. To see the adjacencies, use the following command.

```
> show ip isis adjacencies summary
```

```
                              IP ISIS ADJACENCIES SUMMARY

          INTERFACE| ADJACENCY INDEX|            State|    Neighbor Snpa|
                   |                |    Neighbor Type|      Neighbor Id|
                   |                |            Usage|       Hold Timer|
                   |                | Neighbor Priority|                 |
-----------------  |--------------- |-----------------  |-----------------|
                ip.1|              1|               up|               ""|
                   |                |               iS|   cafe.cafe.2524|
                   |                |               12|               30|
                   |                |              127|
```

DISPLAYING THE LINK STATE DATABASE TABLE

The link state database table maintains all link state advertisements generated and received by the system. The link state packets (LSP) level, system ID, sequence number, checksum, and lifetime are individually displayed. The entire PDU, including the header, is also displayed as an octet string. You can use the show command to see the entire database or individual LSPs.

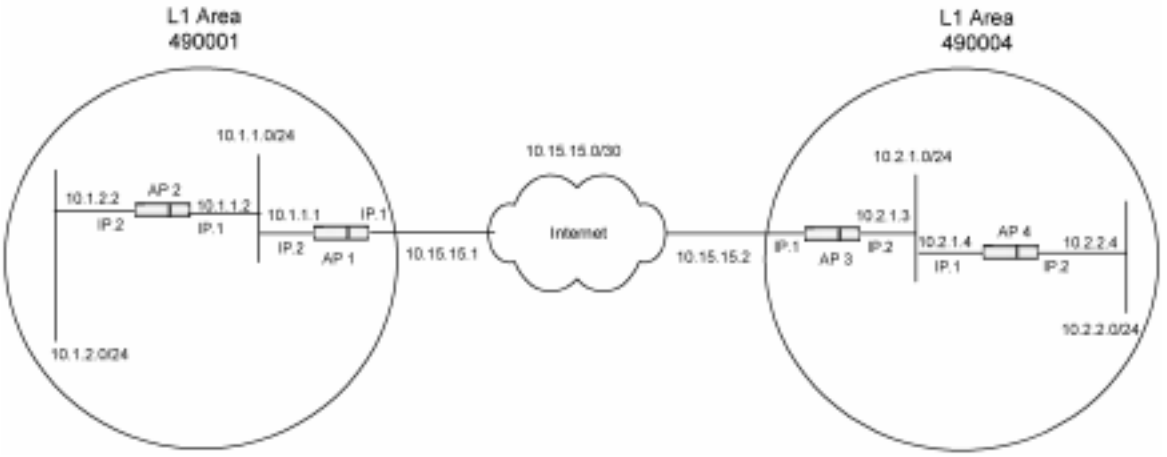
```
> show ip isis link-state-db summary
```

```
                        IP ISIS LINK-STATE-DB SUMMARY

 LSP LEVEL|           LSP ID|  Sequence Number|  Lifetime| Checksum|
          |                 |              Pdu|          |         |
----------|-----------------|-----------------|----------|---------|
         11| cafecafe00770000|         0000025e|      1078|     3bcf|
           |                 |0x831b0106120100>|          |         |
         12| cafecafe00770000|         00000b6f|      1078|     8008|
           |                 |0x831b0106140100>|          |         |
         12| cafecafe25240000|         00000790|      1092|     9927|
           |                 |0x831b0106140100>|          |         |
         12| cafecafe26110000|         00000517|       878|     687e|
           |                 |0x831b0106140100>|          |         |
```

SAMPLE ISIS CONFIGURATION

Figure 18 shows a simple network connecting two ISIS areas over a WAN. Each area contains two APs. By default, AP 1 and AP 3 are configured for level-1-2 operations because they connect out of the L1 area. AP 2 and AP 4 are configured for level-1 operations. In addition, AP 1 and AP 3 each summarize addresses for the areas they serve. The sample commands that follow show how to configure the four APs for the ISIS topology shown in Figure 18.

Figure 18 Sample ISIS Configuration



AP 1

```
> config ip isis system-id 000000000001
> add ip isis manual-area-address.490001
> config ip isis system-type l1l2
> add ip isis level2-summary-address.10.1.0.0 mask.255.255.0.0
> config ip isis mode enabled
> config ip.1 isis mode enabled
> config ip.2 isis mode enabled
```

AP 2

```
> config ip isis system-id 000000000002
> add ip isis manual-area-address.490001
> config ip isis system-type l1
> config ip isis mode enabled
> config ip.1 isis mode enabled
> config ip.2 isis mode enabled
```

AP 3

```
> config ip isis system-id 000000000003
> add ip isis manual-area-address.490004
> config ip isis system-type l1l2
> add ip isis level2-summary-address.10.2.0.0  mask.255.255.0.0
> config ip isis mode enabled
> config ip.1 isis mode enabled
> config ip.2 isis mode enabled
```

AP 4

```
> config ip isis system-id 000000000004
> add ip isis manual-area-address.490004
> config ip isis system-type l1
> config ip isis mode enabled
> config ip.1 isis mode enabled
> config ip.2 isis mode enabled
```

CONFIGURING BGP

OVERVIEW

The Border Gateway Protocol (BGP) is scalable exterior gateway protocol used in TCP/IP networks to determine paths between autonomous systems (AS). Autonomous systems are groups of hosts or networks under a common administration with common routing policies. BGP routes hop-by-hop, from one neighbor or peer to the next, whether that peer is in another AS or in the same AS. BGP peers accomplish this by negotiating a connection between themselves; a series of paths between BGP peers traverses the Internet from source to destination.

BGP works by advertising routes between peer routers to indicate reachable destinations. BGP routers maintain reachable destinations in routing tables, and at start-up, peers exchange routing tables. Subsequently, BGP peers exchange incremental updates about network changes.

BGP imposes no restrictions about the underlying network topology. As a result, autonomous systems can use one or more interior gateway protocols, such as OSPF or RIP, while using BGP to communicate with other AS and possibly within the same AS to another BGP speaking router. BGP running between autonomous systems is known as external BGP (EBGP) and contrasts with internal BGP (IBGP) running between routers within the same AS.

The power of BGP comes from its ability to make decisions at each hop of a path based on the route policy established for each BGP speaker. A BGP speaker can use send and receive policies to filter incoming advertisements, make decisions about routes to propagate, and modify the attributes of those routes before sending an advertisement to a peer. This flexibility, however, adds to BGP's complexity: every BGP router can act differently because of differences in route policies.

The AP supports an RFC 1771 and RFC 1745 compliant BGP implementation tailored to the needs of Internet access networks. The discussion that follows describes BGP configuration within representative access networks. The information presented is not intended as a guidebook or tutorial on BGP and policy routing. Configuring BGP properly requires significant expertise. Familiarize yourself with BGP and routing policies before configuring BGP.

BASIC BGP COMMANDS

The following are the primary BGP configuration commands. These are used to control BGP at the box and peer level. Subsequent sections describe the use of route policy to provide fine control over BGP routing.

ACTIVATING AND DEACTIVATING BGP

To activate BGP, use this command:

```
> config ip bgp mode enabled
```

To deactivate BGP, use this command:

```
> config ip bgp mode disabled
```

CONFIGURING AN AS NUMBER

Use this command to configure an AS number.

```
> config ip bgp local-as <as number>
```

ADDING BGP PEERS

To add a BGP peer, use this command to specify the peer's address. It is not necessary to configure a peer's AS number. This information is automatically learned during peer establishment.

```
> add ip bgp peer.<remote address>
```

EBGP multi-hop is automatically supported for BGP peers. This means that EBGP peers are not required to exist on the same subnet as the local AP interface.

SHOWING BGP PEER INFORMATION

Use this command to show BGP peers.

```
>show ip bgp peers summary
```

Following is a summary screen showing information on BGP peers.

IP BGP PEERS SUMMARY

REMOTE ADDRESS	Identifier Local Address	State Remote AS	Mode Up/Down Time	Negotiated Version Last Update
2.3.4.5	2.3.4.5 0.0.0.0	active 0	start 40	4 0
2.3.4.6	2.3.4.6 0.0.0.0	active 0	start 34	4 0
2.3.4.7	2.3.4.7 0.0.0.0	active 0	start 29	4 0
2.3.4.8	2.3.4.8 0.0.0.0	active 0	start 14	4 0
2.3.4.9	2.3.4.9 0.0.0.0	active 0	start 8	4 0

DELETING BGP PEERS

To delete specific bgp peers or all peers, use one of these commands.

> remove ip bgp peer.<instance>

> remove ip bgp peers

The AP displays back the results of the remove operations, as shown in this example.

```
vpn-bottom> remove ip bgp peers
Removing 2.3.4.5
        Succeeded.
Removing 2.3.4.6
        Succeeded.
Removing 2.3.4.7
        Succeeded.
Removing 2.3.4.8
        Succeeded.
Removing 2.3.4.9
        Succeeded.
5 component(s) requested to be removed
5 component(s) removed
```

SHOWING BGP ROUTE INFORMATION

Use this command to show BGP route information.

```
> show ip bgp path-attributes general
```

```
IP BGP PATH-ATTRIBUTES GENERAL
IP PREFIX| PREFIX LENGTH| PEER| Next Hop|
                                MED|
                          Local Pref|
                  Atomic Aggregate|
                           Aggr AS|
                         Aggr Addr|
                         Community|
---------|-------------|---------|-----------------------|
3.0.0.0|            8|157.130.6.37|157.130.6.37|
                                -1|
                                -1|
                             false|
                                 0|
                           0.0.0.0|
                                  ""|
```

ROUTE REFLECTOR SERVER

You can use route reflection for servers as well as a clients. This configuration allows one IBGP peer to serve a group of clients with routes from other BGP peers. This is advantageous because you can manage peers that are not fully meshed.

CONFIGURING THE SERVER

When you configure a server, you have the following two options:

Type	Syntax	Function
Full Configuration	config ip bgp route-reflection full	Performs route reflection among peer clients as well as between clients and non-clients
Client and non-client only	config ip bgp route-reflection noClientToClient	Performs route reflection only between clients and non-clients

The default configuration is *config ip bgp route-reflection none*, which means that route reflection is disabled for that AP as a server.

CONFIGURING A CLUSTER

A route reflector and its clients form a cluster. All peers of the route reflector that are not in the cluster are called non-clients. To configure the clients in the same cluster, use the following syntax (listing as many clients as necessary):

```
> config ip bgp route-reflection full
> config ip bgp peer.10.1.44.254 route-reflector-client true
> config ip bgp peer.10.1.44.255 route-reflector-client true
```

When more than one route reflector is serving a cluster, you must configure each server in the cluster with a cluster ID number using this syntax:

```
> config ip bgp cluster-id <number>
```

All the route reflector servers in a cluster must use the same cluster ID, and each separate cluster must have a unique ID number.

OVERVIEW OF BGP ROUTE POLICY

BGP is a routing protocol that supports advertisement, reception, and modification of attributes associated with specific routes. These capabilities enable administrators to use BGP to achieve very specific routing objectives. The protocol's great flexibility and ability to provide intelligent control of routing information have made it the protocol of choice on the Internet.

This fine-grained control over the routing process is called "route policy" and provides a flexible, generic configuration and management component. This enables administrators to implement route policies for OSPF and RIP, in addition to BGP. Route policy is described in detail in "Route Policy" on Page 227. This section also provides examples for configuring route policy to achieve specific BGP routing goals.

DEFAULT BGP BEHAVIOR

Before configuring BGP, it is important to know how the AP's route policy framework operates on BGP routes when no policies are configured. Following are default send and receive policies in force when BGP has been activated and one or more peers have been created.

DEFAULT SEND ---> NONE

By default, no routes are advertised to peers, regardless of the peer state. This protects against accidental advertisement of routes which might direct traffic in undesired directions. Route advertisement in the Internet (that is, over BGP) should be a very deliberate process undertaken with a thorough understanding of the desired routing goals and the expected results.

DEFAULT RECEIVE ---> ALL

By default, all routes advertised by a peer are received by BGP and entered into the route selection process. With proper route advertisement policy configuration, default reception of all routes allows routing to occur with a minimum of configuration.

BGP ROUTE SELECTION

BGP uses a decision process to choose which routes it should add to the IP route table. BGP compares all routes that it has learned for the same destination and uses the values of the attributes listed below to choose the best route. If two or more routes have an attribute with the same value, BGP proceeds down the list of attributes searching for a value that determines the best route. BGP starts by choosing the route with the,

1. Highest local preference

   If local preferences are the same, BGP chooses the route with the,

2. Shortest AS path

   If AS paths are the same, BGP chooses the route with the,

3. Lowest origin type

   If origin types are the same, BGP chooses the route with the,

4. Lowest MED

   If MEDs are the same, BGP chooses the route with the,

5. Lowest next hop cost

   If next hop costs are the same, BGP chooses the route with the,

6. External peer over internal peer

   If these are the same, the BGP sees the routes as equal and adds each to the IP route table so that equal cost load balancing can take place.

COMPLEX BGP ROUTE POLICY

Following are BGP policy capabilities provided through the AP's generic route policy framework:

RECEIVE

Following are available BGP receive policy actions.

Receive Action	***Meaning***
Accept Routes	Allow routes to be accepted for consideration into the routing table.
Drop (filter) routes	Do not allow routes to be accepted for consideration into the routing table.
Filter by Community	Check received community list and filter based on contents.
Filter by AS Path	Check received AS path list and filter based on contents.
Set local preference	Set the local preference to be used internally and (possibly) readvertised for this route.

SEND

Following are available BGP send policy actions.

Send Action	***Meaning***
Send routes	Allow routes to be advertised to a peer.
Never send routes	Do not allow routes to be advertised to a peer.
Aggregate routes	Create and advertise an aggregate route for a more specific route.
Import routes	Allow routes learned through other routing protocols to be sent in BGP.
Set Multi-Exit Discriminator (MED)	Include the MED attribute set to a desired value, possibly set based on the metric from an imported protocol.
Set Community	Add a community attribute or append a value if one already exists.
Set next-hop-self	Set the next hop of a route to this router for EBGP received routes.
Prepend AS	Add a specific number of AS elements to the AS path for this route.

See “Route Policy” on Page 227 for information on configuring these capabilities.

SUPPORT FOR AS PATH REGULAR EXPRESSIONS

You can configure a regular expression to match specific AS strings that appear in received BGP updates. Up to 32 characters per expression are allowed. Table 14 explains supported regular expression characters and their meanings.

Table 14 Characters Permitted in Regular Expression Matching

Characters	Meaning
.	Matches any single character.
^	Matches the null string at the start of the input string.
$	Matches the null string at the end of the input string.
x	Matches the character x.
[*chars*]	Matches any single character that appears in *chars*, according to the following criteria: • If the first character of *chars* is ^, the pattern matches any single character not in the remainder of *chars*. • A sequence in the form A-B in *chars* is treated as shorthand for all of the ASCII characters between A and B. • If the first character in *chars* (possibly following a ^) is], it is treated literally (as part of *chars*, rather than as a terminator). • If a dash (-) appears first or last in *chars*, it is treated literally.
(*regexp*)	Matches anything that matches the regular expression regexp. The parentheses are used for grouping and identifying pieces of the matching sub-string.
*	Matches a sequence of 0 or more matches of the preceding atom.
+	Matches a sequence of 1 or more matches of the preceding atom.
?	Matches either a null string or a match of the preceding atom.
regexp1|*regexp2*	Matches anything that matches either *regexp1* or *regexp2*.

The following examples show how certain regular expressions are used:

LOCAL AS PATH MATCH

```
> add ip policy-receive.bgp name.receive_loc autonomous-system "^$"
```

MATCHING ONLY 300

```
> add ip policy-receive.bgp name.receive_300 autonomous-system "^300$"
```

MATCHING 500 OR 600

```
> add ip policy-receive.bgp name.receive_500_600 autonomous-system
  "500|600"
```

ROUTE POLICY

OVERVIEW

Route policy is the mechanism used to provide fine-grained control over IP routing as a whole in the AP. The route policy framework provides a set of configuration and monitoring capabilities which can be used to control routing in a single routing protocol or be used across routing protocols to provide very sophisticated interactions among RIP, OSPF, ISIS, and BGP.

CONTROL OF INDIVIDUAL ROUTING PROTOCOLS

Route policy can be used to provide control within a single routing protocol. For instance, a system administrator can use policy configuration to allow BGP to advertise only routes received from other BGP peers, and only receive a default route from a single peer. In this situation, policy configuration only needs to be entered on behalf of BGP.

CONTROL OF ROUTING PROTOCOL INTERACTION

Route policy can also be used to control interaction between two routing protocols. This is typically used to import or export routes from one protocol to another. For example, if a system administrator wanted to send the default route learned through BGP into OSPF, an OSPF send policy would be configured to look for the default route that BGP provided and would only send it if such a route was found in the routing table.

ROUTE POLICY COMPONENTS

A single *route policy* contains *matching* information and *action* information. If a route matches the matching information, the resulting action is invoked. There are two basic policy types: send and receive policies. Route policies affect the sending and receiving of route information, at times modifying route

information prior to sending it out.

SEND POLICIES

Send policies apply to route information sent to peer or neighbor routers. These policies determine which routes are propagated by a specific protocol. BGP send policies can also modify routes as they are sent to peers.

RECEIVE POLICIES

Receive policies apply to route information received from a neighbor or peer. Receive policies determine what routes are accepted for the specified protocol.

POLICY MATCHING

During send and receive operations, the AP evaluates route information against the match criteria configured for each policy. For each routing protocol, the goal is to determine whether a configured policy matches the route information.

The AP supports RIP, OSPF, ISIS, and BGP routing protocols. For each protocol, evaluation can include the following matching criteria:

- Source protocol - the protocol through which the route was learned
- Peer (See Peer Matching, Page 229, for more detailed information)
- Address start
- Address end
- Prefix-min
- Prefix-max

In addition to these criteria, evaluation of BGP routes also includes:

- AS number
- Community

THE PROTOCOL INDEX AS A MATCH CRITERION

While it is not a match criterion, the protocol index also has a role in route matching. It is part of the matching information because a policy will only match routes that use the protocol specified by the policy's index.

PEER MATCHING

Matching based on peer information differs for RIP, OSPF, ISIS, and BGP protocols. Table 15 explains these differences.

Table 15 Peer Matching Per Protocol

Routing Protocol	***Peer Matching Actions***
RIP	• neighbor source address (receive) • interface address (send) • peer group
OSPF	• wildcard
ISIS	• wildcard
BGP	• peer address • wildcard • peer groups

To see the matching criteria in force for a protocol, issue one of the following commands:

> show ip policy-send-table match-config

> show ip policy-receive-table match-config

```
                        IP POLICY-RECEIVE-TABLE MATCH-CONFIG
PROTOCOL|          NAME|              Peer|     Address Start|       Address End|
        |              | Autonomous System|         Community|        Row Status|
        |              |        Prefix Min|        Prefix Max|                  |
--------|--------------|------------------|------------------|------------------|
     bgp|         first|          1.2.3.8|           0.0.0.0| 255.255.255.255|
        |              |                ""|               0:0|            active|
        |              |                 0|                32|                  |
     bgp|        second|          0.0.0.0|           0.0.0.0| 255.255.255.255|
        |              |                ""|               0:0|            active|
        |              |                 0|                32|
```

PEER GROUPS

Peer grouping is a mechanism that allows a number of peer addresses (identified by IP address) to share the same policy definitions. Up to 32 peer groups are currently supported. Routers made by other manufacturers can be included with the AP in the same peer group.

ADDING AND REMOVING PEER GROUPS AND MEMBERS

You cannot add a peer group without adding a member to it. To add a peer group and add a member to the group, use the following command:

```
> add ip peer-group.[1 -32] address.[peer ip address]
```

For example, you could add two peer groups, each with different member routers, as follows:

```
> add ip peer-group.1 address.198.202.232.11
```

```
> add ip peer-group.2 address.198.200.21.21
```

Similarly, you can remove one or all members from a peer group using the *remove* command as shown here:

```
> remove ip peer-group.2 address.*
```

The interface will display information indicating that one or more instances have been removed. If you remove all members of a group, the group ceases to exist.

ASSOCIATING A POLICY DEFINITION WITH A PEER GROUP

You can associate policy definitions with peer groups. To do this, create a send or receive policy and add peer group information as shown in this example.

```
> add ip policy-send.bgp name.local_send peer 0.0.0.2 send-action normal
```

This command creates a send policy (local_send) that is shared by all routers in peer group 2.

Note that when associating a peer group with a policy, the parameter peer is used to identify the peer group. The group number is represented in the form of an IP address. In this example, the peer group 2 (originally added as peer-group.2) is represented as peer.0.0.0.2.

SHOWING PEER GROUP INFORMATION

You can access information on the IP addresses associated with a peer group. To do this, issue this command:

```
> show ip peer-group-table config
```

```
                        IP PEER-GROUP-TABLE CONFIG

 PEER GROUP|         ADDRESS| Row Status|
-----------|----------------|-----------|
          3|      20.1.101.2|     active|
          3|      20.1.101.3|     active|
          3|      20.1.230.2|     active|
          3|      20.2.102.2|     active|
          3|      20.2.102.3|     active|
          3|      20.2.103.2|     active|
          3|      20.2.104.2|     active|
          3|      20.2.210.2|     active|
          3|      20.2.211.2|     active|
```

The following *show* commands are also applicable:

```
> show ip peer-group.*
```

```
> show ip peer-group.<number> address.<ip address>
```

ROUTE POLICY ACTIONS

Route policy actions affect the sending and receiving of routes. In some instances, route policies also modify route parameters while advertising them. Basic policy actions are common to all supported protocols; a few actions are specific to BGP.

Table 16 shows route policy actions common to all routing protocols.

Table 16 Common Route Policy Actions

Action Type	*Common Policy Actions*
receive-action	• drop Do not receive the route into the protocol. • receive Accept the route into the protocol. • ignore Process the actions in the route policy, but continue the policy search.
send-action	• never Never send the route to peers. • ignore Process the actions in the route policy, but continue the policy search. • normal Send the route to peers. Note: Route import from one protocol to another is achieved with a send-action normal in a protocol's policy configuration with a src-prot (matching criteria) of a different protocol.

RIP-SPECIFIC CAPABILITIES

By default, all routes learned through RIP are advertised on all circuits. However, the AP's implementation of RIP send policies causes RIP to behave differently when explicit RIP send policies are in effect.

The AP's implementation of RIP policies uses the outgoing interface address (that is, the address to which RIP has been added) as the peer address for peer matching because RIP lacks a concept of peer. This differs from BGP, which has a very specific notion of peer.

In addition, the following rules apply:

- *If no send policy exists for local RIP interface addresses (that is, src-prot is local), local routes are sent out over all the RIP circuits under control of the RIP interface configuration. In this case, RIP route policy does not apply to these routes.*

- *If an explicit send policy exists for local RIP interface addresses and the route's address range and peer match, the action of the policy determines the result, rather than the local interface configuration.*

All other RIP route advertisement occurs according to common route policy operations, as explained in "Route Policy" on Page 227 and following.

OSPF-SPECIFIC CAPABILITIES

Because the OSPF link state update process and shortest path first route selection algorithm requires the databases on all routers in the network to have the same contents, complicated send and receive policies are not recommended. Therefore, no OSPF-specific send actions are defined.

OSPF supports route import from other protocols through send policies, and that operation is part of the common send policy described in Table 16.

See "Configuring OSPF" on Page 196 for information on configuring OSPF.

ISIS-SPECIFIC CAPABILITIES

Because the ISIS link state update process and shortest path first route selection algorithm requires the databases on all routers in the network to have the same contents, complicated send and receive policies are not recommended. Therefore, no ISIS-specific send actions are defined.

ISIS supports route import from other protocols through send policies, and that operation is part of the common send policy described in Table 16.

See "Configuring ISIS" on Page 211 for information on configuring ISIS.

BGP-Specific Capabilities

Most advanced BGP configuration capabilities are implemented as route policy actions.

Receive Actions

- Local Preference

  Sets local preference. This attribute is used in route calculation for this node and will be transmitted to peers.

Send Actions

- Aggregate Routes

  Allows a less specific aggregate for a route to be sent in the place of one or more specific routes that fall within the address range of the aggregate route. Aggregate is one of the actions of the send-action parameter. As such, it causes an immediate termination of the policy table search.

  The route policy statement shown here sends all routes that fall within the 198.202.232/24 address range (198.202.232.0 through 198.202.232.255) as a single route.

  ```
  >add ip policy-send.bgp name.aggregate198 address-start \
    198.202.232.0 address-end 198.202.232.255 send-action aggregate
  ```

- Multi-Exit Discriminator (MED)

  Creates and sends a MED path attribute with the specific value in updates to specified external peers. The MED attribute applies to routes sent to remote ASs from the local AS. The lower the MED, the more preferred the route. The value used for MED when no MED is specified is 0 (most preferred).

  See "Using the MED Attribute" on Page 246 for an example that shows how to use the MED attribute.

- AutoMed

  Automatically creates and sends a MED path attribute by adding 1 to the

value of the IGP metric.

- Community

  The AP's implementation of BGP community support complies with RFC 1997. To configure a community, use the send-community parameter as shown here:

  ```
  > config ip policy-send.bgp name.bgpsend peer.189.7.8.9 \
    send-community [community-name]
  ```

where

community-name	Is one of the following values: • noExport • noAdvertise • noExportSubconfed • a number in the range 65536-4294967043

  The text strings *noExport*, *noAdvertise*, and *noExportSubconfed* are well-known names that resolve to 4-byte values. These names have global significance and are implemented in all BGP speakers that implement or have an awareness of the community attribute.

  To define your own community, enter two numbers separated by a colon. These numbers are chosen at the discretion of the administrator. Ordinarily, the first number is based on the AS number of the BGP speaker. The second number can be defined according to policy routing or other criteria used within the AS. For more information, see RFC 1997. An example command follows.

  ```
  > config ip policy-send.bgp name.send_wide peer.189.3.4.5 \
    send-community 1:25
  ```

- Next-Hop-Self

  When a route is received from an EBGP peer, this parameter causes the receiving router to insert its own address (internally reachable within the AS) into subsequent advertisements to other IBGP peers, in place of the next hop of the EBGP peer.

  ```
  > config ip policy-send.bgp name.send_wide peer.189.3.4.5 \
    next-hop-self true
  ```

- AS prepend

  Use AS prepend to add the configured number to the AS path list. The AS

prepend attribute increases the length of the AS path, providing another way to influence route selection by remote ASs. If two routes have the same local preference value, increasing the path length of one can make it less preferred than its alternate.

See “Using AS Prepend” on Page 248 for an example that shows how to use AS prepend.

CONTROLLING POLICY EXECUTION WITH POLICY NAMES

Each policy name is an alphanumeric string with a maximum length of 32 characters. Each string can include a numeric suffix consisting of a decimal point and integer value. The purpose of this suffix is to enable administrators to influence policy execution, while providing the flexibility to use similar names. Where no suffix is included in a policy name, a suffix of .0 is assumed. In determining execution order, a suffix of .0 has priority over any other value. Where a name with a .0 suffix exists, it will execute before policies with suffixes greater than .0, no matter how these policies are named.

In practice, this is simpler than it sounds. The following policy names are listed in order of execution:

- adam
- adam_00
- Zeke
- Zelda
- adam.1
- adam.11
- Ezra.2600

The first policy listed (adam) has no numeric suffix. This is equivalent to a suffix of zero. Because several other policies have .0 suffixes (adam, adam_00, Zeke, and Zelda), these policies execute before any others. Their execution order is determined by the alphabetic sequence of the each name’s affix. That is, adam executes before adam_00, both of which execute before Zeke, and all of which execute before Zelda. Note also that Zeke and Zelda (which have .0 suffixes) execute before adam.1 and adam.11. Ezra.2600 executes last in this list, owing to the size of its suffix.

When naming policies, remember that the AP requires each name to end in a

numeric suffix (such as adam.1). If you name a policy without a suffix (for example, adam), 0 is implied as the numeric suffix. If you name a policy with a non-integer suffix (such as adam.xyz), policy execution order might not match your intentions.

The AP provides *show* commands that display send and receive policies in their order of execution. To see all policies that are in effect, issue one of the following *show* commands:

```
> show ip policy-receive-table action-config|all
```

```
> show ip policy-send-table action-config|all
```

You can issue *show* commands for send or receive policies.

The following display shows a group of BGP policies. These policies are listed in execution order, based on the AP's parsing of all policy names.

```
> show ip policy-send-table action-config
```

IP POLICY-SEND-TABLE ACTION-CONFIG

PROTOCOL	NAME	Send Action Send Community Metric Type Row Status	MED Auto MED Auto IGP Metric	AS Prepend IGP Metric Next Hop Self
bgp	adam	normal 0:0 ospfExtType2 active	-1 off off	0 0 false
bgp	adam_00	normal 0:0 ospfExtType2 active	-1 off off	0 0 false
bgp	adam_01	normal 0:0 ospfExtType2 active	-1 off off	0 0 false
bgp	adam_02	normal 0:0 ospfExtType2 active	-1 off off	0 0 false
bgp	bill_00	normal 0:0 ospfExtType2 active	-1 off off	0 0 false
bgp	bill_01	normal 0:0 ospfExtType2 active	-1 off off	0 0 false
bgp	zeke	normal 0:0 ospfExtType2 active	-1 off off	0 0 false
bgp	adam_01.1	normal 0:0 ospfExtType2 active	-1 off off	0 0 false
bgp	adam_00.25	normal 0:0 ospfExtType2 active	-1 off off	0 0 false

Keep in mind that policies *only* execute if a given peer matches, is a member of a matching peer group, or is the wildcard. Also, if a packet matches a policy and the action is send never or send normal, the search for further policy matches terminates.

CONFIGURING ADMINISTRATIVE DISTANCE

By default, the AP selects routes from different sources in a predetermined order. The AP selects routes on a lowest-to-highest basis, in the following sequence.

- Local 0
- Static 10
- EBGP 20
- OSPF intra 110
- OSPF inter 111
- OSPF external 112
- ISIS Level 1 115
- ISIS Level 2 116
- ISIS external 117
- RIP 120
- IBGP 200
- ICMP 220

Special circumstances may require you to establish a different order for reception of routes. For example, let's say that, for a certain address range, you want to cause BGP routes to be applied first to received packets followed by static routes.

This is the reverse of the default precedence order shown above. You can change this by using the distance parameter as shown here. First, assign a distance value to BGP routes for the range desired:

```
> add ip policy-receive.bgp name.test1 address-start 2.3.4.0 address-end
  2.3.4.255 distance 5
```

Next, add another change for a new receive policy called test2. This policy applies to static routes received from the same address range:

```
> add ip policy-receive.static name.test2 address-start 2.3.4.0 address-end
  2.3.4.255 distance 8
```

This command causes the AP to accept BGP routes for this range before static

routes, since the distances for these protocols are 5 and 8 respectively. Note that, given the default order of precedence for received routes, the second command was not necessary. The AP already assigns a distance of 10 to static routes.

Note that the distance value assigned to local routes is 0. This value cannot be changed. If you configure a receive policy and fail to provide a distance value, the default value of -1 is automatically added. In this case, the policy uses the default value specified for that protocol.

To show the administrative distance for all received routes, use this *show* command:

```
> show ip policy-receive-table action-config
```

```
                    IP POLICY-RECEIVE-TABLE ACTION-CONFIG

PROTOCOL|          NAME|          Peer|Receive Action| Preference|
        |              |      Row Status|      Distance|           |
--------|--------------|----------------|--------------|-----------|
     bgp|         test1|       0.0.0.0|       receive|         -1|
        |              |        active|             5|           |
  static|         test2|       0.0.0.0|       receive|         -1|
        |              |        active|             8|           |
```

USING A PREFIX TO INFLUENCE POLICY EXECUTION

Prefix matching criteria allow further refinement in determining whether to execute a policy action. When min and max prefix values are added to a policy, the policy's action only executes if the IP address is within the prefix length range specified. A prefix range can be set for both send and receive policies. The range of the prefix length can be set with the prefix-min and prefix-max configuration options. Values range from 0 to 32. Following are examples:

In this case, the policy action will be executed if the prefix of an evaluated packet is less than or equal to 24.

```
> config ip policy-send.bgp name.new-policy prefix-min 0 prefix-max 24
```

Likewise, this definition applies to packets with a prefix that is greater than or equal to 24. The upper limit of 32 is the highest prefix value possible.

```
> config ip policy-send.bgp name.new-policy.01 prefix-min 24 prefix-max 32
```

In this case, the policy action executes if the prefix is greater than or equal to

24 and less than or equal to 29.

```
> config ip policy-send.bgp name.new-policy.02 prefix-min 24 prefix-max 29
```

SAMPLE ROUTE POLICY SCENARIOS

Following are three sample scenarios that show how route policies can be implemented in representative access networks. The samples include:

- A RIP POP network (Page 241)
- A BGP stub network (Page 244)
- A BGP multihomed network with MED and local preference (Page 245)
- A BGP multihomed network with communities and AS prepend (Page 247)

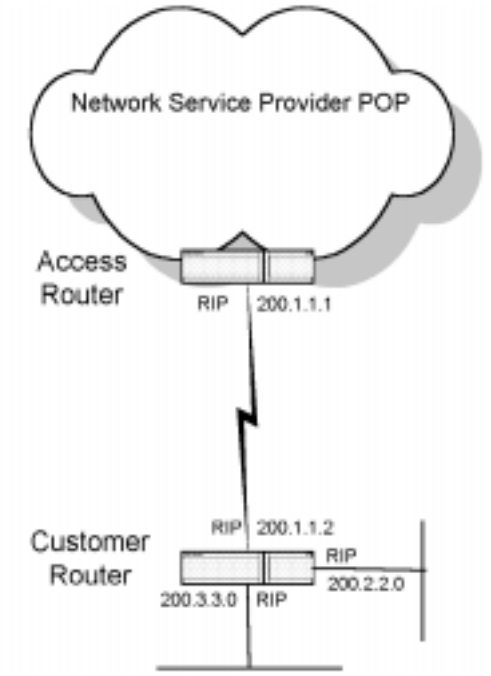
NOTE *In each sample scenario, it is assumed that IP addresses have already been added, and that other required configuration has been performed.*

A RIP POP NETWORK

The basic intent of this configuration is to configure the AP in the network provider's POP to control the advertisement of RIP routes to the customer router. To this end, the configuration uses route policies to accomplish the following:

- Only the default route is advertised to the customer. All other routes are sent to the internal RIP network.
- Only local routes are received from the customer.

Figure 19 RIP Policy Route Example, Network Provider POP



SEND POLICIES

The policy definitions that follow assure that only a default route to 200.1.1.1 (the Network POP) is advertised. Other routes (for instance, to destinations within the network provider's network) are not sent.

Matching is performed on the peer, which for RIP send policies is the IP address of the interface from which the update is sent. The IP address range of 0.0.0.0 - 0.0.0.0 indicates that this route is the default. Note also that BGP is provided as the source protocol. To import the route from some other protocol (such as OSPF) would require a separate policy.

```
> add ip policy-send.rip name.riptx10 peer 200.1.1.1 address-start 0.0.0.0 \
  address-end 0.0.0.0 src-prot bgp send-action normal
```

The policy statement that follows is a safety measure, designed to prevent the advertisement of routes. It will only be effective if the previous send policy were removed. The IP address range specified assures that the route will match any IP address.

```
> add ip policy-send.rip name.riptx20 peer 0.0.0.0 address-start 0.0.0.0 \
```

```
address-end 255.255.255.255 send-action never
```

RECEIVE POLICIES

These policy statements assure that only local routes are received from the customer. The source address of the RIP updates match the value specified for peer (200.1.1.2) and the address ranges match the customer's local interface addresses.

```
> add ip policy-receive.rip name.riprx10 peer.200.1.1.2 address-start \
  200.2.2.0 address-end 200.2.2.0 receive-action receive

> add ip policy-receive.rip name.riprx20 peer.200.1.1.2 address-start \
  200.3.3.0 address-end 200.3.3.0 receive-action receive
```

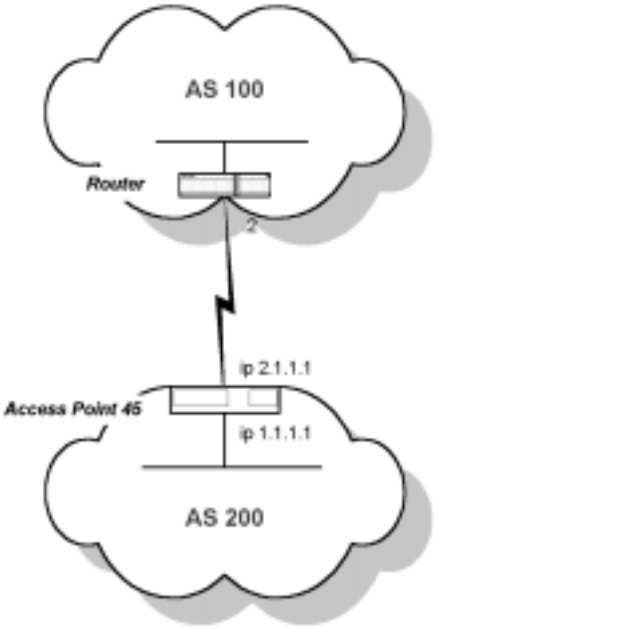
The policy that follows provides another safety measure. It assures that no non-local routes will be received on interface 200.1.1.2. This policy would not take effect unless both of the preceding receive policies were disabled.

```
> add ip policy-receive.rip name.riprx90 peer 200.1.1.2 receive-action drop
```

A BGP STUB NETWORK

Figure 20 shows a basic stub network. In this scenario, BGP is used to accomplish the same goals as static routes. The benefits are greater scalability and consistent provisioning of routing resources.

Figure 20 BGP Stub Network



The following commands show how to configure BGP. Begin by configuring a local AS.

```
> config ip bgp local-as 200
```

Next, add a BGP peer (router interface 2.1.1.2), and configure a send policy.

```
> add ip bgp peer.2.1.1.2
```

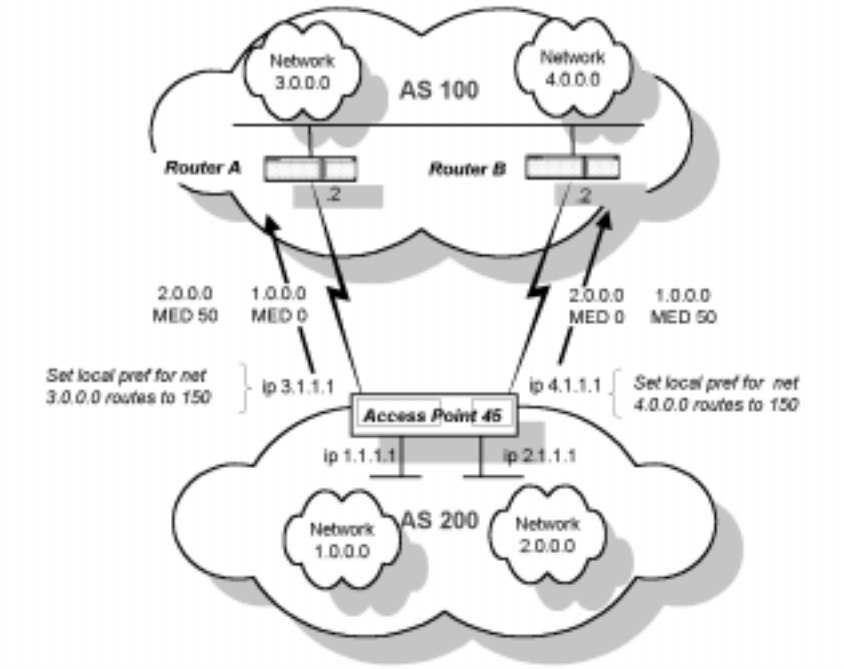
```
> add ip policy-send.ospf name.<name> peer.0.0.0.0 src-prot.bgp \
  send-action aggregate
```

In this example, BGP speaker 2.1.1.1 has received a default route for AS 100 that is imported from BGP into OSPF. The configuration has the additional effect of exporting a default route into OSPF (on eth.1/ip.1) when BGP is active.

A BGP MULTIHOMED NETWORK WITH MED AND LOCAL PREFERENCE

The multihomed network shown in Figure 21 illustrates redundancy and load balancing to the same external AS using the local preference and MED (Multi-exit Discriminator) attribute actions of a policy match. In this scenario, the AP in AS 200 peers with two border routers in AS 100.

Figure 21 BGP Multihomed Network



First, configure a local AS (in this case, 200). Then add BGP peers.

```
> config ip bgp local-as 200
> add ip bgp peer.3.1.1.2
> add ip bgp peer.4.1.1.2
```

The two sections that follow explain how the local preference and MED attributes are used to establish send and receive policies.

USING THE LOCAL PREFERENCE ATTRIBUTE

The local preference attribute applies to routes sent from a remote AS to a local AS.

In this example, the attribute specified in the receive policy filters is 150. This is higher than the default local preference value (100), and marks each route as strongly preferred (for example, over a default value). The statements thus mark the link to router A (BGP peer 3.1.1.2) as preferred (preference set to 150) for routes going to network 3.0.0.0. The link to router B (BGP peer 4.1.1.2) is preferred for routes going to network 4.0.0.0.

```
> add ip policy-receive.bgp name.<name> peer.3.1.1.2 address-start 3.0.0.0 \
  address-end 3.255.255.255 preference 150

> add ip policy-receive.bgp name.<name> peer.4.1.1.2 address-start 4.0.0.0 \
  address-end 4.255.255.255 preference 150
```

USING THE MED ATTRIBUTE

In this case, the MED values determine that the link to router A (network 3.0.0.0) is preferred (med set to 0) for traffic destined for network 1.0.0.0. This same link is less preferred (med set to 50) for traffic to network 2.0.0.0.

```
> add ip policy-send.bgp name.<name> peer.3.1.1.2 src-prot local \
  address-start 1.0.0.0 address-end 1.255.255.255 med 0

> add ip policy-send.bgp name.<name> peer.3.1.1.2 src-prot local \
  address-start 2.0.0.0 address-end 2.255.255.255 med 50
```

In these statements, the MED values determine that the link to router B (network 4.0.0.0) is preferred (med set to 0) for traffic destined for network 2.0.0.0. This same link is less preferred (med set to 50) for traffic to network 1.0.0.0.

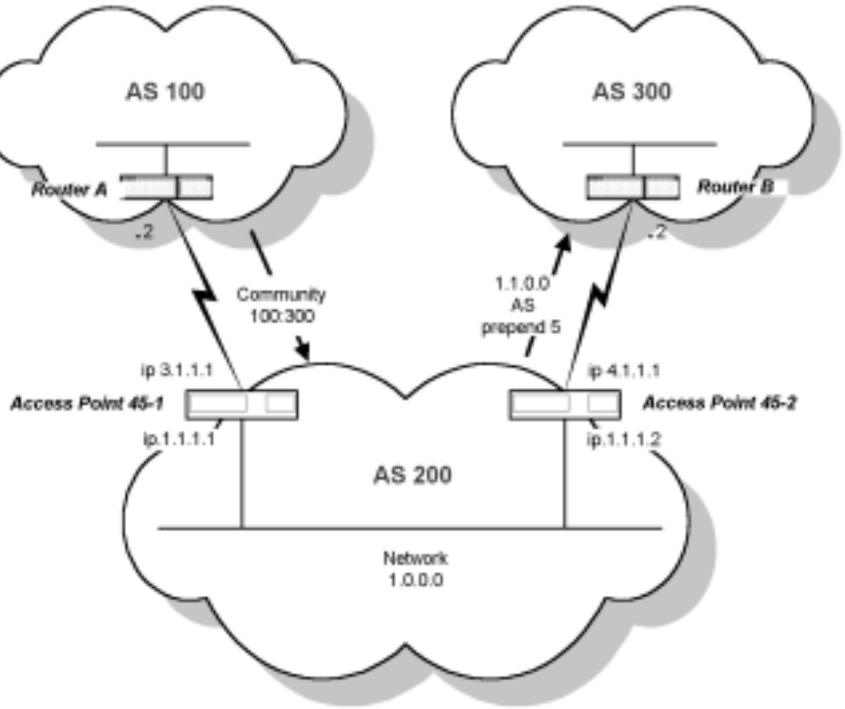
```
> add ip policy-send.bgp name.<name> peer.4.1.1.2 src-prot local \
  address-start 2.0.0.0 address-end 2.255.255.255 med 0

> add ip policy-send.bgp name.<name> peer.4.1.1.2 src-prot local \
  address-start 1.0.0.0 address-end 1.255.255.255 med 50
```

BGP MULTIHOMED NETWORK WITH COMMUNITIES AND AS PREPEND

Figure 22 shows an access network supported by BGP. In this scenario, the two AP 45 platforms are IBGP peers, capable of exchanging routing information.

- AS 100 is advertising some routes with the community attribute.
- AS 200 is using AS prepend to advertise to Router B that this link is not preferred for some destinations in 1.0.0.0.

Figure 22 Complex BGP Multihomed Network



The commands given here illustrate IBGP peering, use of the community attribute usage to accomplish receive filtering, and use of the AS prepend attribute.

```
!Configuration for Access Point 45-1
  > config ip bgp local-as 200
  > add ip bgp peer.1.1.1.2
  > add ip bgp peer.3.1.1.2
```

USING THE COMMUNITY ATTRIBUTE

The command shown here specifies that routes from community 100:300 will be preferred (local preference set to 150) to routes with a default local preference (100).

```
> add ip policy-receive.bgp name.<name> peer.3.1.1.2 community 100:300 \
  preference 150
```

SETTING UP IBGP PEERS

Explicit identification of IBGP peers is unnecessary. Each BGP peer in AS 200 has been identified separately as belonging to this AS, implicitly identifying them as IBGP peers.

```
!Configuration for Access Point 45-2
  > config ip bgp local-as 200
  > add ip bgp peer.1.1.1.1
  > add ip bgp peer.4.1.1.2
```

USING AS PREPEND

In this example, use of this attribute adds 5 AS numbers (hops) to the AS path list between AP 45-2 and Router B (BGP peer 4.1.1.2) for addresses on network 1.0.0.0.

```
> add ip policy-send.bgp name.<name> peer.4.1.1.2 src-prot local \
  address-start 1.1.0.0 address-end 1.1.255.255 as-prepend 5
```

CREATING AND SELECTING A FORWARDING POLICY

You can use CBQ to select a predefined forwarding profile. A forwarding profile specifies a policy that selectively lets you forward packets. This is also known as policy forwarding. It works like this. First, create a forwarding profile and associate it with the actions you want to take. Then using CBQ, you can classify incoming packets and match them with the predefined forwarding actions.

In policy forwarding, each classified packet is matched against the list of actions. When the packet encounters a valid action, such as drop or forward, the packet is dealt with according to the specified action. Once action is taken on a packet, the packet is no longer processed by the other actions on the list.

A forwarding profile can contain several different policies. The policy in a profile with the highest priority (smallest number in the action-preference parameter) is used first. If the policy is not applicable at the moment the packet is being forwarded (that is, a target interface is down), then the next highest priority policy is examined, and if necessary, the next one, until there is a valid policy. If all the policies in a profile are all invalid, the forwarding by destination (regular forwarding) is used.

DEFINING FORWARDING ACTIONS

You can define as many actions in a policy as you want. A list of possible actions against which classified packets can be matched and a sample command follow.

- Dump packets

  The IP headers of the packets are dumped into the log.

  ```
  > add ip forward-profile.profile1 action-preference.1 action dumpPacket
  ```

- Forward by destination

  Forwarding equivalent of when there is no policy routing.

  ```
  > add ip forward-profile.profile1 action-preference.1 action \
    forwardByDestination
  ```

- Drop packets

  Packets are dropped.

```
> add ip forward-profile.profile1 action-preference.1 action drop
```

- Forward packets to next hop

  Packets are forwarded to a next hop IP address that must be reachable through a local interface.

  ```
  > add ip forward-profile.profile1 action-preference.1 action \
    forwardToNextHop ip-addr 6.6.6.4
  ```

- Forward packets to next address

  Forwarding can use local or remote addresses. If the packet destination address is remote, the route table is consulted for the actual next hop. For a local address, packets are forwarded using the local interface.

  ```
  > add ip forward-profile.profile1 action-preference.1 action \
    forwardToAddress ip-addr 5.5.5.4
  ```

- Forward packets out interface

  Configure a list of interfaces. The interface list can include either an IP address or an IP instance. The list can also include virtual interfaces, such as those used by IPSec. The first active interface in the list is used for forwarding the packets.

  ```
  > add ip policy-interface.interface1 priority.1 interface 3.3.3.3
  ```

  ```
  > add ip forward-profile.profile1 action-preference.1 action \
    forwardOutInterface interfacelist interface1
  ```

- Forward by default interface list

  Configure a default list of interfaces. If a packet can’t reach its destination by means of regular forwarding, the default interface list is consulted. The packet is routed to the first active interface in the list of specified default interfaces or IP addresses.

  ```
  > add ip policy-interface.defaultinterface priority.1 interface 3.3.3.3
  ```

  ```
  > add ip forward-profile.profile1 action-preference.1 action \
    useDefaultInterfaceList interfacelist defaultinterface
  ```

The sections that follow explain how to configure a forwarding policy.

CREATING A FORWARDING PROFILE

To create a forwarding policy, use IP to specify a *forward-profile.* Think of the *forward-profile* as a container of prioritized policies that will be associated with a CBQ class. A typical series of commands follows. The first command adds an interface to a list of interfaces and gives it a priority.

```
> add ip policy-interface.<list of interfaces> priority.<instance> interface \
  <interface>
```

The following table describes the parameters used:

Parameter	Description
policy-interface	User-defined name for the interface list. It is a unique identifier for the policy interface list.
priority	Numerical priority of the interface.
interface	The actual interface.

The second command creates a forward profile that specifies an action and priority and also associates the list of interfaces from the first command with the forward profile.

```
> add ip forward-profile.<profile name> action-preference.<instance> action
  <forwarding action> interfaceList <list of interfaces>
```

The following table describes the parameters used:

Parameter	Description
forward-profile	User-defined name for the forwarding policy. It is a unique identifier for the policy that is associated with a CBQ class.
action-preference	Numerical priority of the action.
action	Forwarding action to take, such as dropping packets or forwarding packets to the next hop.
interfacelist	The list of IP interfaces.

The following example illustrates how to configure a profile called forwardout. The first two commands in the example below define and prioritize two interfaces in an interface list called interface1. Next, add forwardout to IP as

the forward-profile, and configure interfaceList for forwardOutInterface and specify interface1, associating the list of interfaces from the previous commands with the forward profile. Forwardout has a primary action that forwards the packet out interface ip.2 that was defined in the interfaceList and a secondary action that forwards the packet out interface 3.3.3.3. Finally, if neither of these interfaces is available, the subsequent command drops the packet.

```
> add ip policy-interface.interface1 priority.1 interface ip.2

> add ip policy-interface.interface1 priority.2 interface 3.3.3.3

> add ip forward-profile.forwardout action-preference.1 action \
  forwardOutInterface interfaceList interface1

> add ip forward-profile.forwardout action-preference.2 action drop
```

You can display information about the profile, forwardout, with the following command:

```
> show ip forward-profile.forwardout
```

You can display the actions being used with the following command:

```
> show ip forward-profiles action summary
```

DEFINING A CBQ CLASS

Define a CBQ class to use the policy in the forwarding profile. (For information about CBQ, see Chapter 7.) For policy forwarding, you should classify CBQ on input. The following command configures the class sales to use the profile forwardout. All the packets that are classified by this class are forwarded according to the policy defined in forwardout.

```
> add cbq.1 traffic-class.sales dest-ip-addresses 3.3.3.2 parent \
  root-input-tree forwarding-profile forwardout

> config cbq.1 traffic-class.root-input-tree row-status active
```

Here's another example:

You can use policy forwarding to redirect traffic. For example, users on the network 10.1.1.x generate HTTP traffic, but rather than having Web traffic handled there, you can use policy forwarding to redirect the traffic to a transparent external server (i.e., a Web proxy) that handles all HTTP functions. Assume the proxy IP address is 11.5.5.5.

Add a profile named web-redirect and define the forwarding profile to forward to the next hop address 11.5.5.5. Then using CBQ, define traffic class web-proxy for every inbound interface that needs redirection. CBQ classifies all HTTP traffic generated from 10.1.1.x. All classified traffic is forwarded to the proxy server according to profile web-redirect, bypassing the normal routing.

```
> add ip forward-profile.web-redirect action-preference.1 \
  action forwardToNextHop ip-addr 11.5.5.5

> config cbq.1 traffic-class.root-input-tree row-status active

> add cbq.1 traffic-class.webproxy src-ip-addresses 10.1.1.1- 10.1.1.255 \
  applications http forwarding-profile web-redirect
```

CONFIGURING VRRP

ABOUT VRRP

The purpose of Virtual Router Redundancy Protocol (VRRP) is to enable a group of routers to provide backup and failover capability for hosts configured with a default gateway. VRRP accomplishes this by enabling these routers to share a single interface. While only one of the routers (called the Owner) actually "owns" this interface, other routers can borrow it and begin forwarding traffic if the router using the interface for forwarding goes down. Thus, from a functional perspective, VRRP would be better named something like "Virtual Router Interface Protocol." It is not routing capabilities that VRRP makes redundant, but the use of a single IP address.

This implementation of VRRP provides redundancy for hosts configured with a default gateway configuration. VRRP offers fast failover (mean time is 3 seconds), and complies with RFC 2338.

HOW IT WORKS

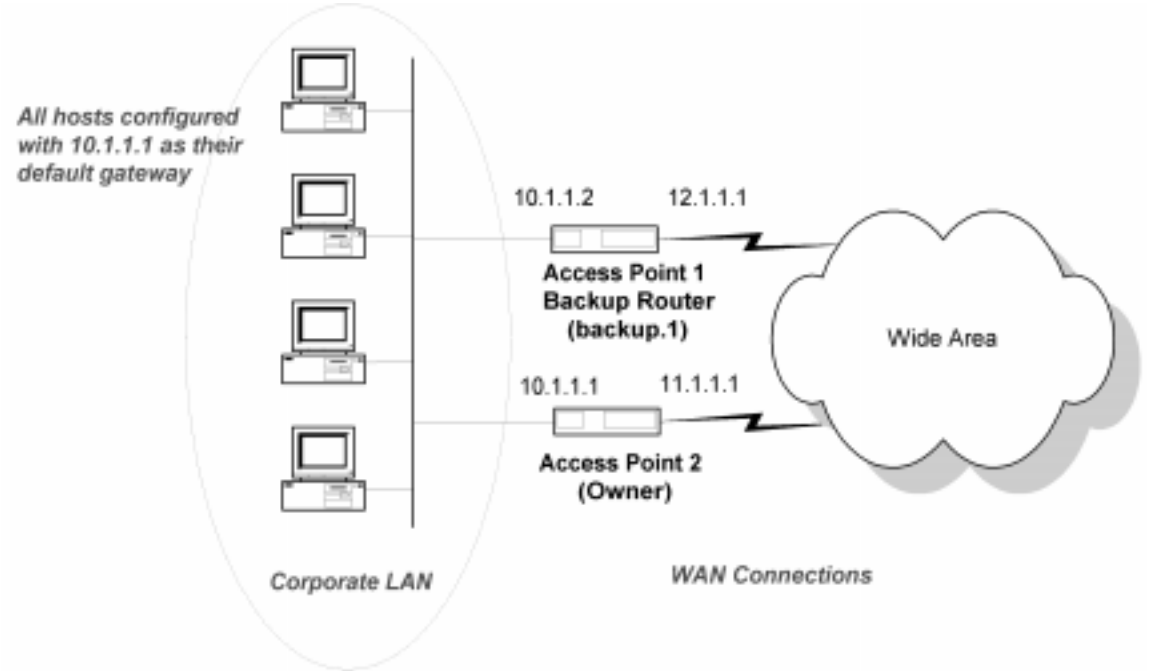
In a VRRP configuration, a group of routers act as a single "virtual" router for a set of hosts. Of this group, one router is designated as the master. This router is responsible for forwarding packets at any given time. Other routers in the configuration are backups, which monitor the health of the master. In essence, VRRP provides a means of promoting one of the listening backup routers to master if the current master goes down.

VRRP master routers use the hello protocol to let backup routers know that it

is functioning. When these messages stop arriving, one of the backup systems takes the place of the failed master.

Figure 23 shows a simplified VRRP configuration with a default gateway and a backup default gateway.

Figure 23 VRRP Configuration Showing Default and Backup Gateways



KEY TERMS

Setting up VRRP is straightforward. To understand what's going on, it helps to use a couple of standard designations to make the role of each router clear.

OWNER

The Owner is the router that is configured with the IP address that serves as the default gateway for connected hosts. In any VRRP setup, only one router can be the actual owner of the interface. A non-owner router can instantiate this address; this is what happens when it becomes master.

MASTER

A master router is the one that acts as the default gateway for a group of connected hosts. The master is often, but not always, the actual owner of the IP address.

From an operational standpoint, the master answers all ARP requests for its IP address and listens for the VRRP MAC address.

BACKUP

Backup routers listen to the hellos transmitted by an existing master. When hellos cease arriving, one of them becomes the master. Which router becomes master is determined by the priority setting assigned to each backup. Each router has the same default priority setting, but you can adjust the priority setting of any or all backup routers to create an order of succession among them.

CONFIGURING VRRP

You configure VRRP at the IP level with this command:

```
> config ip backup mode enable|disable
```

The steps that follow show how to configure redundancy between the routers shown in Figure 23. It is important that you perform each of these steps on each of the routers involved in your VRRP configuration.

1. Backup mode is disabled by default. To enable backup mode, use this command:

   ```
   > config ip backup mode enabled
   ```

2. Add backup routers as shown here.

   ```
   > add ip.1 backup.1
   ```

   To add a backup router, choose an arbitrary instance number. No more than four backup router instances are allowed on a single LAN segment.

3. Add the associated backup address to each backup instance.

   ```
   > add ip.1 backup.1 address.10.1.1.1
   ```

ESTABLISHING BACKUP PRIORITY

You can establish an order of succession, or priority, among backup routers by using this command:

```
> add ip.1 backup.1 priority <number from 1 - 254>
```

By default, each backup instance has a priority of 100. If the master router fails, and no router has been designated with a higher priority, any of the back-ups may become master. Otherwise, the backup instance with the highest priority number becomes master.

The numbers 0 and 255 are used for other purposes by VRRP. See Table 17 for details.

Table 17 VRRP and VBRP Options

Option	Purpose and Use
mode	Enables or disables standby processing on backup instances.
priority	Used by the VRRP in electing a new master router when the current master fails. The range 1 - 254 is available. 100 is the default value. 255 is reserved for the router that owns the IP address. 0 is reserved for the master router to indicate that it is relinquishing responsibility for the virtual router.
authentication-key	Use of authentication-key is not required, and no default value is provided. This option specifies a plain text 8-character string, used as a password to indicate either no authentication or reliance on some other mechanism, such as use of the IP authentication header.
hello-interval	Specifies the time interval between hello advertisements in seconds. Default is 1 second. Corresponds to advertisement-interval in the VRRP specification.
protocol	Specifies the protocol being controlled by VRRP. IP and bridge are possible values. IP is the default.

TRACKING AN IP INTERFACE

You can configure a VRRP instance to *track* an IP address. In this context, tracking is a mechanism used to have a VRRP instance state follow the state of another IP address. When a tracked IP interface is down, the VRRP instance will not become master. This can be useful in certain circumstances. For instance, if a default route on a WAN link provides the only access out of a site, it is not useful for VRRP to become the master if that link is down.

It works like this. When using the *tracked-ip* parameter, the associated IP address of the backup instance that is tracking another interface must be an IP address that is not owned by any of the VRRP instances on the LAN, and in a sense, it must not really exist on any interface on the network.

The following example uses the illustration in Figure 23. If backup.1 is on Access Point 1 (interface with 10.1.1.2) and it is tracking 12.1.1.1, the associated IP address for backup.1 must be configured with a non-existent IP address such as 10.1.1.3. Use the following command:

```
> add ip.1 backup.1 address.10.1.1.3
```

```
> config ip.1 backup.1 tracked-ip 12.1.1.1
```

Do the same for Access Point 2:

```
> add ip.1 backup.1 address.10.1.1.3
```

```
> config ip.1 backup.1 tracked-ip 11.1.1.1
```

Initially, both Access Point 1 and Access Point 2 come up in the backup mode. One of the APs becomes the master when it fails to hear from the other because there is no owner VRRP participant on address 10.1.1.3. From that point on, VRRP works as described in the previous sections. That is, one router stays in the backup mode while the first to become the master sends hellos. If that VRRP goes down because the tracked-ip address goes down, the other AP becomes master. This new master then sends out an ARP for the non-existent IP address 10.1.1.3 and traffic continues to flow.

NOTE *This means that the hosts on the corporate LAN need to configure 10.1.1.3 as their default gateway.*

6

CONFIGURING BRIDGING

The Access Point system (AP) can function as a LAN or WAN bridging device or a full-fledged Internet router. At boot time, the AP defaults to bridge mode if no WAN interfaces are detected. In this case, the AP functions as a transparent bridge between two (or more) Ethernet LAN segments.

The AP can also function as a bridging router. In this configuration, the AP routes all IP traffic, and bridges other LAN-based protocols (such as NetBEUI, NetBIOS, or IPX).

The sections that follow describe how to set up LAN and WAN bridging on the AP.

Converting a Router to a Bridge 260
Creating a Bridge-Router Configuration 261
Converting a Bridge to a Router 262
Configuring WAN Bridging 263
Routing Over Bridged Encapsulation 266
Configuring Bridge Redundancy 267

OVERVIEW

The AP supports the SNMP Bridge MIB - RFC 1493. Up to 10,000 bridge forwarding entries are supported. All Address Resolution Protocol (ARP) activity for bridged interfaces is conducted from the bridge layer. Commands used to configure and monitor ARP are located under the data link layer.

When operating in standard bridge mode, all of the AP's layers uplink to a single instance of IP. Standard learning and aging functions are included and are always enabled. A single MAC address is used for AP originated traffic. The MAC address used is the lowest lexical MAC address of any of the bridged interfaces. Static filters and Spanning Tree are not supported.

CONVERTING A ROUTER TO A BRIDGE

To convert a router to a bridge, use the *stack* command to remove each IP instance, then add a bridge instance. The general form of these commands are:

```
> stack slot<number> cbq.<instance> eth.<instance> no-ip
```

```
> stack slot<number> cbq.<instance> eth.<instance> bridge.new
```

The first command deletes an uplinked IP instance from a given slot; the second adds a new bridge instance to the same slot. Any other specific instance number, such as bridge.1 works just as well.

After adding new bridge instances, add an IP instance to any one of them as follows:

```
> stack slot<number> cbq.<instance> eth.<instance> bridge.<instance>
  ip.<new>
```

Together, the bridge instances function as a single bridge layer. All of the instances uplink to the new IP instance you have added. A status screen like that in Figure 24 shows the result of these changes. Note that each bridge instance uplinks to ip.1.

Figure 24 Status Screen Following Router to Bridge Conversion

```
> status

                                SYSTEM STATUS

 ROW|                                                          Layer Status|
----+----------------------------------------------------------------------+
   1|          slot.3.1 Ethernet Driver                  enabled      up   |
   2|          slot.3.1   eth.3                          enabled      up   |
   3|          slot.3.1    bridge.5                      enabled      up   |
   4|          slot.3.1     ip.1                         enabled      up   |
   5|          slot.4.1 Ethernet Driver                  enabled      up   |
   6|          slot.4.1   cbq.5                          enabled      up   |
   7|          slot.4.1    eth.4                         enabled      up   |
   8|          slot.4.1     bridge.4                     enabled      up   |
   9|          slot.4.1      ip.1                        enabled      up   |
  10|           virtual  Virtual Tunnel Transport        enabled    down   |
  11|           virtual   cbq.3                          enabled    down   |
  12|           virtual    ipsec.1                       enabled    down   |
  13|           virtual     ip.2                         enabled    down   |
```

In addition to the IP instance used by the bridging layer (ip.1), note that another IP instance (ip.2) lives in the Virtual Tunnel Transport Stack. Changes to the AP routing or bridging configuration do not affect this stack.

CREATING A BRIDGE-ROUTER CONFIGURATION

You can configure the AP to bridge all non-IP traffic and route IP traffic. To create a bridge-router configuration, simply use *stack* to insert a bridge instance above Ethernet.

```
> stack slot.<number> cbq.<instance> eth.<instance> bridge.new
```

After adding a bridge layer, issue a *status* command. The slot containing the bridge-router configuration you have added should look like the one shown in Figure 25. Note that the IP instance above the Ethernet layer remains in place. This instance continue to forward IP traffic; all other traffic is bridged.

Figure 25 Bridging Router Configuration

```
  10|          slot.4.1 Ethernet Driver                  enabled      up   |
  11|          slot.4.1   cbq.3                          enabled      up   |
  12|          slot.4.1    eth.2                         enabled      up   |
  13|          slot.4.1     ip.3                         enabled      up   |
  14|          slot.4.1     bridge.1                     enabled      up   |
  15|          slot.4.1      ip.10                       enabled    down   |
```

Note that adding a bridge instance to the stack adds an additional (and unnecessary) IP instance to the stack; in this case, ip.10 has been added. Because this IP instance is in a down state, it is not functional. It can be removed without affecting other aspects of the AP configuration.

One thing this additional IP instance provides is failover capability. In this sample, removal of ip.3 would cause the bridging setup to fail. In this case, you could assign an IP address to ip.10 to bring up the bridge. Under no circumstances should these IP addresses be in an up state simultaneously.

CONVERTING A BRIDGE TO A ROUTER

You can also use *stack* to convert a bridge to routing mode. First, remove the bridge instance, then add an IP instance to each AP slot. The general form of the commands that accomplish these goals are:

```
> stack slot<number> cbq.<instance> eth.<instance> no-bridge
```

```
> stack slot<number> cbq.<instance> eth.<instance> ip.new
```

The second command creates a new IP instance. Any other specific IP instance number, such as ip.1 works just as well as ip.new. The only caveat is that you must choose an IP instance that the AP is not using for other purposes.

After adding IP instances to each AP slot, assign an IP address and mask to each of them. To do this, use the *config* command as follows:

```
> config ip.<instance> address.<IP address> net-mask.<network mask>
```

CONFIGURING WAN BRIDGING

You can use the *stack* command to insert bridge interfaces over Ethernet, Frame Relay, ATM and PPP. For each of these media types, bridge forwarding operates on source and destination Ethernet MAC addresses.

On non-Ethernet interfaces, the AP simply encapsulates the Ethernet frames into a media-specific header described in the relevant RFCs. The sections that follow explain how to configure bridging for ATM, Frame Relay and PPP.

BRIDGING OVER FRAME RELAY

The AP supports bridging over Frame Relay connections, as described in RFC 1490. To configure bridging over Frame Relay, use the *stack* command to create an uplinked bridge instance on the desired Frame Relay interface. You can create the bridge on either the integral Frame Relay interface, or a Frame Relay subinterface.

This command creates a bridge instance above the integral Frame Relay interface:

```
> stack slot.3.1 cbq.1 hssi.1 frame-relay.2 bridge.new
```

This sample status screen shows the results of this command:

SYSTEM STACK

ROW		Layer	Status
1	slot.1.1 Ethernet Driver	enabled	up
2	slot.1.1 cbq.1	enabled	up
3	slot.1.1 eth.1	enabled	up
4	slot.1.1 ip.1	enabled	up
5	slot.3.1 LMC5200 HSSI Driver	enabled	up
6	slot.3.1 cbq.2	enabled	up
7	slot.3.1 hssi.1	enabled	up
8	slot.3.1 frame-relay.2	enabled	testing
9	slot.3.1 ip.2	enabled	down
10	slot.3.1 bridge.1	enabled	down
11	slot.3.1 ip.16	enabled	down

This command creates a bridge instance above a FR subinterface:

```
> stack slot.3.1 cbq.2 hssi.1 frame-relay.2 frame-relay.2.1 bridge.1
```

This sample status screen shows the results of this command:

SYSTEM STACK

ROW		Layer	Status
1	slot.1.1 Ethernet Driver	enabled	up
2	slot.1.1 cbq.1	enabled	up
3	slot.1.1 eth.1	enabled	up
4	slot.1.1 ip.1	enabled	up
5	slot.3.1 LMC5200 HSSI Driver	enabled	up
6	slot.3.1 cbq.2	enabled	up
7	slot.3.1 hssi.1	enabled	up
8	slot.3.1 frame-relay.2	enabled	testing
9	slot.3.1 ip.2	enabled	down
10	slot.3.1 frame-relay.2.1	enabled	down
11	slot.3.1 bridge.1	enabled	down
12	slot.3.1 ip.16	enabled	down

BRIDGING OVER PPP

To add a bridge layer above PPP, use the *stack* command as follows.

```
> stack slot.3.1 cbq.2 hssi.1 ppp.1 bridge.new
```

A sample stack with the bridge instance added is shown below. Note that the IP forwarder stacked above the bridge (ip.14) is not forwarding traffic in this example. All IP traffic on the PPP link is forwarded via ip.2. All non-IP traffic is bridged, via bridge.1.

SYSTEM STACK

ROW		Layer	Status
1	slot.1.1 Ethernet Driver	enabled	up
2	slot.1.1 cbq.1	enabled	up
3	slot.1.1 eth.1	enabled	up
4	slot.1.1 ip.1	enabled	up
5	slot.3.1 LMC5200 HSSI Driver	enabled	up
6	slot.3.1 cbq.2	enabled	up
7	slot.3.1 hssi.1	enabled	up
8	slot.3.1 ppp.1	enabled	testing
9	slot.3.1 ip.2	enabled	down
10	slot.3.1 bridge.1	enabled	down
11	slot.3.1 ip.14	enabled	down

BRIDGING OVER ATM

You can add a bridge instance to the integral ATM interface or an ATM sub-interface. To add a bridge instance the integral interface, use a *stack* command as in the following example:

```
> stack slot.1.1 cbq.1 atm-ds3.1 atm.1 ipoa.1 bridge.1
```

A sample ATM stack with a bridge added to the integral interface is shown here. The ipoa layer handles forwarding for IP traffic (by means of its IP forwarder) and non-IP traffic (using a bridging layer).

SYSTEM STACK

ROW		Layer	Status
1	slot.1.1 ATM Driver	enabled	up
2	slot.1.1 cbq.1	enabled	up
3	slot.1.1 atm-ds3.1	enabled	up
4	slot.1.1 atm.1	enabled	up
5	slot.1.1 ipoa.1	enabled	up
6	slot.1.1 ip.1	enabled	up
7	slot.1.1 bridge.1	enabled	up
8	slot.1.1 ip.8	enabled	down

To add bridge instances to two ATM subinterfaces, use a *stack* command like the following:

```
> stack slot.1.1 cbq.1 atm-ds3.1 atm.1 atm.1.1 ipoa.2 bridge.2
```

```
> stack slot.1.1 cbq.1 atm-ds3.1 atm.1 atm.1.2 ipoa.3 bridge.3
```

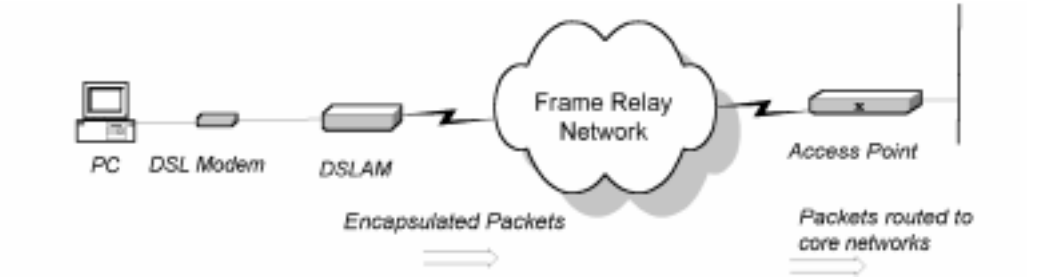
Following is a sample status screen with a bridge instance stacked above the integral ATM interface and two subinterfaces.

ROW		Layer	Status
1	slot.1.1 ATM Driver	enabled	up
2	slot.1.1 cbq.1	enabled	up
3	slot.1.1 atm-ds3.1	enabled	up
4	slot.1.1 atm.1	enabled	up
5	slot.1.1 ipoa.1	enabled	up
6	slot.1.1 ip.1	enabled	up
7	slot.1.1 bridge.1	enabled	up
8	slot.1.1 ip.8	enabled	down
9	slot.1.1 atm.1.1	enabled	up
10	slot.1.1 ipoa.2	enabled	testing
11	slot.1.1 ip.9	enabled	down
12	slot.1.1 bridge.2	enabled	down
13	slot.1.1 ip.8	enabled	down
14	slot.1.1 atm.1.2	enabled	up
15	slot.1.1 ipoa.3	enabled	testing
16	slot.1.1 ip.10	enabled	down
17	slot.1.1 bridge.3	enabled	down
18	slot.1.1 ip.8	enabled	down

ROUTING OVER BRIDGED ENCAPSULATION

You can configure the AP to accept RFC 1490 traffic over a Frame Relay interface. An example of this is shown in Figure 26. Here, DSLAM applications are encapsulated within RFC 1490 bridged Ethernet frames.

Figure 26 Bridging Encapsulated Traffic



To support the configuration shown in Figure 26, add a bridge instance above a Frame Relay interface or subinterface, and an uplinked IP instance. A sample command is shown here:

```
> stack slot.3.1 cbq.1 hssi.1 frame-relay.1 frame-relay.1.1 bridge.new ip.new
```

The stack created on the AP should look like the one shown here:

SYSTEM STACK

ROW	Layer		Status	
1	slot.1.1	Ethernet Driver	enabled	up
2	slot.1.1	cbq.1	enabled	up
3	slot.1.1	eth.1	enabled	up
4	slot.1.1	ip.1	enabled	up
5	slot.3.1	LMC5200 HSSI Driver	enabled	up
6	slot.3.1	cbq.2	enabled	up
7	slot.3.1	hssi.1	enabled	up
8	slot.3.1	frame-relay.1	enabled	up
9	slot.3.1	frame-relay.1.1	enabled	up
10	slot.3.1	bridge.1	enabled	up
11	slot.3.1	ip.2	enabled	up

In this instance, the bridge instance resides above the Frame Relay subinterface, frame-relay.1.1. Alternatively, the bridge can reside over the integral Frame Relay interface, frame-relay.1.

A sample *stack* command used to add a bridge over the integral interface is as follows:

```
> stack slot.3.1 cbq.2 hssi.1 frame-relay.1 bridge.1 ip.new
```

Here is the stack produced with this command:

SYSTEM STACK

ROW		Layer	Status
1	slot.2.1 Ethernet Driver	enabled	up
2	slot.2.1 cbq.1	enabled	up
3	slot.2.1 eth.1	enabled	up
4	slot.2.1 ip.1	enabled	up
5	slot.3.1 LMC5200 HSSI Driver	enabled	up
6	slot.3.1 cbq.2	enabled	up
7	slot.3.1 hssi.1	enabled	up
8	slot.3.1 frame-relay.1	enabled	up
9	slot.3.1 ip.2	enabled	up
10	slot.3.1 bridge.1	enabled	up
11	slot.3.1 ip.8	enabled	down

CONFIGURING BRIDGE REDUNDANCY

ABOUT VIRTUAL BRIDGE REDUNDANCY PROTOCOL

The Virtual Bridge Redundancy Protocol (VBRP) uses VRRP mechanisms to provide backup capability in AP bridge configurations. VBRP is a proprietary implementation that provides fast-failover and loop-free forwarding between two LANs. The feature's use is currently limited to 2-port bridge configurations.

HOW IT WORKS

VBRP is a configurable option in which a forwarding bridge (called a master) sends "hello" messages on a regular basis to notify backup bridges that it is operational.

Backup bridges listen for these messages. When the established "hello" interval has elapsed and no message has been received, failure of the current master is assumed. A new master bridge is chosen from among the backups. Bridge forwarding continues, with a default failover time of 6 seconds.

Figure 27 shows a simplified bridge configuration with a master and two backup bridges.

KEY TERMS

Setting up VBRP is straightforward. To understand what's going on, it helps to use a couple of standard designations to make the role of each bridge clear.

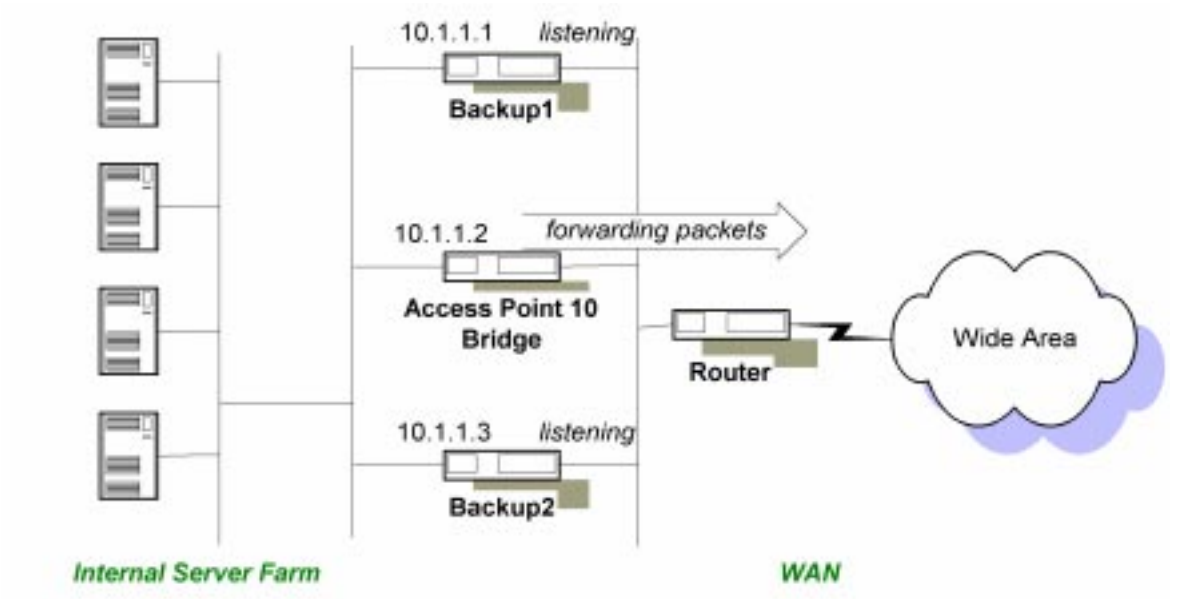
MASTER

A master bridge is the only one forwarding packets. To prevent forwarding loops, VBRP sees to it that there can only be one master at any time.

BACKUP

Backup bridges are bridges that are listening to an existing master. As long as the master continues to transmit hello messages, backup bridges do nothing but listen. When hellos cease arriving, one of the backup bridges becomes the master. Which bridge becomes master is determined by the priority setting assigned to each backup. The instructions that follow explain how to set up this backup framework.

Figure 27 Bridge Configuration Showing Master and Backup Bridges



In this example, the master bridge (bridge interface 10.1.1.2) forwards packets between the router and server farm. It is also sending multicast hello messages to listening backup bridges 1 and 2.

CONFIGURING VBRP

To configure VBRP, follow the steps outlined below. With the exception of step 1, in which you assign an IP address to each bridge, all other steps must be performed *exactly the same for each bridge*. This allows all backup bridges to act in concert in the event the master fails.

1. Configure IP on each bridge.

   If you have not configured an IP instance for general management and IP connectivity of each bridge, do this first.

   ```
   > config ip.1 address.10.1.1.1
   ```

   Again, the address configured here will differ for each bridge. In the case of the example shown in Figure 27, 10.1.1.1, 10.1.1.2, and 10.1.1.3 will have to be configured separately on each bridge.

2. Configure each bridge for backup.

   For the next step, configure the backup feature on each bridge as follows:

   ```
   > config bridge transparent-bridge backup-mode enabled
   ```

   Note that when you enable backup mode for the bridge, forwarding will stop until VBRP is completely configured and a master bridge has been selected.

3. Enable IP backup mode on each bridge.

   ```
   > config ip backup mode enabled
   ```

4. Add a bridge and backup number.

   This command adds an instance of the backup bridge.

   ```
   > add ip.1 backup.[1 - 255]
   ```

   The backup number provided here is an index number necessary for identifying a group of backup bridges. This number must be the same for each backup bridge.

5. Set protocol for each backup instance to bridging.

   ```
   > config ip.1 backup.[number] protocol bridge
   ```

6. Add an associated backup address to each backup instance.

   ```
   > add ip.1 backup.[number] address.[ip address]
   ```

You must assign one of the IP addresses assigned to the bridge group as the associated IP address. This bridge becomes the master bridge, and begins forwarding operations. In this case, the bridge chosen (see Figure 27) is 10.1.1.2. The other two bridges automatically become backups, and begin listening for hello messages.

In VRRP terms, this designated master is also the owner of the IP address.

7 Set a priority among backup bridges.

This is an optional step. If you do not designate backup priorities, each backup will attempt to become master in the event of a failure. One bridge will succeed; the others will go back to listening.

To designate an order, use this *config* command on each bridge:

```
> add ip.1 backup.[number] priority.[number]
```

The priority number provided must be from 1-254. By default, each bridge has a priority of 100. 0 and 255 are reserved for the master bridge; the next highest numbered backup router would have priority in the event the master fails. See Table 18 for details.

Table 18 VRRP and VBRP Options

Option	Purpose and Use
mode	Enables or disables standby processing on backup instances.
priority	Used by the VRRP in electing a new master router when the current master fails. The range 1 - 254 is available. 100 is the default value. 255 is reserved for the router that owns the IP address. 0 is reserved for the master router to indicate that it is relinquishing responsibility for the virtual router.
authentication-key	Use of authentication-key is not required, and no default value is provided. This option specifies a plain text 8-character string, used as a password to indicate either no authentication or reliance on some other mechanism, such as use of the IP authentication header.
hello-interval	Specifies the time interval between hello advertisements in seconds. Default is 1 second. Corresponds to advertisement-interval in the VRRP specification.
protocol	Specifies the protocol being controlled by VRRP. IP and bridge are possible values. IP is the default.

7

CONFIGURING CBQ FOR TRAFFIC MANAGEMENT

This chapter explains how to configure and use Class-Based Queuing (CBQ) to manage bandwidth utilization. It contains the following sections.

Traffic Management Overview 272
Configuring Traffic Classes 277
Basic CBQ Components 281
Using Traffic Parameters 283
Using Stateless and Stateful Classification 305
Managing CBQ 317
Using AutoClass 324

TRAFFIC MANAGEMENT OVERVIEW

In essence, CBQ is a link-sharing mechanism that allows administrators to *classify* and *schedule* IP traffic. CBQ classifies packets by matching them against a set of traffic classes. Each traffic class instantiates a classification policy for a defined type of traffic. For instance, an FTP packet arriving on an interface may match to a specific traffic class called *FTP-outbound-to-DMZ*.

Once the packet is classified in this way, CBQ schedules the packet with a specific amount of the bandwidth available on the link (for instance, 500Kbps). Subsequent packets that match the same class are guaranteed the same amount of bandwidth. Moreover, if all 500Kbps allocated to the class is in use, CBQ can be configured to allow the class to borrow unused bandwidth from other classes on the link.

These classes exist in a structure of parent-child relationships called a *CBQ tree*. Together, the classes you create on any AP interface determine *how* bandwidth is to be managed, and in what *direction* (either on input, or output).

CBQ AS A BANDWIDTH MANAGER

Traffic classes are a tool, providing you with fine-grained control of the way bandwidth is apportioned on any AP interface. Often referred to as *traffic shaping*, CBQ's bandwidth management capabilities include:

- Tight bandwidth provisioning, based on host addresses, transport protocol, or application, transport ports, domain names, and other factors.
- Stateful application classification, for ease-of-configuration, enhanced QoS, and greater visibility into traffic passing through the AP
- Controlled link borrowing or "bursting capabilities" for selected classes, based on excess bandwidth available on a link
- Automatic class creation with AutoClass
- Diff-Serv implementation, based on ToS byte classification

These and other bandwidth management capabilities are made possible by a hierarchical set of traffic classes (or *CBQ tree*) you configure on an AP interface. For a brief example showing how to configure traffic classes, see Page 278. See Page 283 for a description of CBQ parameters and their uses.

CBQ AS A FIREWALL

In addition shaping IP-based traffic, you can configure CBQ to function as a firewall. Chapter 8 explains how to configure CBQ to filter specific traffic classes, and how to protect the AP and internal networks from a number of well-known attacks including:

- ICMP fragment attacks
- SYN flooding

The discussion that follows provides a general overview of CBQ capabilities and functions. Some examples in this chapter make reference to the AP's firewalling capabilities. For detailed instruction on configuring CBQ as a firewall, see Chapter 8.

ABOUT TRAFFIC CLASSIFICATION

The idea behind CBQ is to provide a way to apportion bandwidth to the packets that pass through AP interfaces, providing guaranteed levels of service (bandwidth) to certain classes, and filtering other classes.

CBQ uses a few basic mechanisms to accomplish these goals:

- A traffic classification hierarchy, or *CBQ tree*, that organizes traffic classes in parent-child relationships
- Traffic classes that provide committed service levels (QoS) for certain types of traffic
- Traffic classes designed to filter (drop) certain types of traffic.

On each AP interface, you can configure CBQ to classify traffic on input (received), output (transmitted) traffic, or in both directions. This provides great flexibility in the management of QoS and security considerations on AP interfaces.

HOW CBQ TREES WORK

Each CBQ hierarchy, or "tree," imposes a hierarchical classification on IP traffic flowing in one direction through the interface. Often called a *CBQ Tree*, this hierarchy determines the shape, rate, and other flow characteristics of traffic transmitted in one direction on an interface.

Individually, each traffic class determines how much bandwidth to allocate (if any) for IP packets that match a specific set of characteristics. Each class accomplishes this by provisioning bandwidth to packets that match to a specific set of characteristics. Working together as a whole, the traffic classes in a CBQ tree determine how to allocate bandwidth on all packets flowing through the interface.

CLASS RELATIONSHIPS IN A CBQ HIERARCHY

Figure 28 shows the relationships between a number of traffic classes in a simple CBQ tree. Classes Red, Blue and Green are children of the root class (either root-input-tree or root-output-tree). These classes share the total amount of bandwidth owned by root (by default, each root class has 100 Mbps). Each of these classes is a parent to two child classes, which share in its bandwidth. For example, the 40 Mbps owned by class Red is shared unequally by Red1 and Red2.

Figure 28 Sample CBQ Tree Showing Parent-Child Relationships

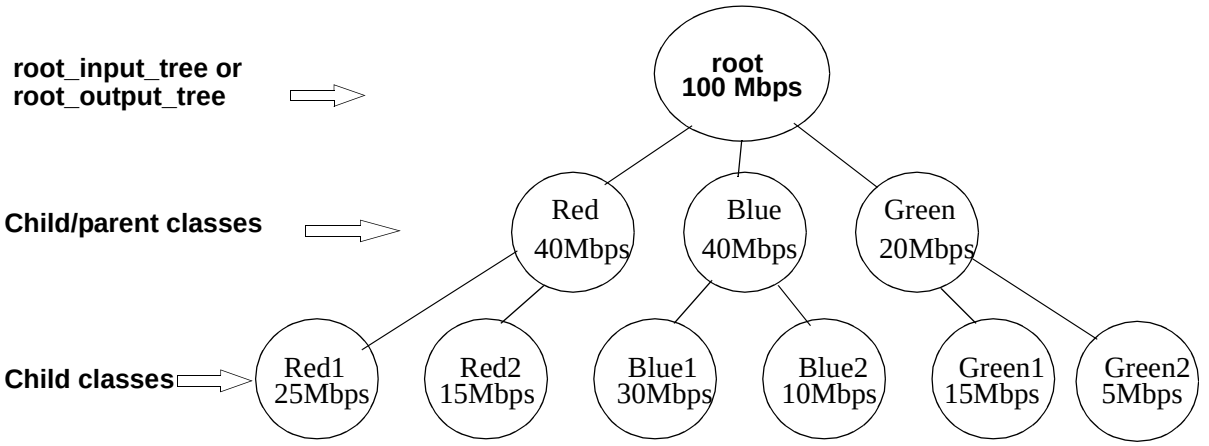
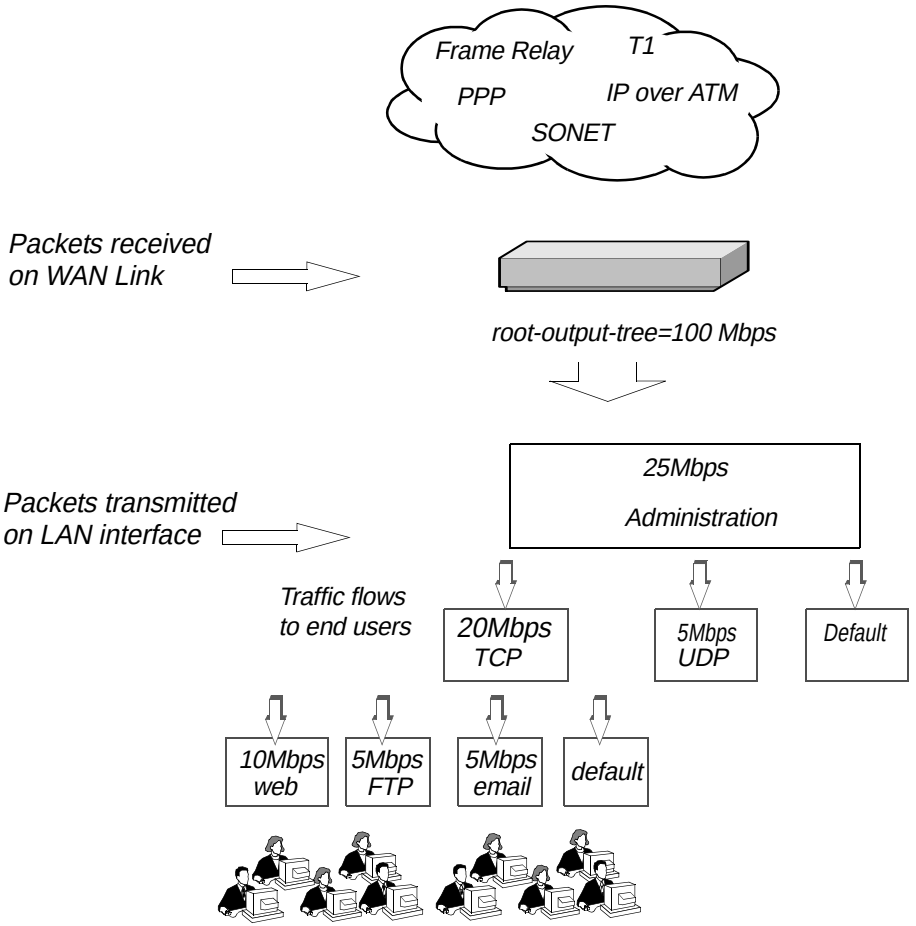


Figure 29 shows how traffic is classified as it is transmitted to a simple LAN. As traffic enters the root-output-tree, packets matching the Administration class are classified. Packets are further classified by protocol (TCP or UDP) and scheduled (assigned bandwidth) accordingly. Packets belonging to the TCP branch are also classified by application (port number).

Figure 29 Traffic Management, WAN-to-LAN



Notice that two default classes exist: one is a child of Administration, the other is a child of class TCP. Default classes match to any packet not otherwise matched. Thus, an ICMP packet classified by Administration would be further classified by the default class.

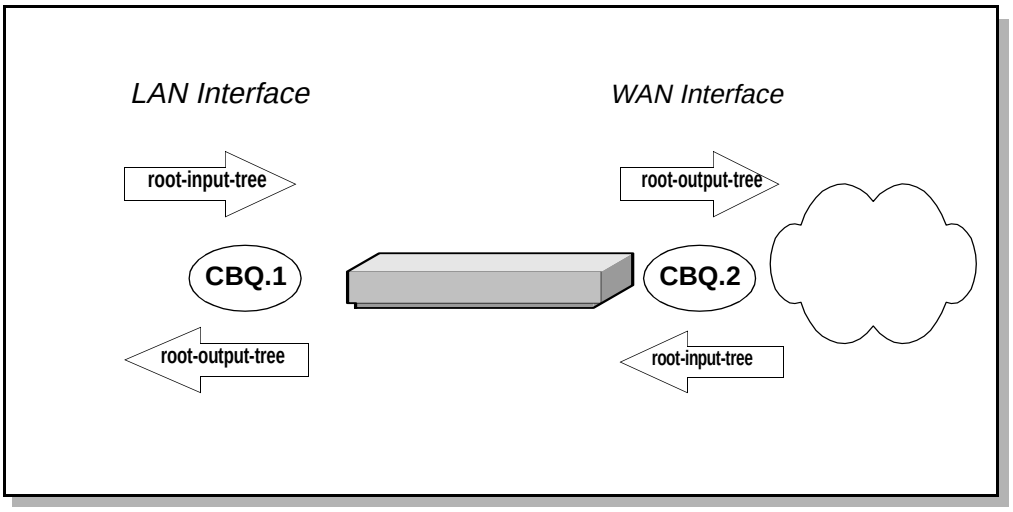
While this tree has only three levels, additional classification and scheduling could be performed based on the source or destination domain name of packets received, or using the 1 byte ToS field in each packet header.

For fully bidirectional bandwidth management, another CBQ tree could be added on the *root-output-tree* of the WAN interface. Classes on this tree could be used to manage bandwidth and firewall packets sent from the LAN to the WAN.

CLASSIFYING TRAFFIC ON INPUT AND OUTPUT

By default, each AP interface supports two independent CBQ trees: one tree classifies traffic on input (reception). The other classifies traffic on output (transmission). The word “can” is important in this context. You can configure CBQ to classify traffic in either, neither, or both directions. In the simple picture shown in Figure 30, Classes on the *root-input-tree* of CBQ.2 classify traffic *entering* the AP from the WAN. Conversely, classes on the root-output-tree of CBQ.2 classify traffic as it is *transmitted* to the WAN.

Figure 30 Root Input and Output Trees on CBQ Instances



The same directional rule applies to the CBQ trees on CBQ.1: the classes on the *root-input-tree* classify traffic on input, while those on the *root-output-tree* classify traffic on transmit (in this instance, before the traffic enters the LAN).

Whether to use CBQ on input or output traffic, or in both directions, depends on specific goals of your configuration. As a pure link-sharing mechanism, CBQ can be used to guarantee committed service levels. However, you can also use CBQ to implement a firewall. See Chapter 8 for information on using CBQ to filter unwanted traffic.

CONFIGURING TRAFFIC CLASSES

To configure a traffic class, use the *config* command followed by the CBQ instance you are configuring, and a set of parameters that specify a match policy for the class.

Each traffic class declaration has this form:

```
> config cbq.<instance> traffic-class.<classname> parent <parenting>
```

This command specifies the minimum parameters needed to define a class. It specifies a specific CBQ instance, a unique traffic-class name, and a parent. Most class definitions also specify the bandwidth allocation allotted to the class.

A SAMPLE CLASS DEFINITION

Here, for example, is a representative traffic class definition for a sample class called *tcp-inbound*.

```
> config cbq.1 traffic-class.tcp-inbound protocols tcp parent root-input-tree
  bandwidth-allocation 500000 bounded false max-bandwidth 600000 priority
  1
```

The definition for tcp-inbound incorporates the following information:

- The class is bound to cbq.1, and its parent is root-input-tree.
- The class matches all TCP-based packets.
- Total bandwidth allocated to the class is 0.5 Mbps.
- The class can use a maximum of 0.6 Mbps. See “Using the Max Bandwidth Parameter” for more information on the max-bandwidth parameter.
- The class’s borrowing priority is 1, which gives it preference among borrowing classes at its level.

Together, these characteristics comprise a *classification policy* for a single class in cbq.1. Note that the parameters used in this (or any) class definition are position-independent. When creating a traffic class, you can enter parameters in any order

A SIMPLE CONFIGURATION

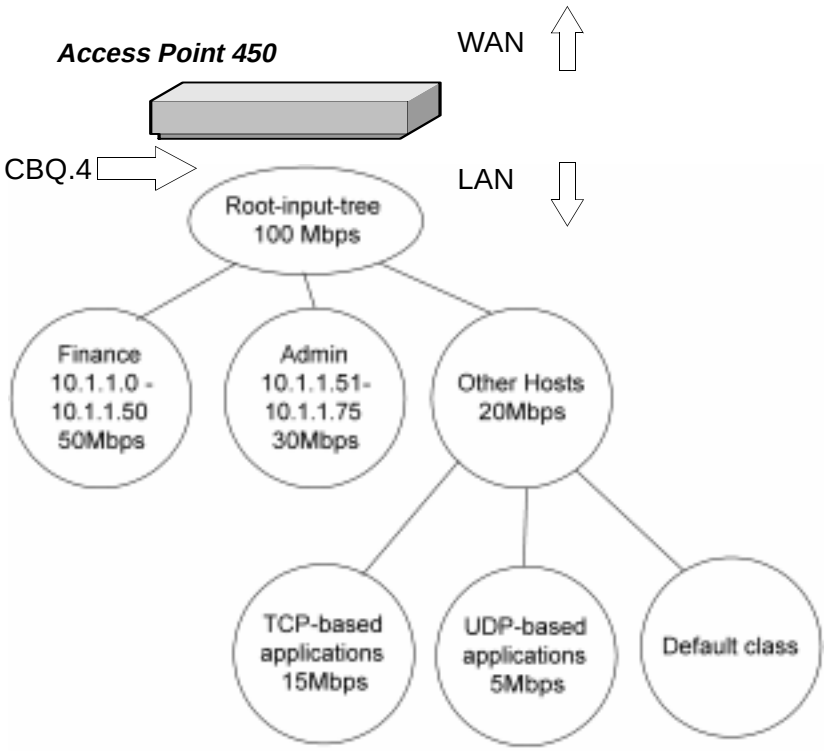
A simple example is perhaps the best illustration of what CBQ is, and how to use it. Let's say you want to manage the bandwidth available to users on in a particular organization. A total of 100Mbps exists on the LAN interface. Rather than allowing bandwidth to be used on a best-efforts basis, you want to share the bandwidth in specific ways, providing guaranteed levels of service to certain users, and best efforts service to others:

The specific goals of this sample configuration are as follows:

- Reserve 50Mbps for one functional group (finance)
- Reserve 30Mbps for another group (administration)
- Reserve the remaining 20Mbps for other hosts in the organization, and break this down as follows:
  - 15Mbps for these TCP-based applications: http, email, telnet, ftp
  - 5Mbps for these UDP-based applications: snmp, tftp
  - A default class that allocates bandwidth to other TCP and UDP-based services

In addition to these goals, you want to allow finance and admin to borrow additional, unused bandwidth from one another (or from the bandwidth reserved for class other-hosts) as needed. Implementing each of these criteria will require you to create a CBQ tree. You can represent the tree needed to implement these policies as shown in Figure 31:

Figure 31 Simple CBQ Tree



Notice that the tree shown in here apportions available link bandwidth in a set of classes bound together in a hierarchy (or CBQ tree). The bubbles represent traffic classes; connecting lines between them represent parent-child relationships. Parent classes control a certain amount of bandwidth, and dole this out to their children. Classes Finance, Admin, and Other-Hosts are children of root-input-tree; classes tcp-based-apps, udp-based-apps and Other-default are children of Other-hosts.

CBQ makes it easy to create the traffic classes needed to support the goals of this configuration. Let’s complete the configuration step-by-step.

TURN ON THE ROOT-INPUT-TREE

The first step (or the last) in any configuration is to turn on the root class at the top of the tree. In this case, the root class is the *root-input-tree* of CBQ.4.

```
> config cbq.4 traffic-class.root-input-tree row-status active
```

CREATE FINANCE AND ADMIN CLASSES

```
> config cbq.4 traffic-class.finance src-ip-addresses 10.1.1.0-10.1.1.50 band-
  width-allocation 50000000 parent root-input-tree

> config cbq.4 traffic-class.admin src-ip-addresses 10.1.1.51-10.1.1.75 band-
  width-allocation 30000000 parent root-input-tree
```

CREATE A NULL CLASS FOR OTHER-HOSTS

Class Other-hosts is designed to share bandwidth that has not been allocated to Finance or Admin.

```
> config cbq.4 traffic-class.other-hosts parent root-input-tree
```

Rather than sharing this bandwidth according to IP addresses (as in the case of Finance and Admin), the 20Mbps allocated to this class is shared according to protocol.

CREATE CHILD CLASSES OF OTHER-HOSTS

These child classes share the bandwidth available to other-hosts. The default class (called other-default) assures that any packets that do not match to either of these child classes will be transmitted.

```
> config cbq.4 traffic-class.tcp-based-apps bandwidth-allocation 15000000
  parent root-input-tree

> config cbq.4 traffic-class.udp-based-apps bandwidth-allocation 5000000
  parent root-input-tree

> config cbq.4 traffic-class.other-default parent other-hosts
```

In the absence of a default class, any packets that do not match a traffic class are dropped. See Page 282 for information on using default classes.

SHOW THE RESULTS OF THE CONFIGURATION

To show the results of the traffic classes created on CBQ.4, issue this command:

```
> show cbq.4 traffic-class-tree
```

CBQ.4 TRAFFIC-CLASS-TREE

INTERFACE	ROW	Class Hierarchy	
cbq.4	1	root-input-tree	Rate=1000000000
cbq.4	2	finance	Rate= 50000
cbq.4	3	admin	Rate= 30000
cbq.4	4	other-hosts	Rate= 20000
cbq.4	5	tcp-based-apps	Rate= 15000
cbq.4	6	udp-based-apps	Rate= 5000
cbq.4	7	other-default	Rate= 0

BASIC CBQ COMPONENTS

Traffic classes are the building blocks of CBQ. Each traffic classification hierarchy may consist of four different types of classes. These are:

- A Root class (root-input-tree or root-output-tree)
- Ordinary traffic classes
- AutoClasses
- Default classes
- Null classes

As the name implies, ordinary traffic classes are the kind used most often. However, default and Null classes also serve important functions.

ABOUT ROOT CLASSES

Each AP interface has a single CBQ instance. Each CBQ instance can have two CBQ trees associated with it: one tree for input, and one for output traffic. Each of these trees has a master or *root* class associated with it. The root class is literally the root of the CBQ tree that grows beneath it. By default, the root classes on each interface are named as follows:

- root-input-tree
- root-output-tree

By default, each root class (whether *root-input-tree* or *root-output-tree*) has 100 Mbps. The structure of traffic classes you configure on each interface lives

below a root class, as shown in Figure 28. The root class owns all bandwidth assigned to the CBQ hierarchy, up to the total amount available on the link.

While a root-input-tree and a root-output-tree exist by default on each CBQ instance, you must activate (that is, turn on) the tree either before or after creating traffic classes for the classes to become active. The section called "Turn On the root-input-tree" on Page 279 explains how to activate the root class.

ORDINARY TRAFFIC CLASSES

A traffic class is a mechanism used to specify a classification policy at a specific level in the CBQ tree. In this context, "ordinary" is used to distinguish traffic classes from AutoClasses. While ordinary classes are configured by the administrator, AutoClasses are automatically generated by packets passing through an Access Point interface. Traffic classes allocate a specific amount of bandwidth to a defined class of traffic. The definition of each class can incorporate one or more of several factors, including:

- Source/destination IP address of packets
- Transport protocol used
- Source or destination ports
- Domain name
- ToS byte value specified
- Applications

Classification by ToS byte value enables implementation of Diff-Serv. This is a specialized application of CBQ. For details on classifying by ToS values, see "Using the ToS Parameters to Implement Diff-Serv" on Page 335.

DEFAULT CLASSES

Like a wildcard definition, a default class applies to all types of traffic that are unclassified at their level in the hierarchy. Adding a default class guarantees that the AP will pass all traffic through to its destination — whether or not it can be matched to classes you have specifically defined. You can add default classes at any level of the traffic hierarchy. The default class syntax consists of a name and the suffix *-default*. For example:

```
> config cbq.1 traffic-class.email-default parent root-input-tree
```

NULL CLASSES

Null classes are useful in cases where you want to subdivide the bandwidth allocated to a specific business area. Unlike other classes, Null classes have no classification requirements, however bandwidth can be allocated to them.

AUTOCLASSES

AutoClasses provide you with a way to automatically classify and schedule traffic on AP interfaces. When AutoClass is turned on for a traffic class (auto-class-enabled=true), child classes are classified (created) and scheduled (assigned bandwidth) based on the packets that match the parameters configured for the AutoClass parent.

AutoClasses are classes created by packets as they pass through an AP interface. This is also a somewhat specialized area within CBQ. See "Using AutoClass" on Page 324 for detailed information on configuring AutoClasses.

USING TRAFFIC PARAMETERS

CBQ provides a rich set of traffic parameters for classifying traffic. These can be used singly or in combination. A few examples follow.

Each of the brief sections that follow explains how to use a specific classification parameter. For the purpose of clarity, the examples that illustrate each section use these parameters singly. However, you can use CBQ parameters in combination to create very specific classification policies for IP traffic. For detailed information, review each of these sections, then see "Using Traffic Parameters in Combination" on Page 300.

CLASSIFYING ON HOST OR HOST RANGE

To classify packets based on host range, use the following parameters.

- src-ip-addresses
- dest-ip-addresses

USAGE NOTE

The src-ip-addresses and dest-ip-addresses parameters are used to specify an IP address or address range. You can enter one of the following:

- An IP address
- Two IP addresses separated by a hyphen to specify an address range
- More than one IP address separated by commas to specify a set of non-contiguous addresses

EXAMPLE - DESTINATION IP ADDRESS RANGE

This example creates a traffic class called abc-net with a specific destination IP address range (presumably, that of an ISP called ABCnet).

```
> config cbq.1 traffic-class.abc-net dest-ip-addresses \
  198.202.232.0-198.202.232.255 parent root-input-tree \
  bandwidth-allocation 100000000
```

This traffic class will match transmitted packets with the destination IP address range specified. The class is allocated 100 Mbps; hence, traffic bound for hosts in this range can use this amount of bandwidth.

EXAMPLE - SOURCE IP ADDRESS RANGE

This example creates a traffic class called finance with a specific source IP address range (presumably that of a finance group).

```
> config cbq.1 traffic-class.finance src-ip-addresses 128.2.3.45-128.2.3.67 \
  parent departments bandwidth-allocation 500000 bounded true
```

This traffic class will match transmitted packets with the source IP address range specified. The class is allocated 500 Kbps (500000). Because this class is not allowed to borrow additional bandwidth (bounded true), this is the maximum bandwidth utilization for the class as a whole.

CLASSIFYING ON APPLICATIONS

You can use the applications parameter to classify a list of commonly used services (for example tftp, telnet, ftp, dns and http) by name, rather than by port number. Generally, this parameter is used to create a stateful pair of traffic classes. However, you can use this parameter when configuring well-known applications (see the list on Page 308). Doing so is the equivalent of classifying an application by source or destination port.

CONFIGURING A STATEFUL PAIR

The first member of a stateful pair is an ordinary traffic class, and handles initiated connections. An example is the class called http-out, defined as follows:

```
> config cbq.1 traffic-class.http-out applications http bandwidth-allocation
  50000 parent root-output-tree
```

The second member is an established traffic class, and handles data returning on the initiated flow. A return (established) class that works with http-out to create a stateful application flow might be configured as follows:

```
> config cbq.1 traffic-class.http-return applications httpEstalished bandwidth-
  allocation 50000 parent root-input-tree
```

Notice that these two classes are attached to the root-input-tree and root-output-tree of CBQ.1. While each member of a stateful pair can be attached to different CBQ instance, the flows must be in opposite direction. That is, an initiated flow from LAN to WAN must be matched by a complementary flow from WAN to LAN. Table 19 (Page 308) lists the applications that are explicitly supported by the AP.

USING THE APPLICATIONS PARAMETER FOR STATELESS CLASSIFICATION

You can use the applications parameter for ordinary (stateless) classification of traffic. To do this, create a class definition (like http-out, shown above) without a complementary established class. Doing this is the equivalent of classifying an application by source or destination port.

MORE ON STATEFUL TRAFFIC CLASS PAIRS

This section offers a brief description of the applications parameter. If you intend to use CBQ as a firewall, or need to configure stateful classes for enhanced QoS, refer to Page 308 for complete information on configuring and using stateful applications.

CLASSIFYING ON PROTOCOL OR PROTOCOL RANGE

To classify packets based on protocol or protocol range, use the protocols parameter.

USAGE NOTE

The protocols parameter is used to specify the following values:

- The word “any” or a text string for one of the following protocols: icmp, tcp, udp, ospf
- An integer specifying a standard protocol number

Two values separated by a hyphen specify a protocol range. You can enter multiple values separated by a comma.

EXAMPLE- PROTOCOL RANGE

This example creates a traffic class called TCP-out that matches all TCP-based packets. Note that the title, TCP-out, implies that the class is implemented on an outbound interface.

```
> config cbq.1 traffic-class.TCP-out protocols tcp\
  bandwidth-allocation 20000000 parent root-output-tree
```

To classify according to a range of protocols, you can use protocol numbers, as shown in this example. The protocol number for TCP is 6; the number for UDP is 17.

```
> config cbq.1 traffic-class.TCP-UDP protocols 6-17 \
  bandwidth-allocation 40000000 parent root-output-tree
```

Because this classification specifies a range, note that the definition matches not just protocols 6 and 17, but protocols 7 - 16 as well.

CLASSIFYING ON PORT OR PORT RANGE

To classify packets based on port or port range, use the following parameters.

- src-ports
- dest-ports

USAGE NOTE

The src-ports and dest-ports parameters are used to specify these values:

- The word “any” or a text string for one of the following services: ftpdata, ftp, telnet, smtp, domain, bootps, bootpc, tftp, gopher, finger, http, nntp, ntp, snmp, snmptrap, bgp
- An integer specifying a standard port number
- Two integers separated by a hyphen to specify a port range

You can enter multiple values separated by a comma.

Text strings and integers can be used in conjunction, provided a lower value text string is used for the start value. For example, if using one of the supported text strings (say, telnet, which maps to port 25), a higher port number (that is, greater than 25) can be supplied for the end value.

EXAMPLE- SOURCE PORT RANGE

This example creates a traffic class called all-snmp with a specific source port definition. The idea is to match all SNMP packets, and assign a fixed amount of bandwidth to them.

```
> config cbq.1 traffic-class.all-snmp src-ports 161 bandwidth-allocation
  500000 parent services
```

EXAMPLE - DESTINATION PORT RANGE

This example creates a traffic class called tcp-services. The definition uses a range of destination ports that encompasses three commonly-used services: FTP (port 21), email (port 25), and HTTP (port 80).

```
> config cbq.1 traffic-class.tcp-services dest-ports 21-80 \
  bandwidth-allocation 1000000 parent root-output-tree
```

Because the range of ports specified encompasses many more port numbers (and hence, service types) than the three mentioned, this definition may be too

coarse to be useful. When defining traffic classes for specific ports, it is often best to configure a Null class, and make this the parent class for a number of individual services. For a discussion of Null classes and their uses, see Page 283.

USING AN OPERATOR

When defining traffic parameters in combination, you can specify that packets must match *all* or *any* of the parameters before the packet can be classified within the defined class by using the operator parameter. The and operator indicates that all traffic parameters listed must match for a packet to belong to a class. This is the default operator for CBQ classes. The or parameter indicates that any one of the parameters listed is sufficient to classify a packet as belonging to a class.

For the following examples, 198.2.3.15 and UDP are specified as the source IP address and protocol for your classification parameters.

To specify that both these parameters must match before a packet can be classified within the defined class, use this command:

```
> config cbq.1 traffic-class.finance-web src-ip-addresses 198.2.3.15 \
  protocols udp operator and
```

To specify that either of these parameters must match before a packet can be classified within the defined class, use this command:

```
> config cbq.1 traffic-class.finance-web src-ip-addresses 198.2.3.15 \
  protocols udp operator or
```

USING DOMAIN NAME CLASSIFICATION

Classification of traffic by source and destination domain name is supported by two CBQ parameters:

- src-domain-names
- dest-domain-names

To classify traffic by either source or destination domain name, provide a qualified domain name string along with the appropriate parameter as shown here:

```
> config cbq.1 traffic-class.abcplastics src-domain-names\
  somehost.abcplastics.com parent root-output-tree \
  bandwidth-allocation 50000
```

Packets belonging to the specified domain will be resolved by the AP's DNS client and matched to the class definition.

USING URL NAMES

Use of URL names is a problematic area, chiefly owing to fact that many companies routinely alias whole groups of machines to the same URL. For this reason, classifying traffic based on URL names may not produce reliable results. This is particularly true in cases where a URL name is used to classify incoming traffic. An initial packet seen on the interface will be resolved to the domain name specified by the URL. Succeeding packets may not be resolved before timeout occurs on the link.

If using URL names, note that for most (though not all) of them, the prefix www is required. Most domain name strings will be in the form:

- www.<name>.(com, org, gov, net)

However other names, like the following, are also resolvable by DNS.

- ftp.<companyname>.com

Use of domain name parameters depends on the AP's ability to resolve the IP addresses of incoming packets. Owing to this, you must configure the AP's native DNS client prior to using these parameters in class definitions. See "Configuring the DNS Client" on Page 62 for information on configuring the DNS client.

USING THE MAX BANDWIDTH PARAMETER

You can use the max-bandwidth parameter to specify a bandwidth threshold for classes allowed to borrow.

For example, consider the following simple class definition:

```
> config cbq.1 traffic-class.ip-out parent root-input-tree bandwidth-allocation\
  500000 bounded false max-bandwidth 750000
```

This command configures traffic class ip-out with a bandwidth allocation of 0.5 Mbps. Because ip-out is not bounded (bounded is set to false), it is allowed to borrow idle bandwidth on the link. The max-bandwidth parameter sets a limit on the total bandwidth utilization for this class. In this case, max-bandwidth is set to 0.75 Mbps. This means ip-out is authorized to borrow no more than 0.25 Mbps beyond its allocation.

You can specify either an upper or lower limit for the average idle time with the *max-idle* and *min-idle* parameters. The *max-idle* parameter limits the credit given to a class that has underutilized its bandwidth allocation recently. If the *min-idle* parameter has a negative value, it indicates that a class has used more than its allocated bandwidth recently.

SETTING THE Q-ELASTICITY FACTOR

The q-elasticity-factor is a traffic shaping parameter for the number of packets that can be queued by a traffic class. Setting this number above the default value (100) increases the class' ability to buffer traffic during bursty periods.

This example shows how to use this parameter in conjunction with a class named *upload-finance*.

```
>config cbq.1 traffic-class.upload-finance q-elasticity-factor 300
```

Increasing a traffic class' buffer size reduces the potential for packet loss during such periods. Conversely, reducing the buffer size (for example, by setting the q-elasticity-factor to 0) means that few or no buffers exist to handle over-limit packets during such periods. In this case, packet loss is commensurate with the extent to which packet rates exceed a class' bandwidth allocation. The default value for q-elasticity-factor is 100. Acceptable values range from 0 to 1000 packets.

MANAGING TRAFFIC CLASS QUEUES WITH RANDOM EARLY DETECTION (RED)

CBQ provides Random Early Detection (RED) to prevent network congestion. In this case, congestion means a steady and possibly growing backlog of packets queued, waiting for transmission in a CBQ class. RED operates on queues that correspond to these CBQ classes, detecting incipient congestion, and then reducing it by randomly dropping packets before the congestion has deleterious effects on the connection.

RED detects congestion by monitoring the average queue size, rather than using a snapshot of the queue's size. This allows transient bursts of packets to pass efficiently through the class without intervention by RED.

When RED detects congestion, it drops a packet. This notifies the sender that there is congestion in the network. Dropping a packet is most effective with transport layer protocols such as TCP, where the connected host reacts to the dropped packet by decreasing its window size, thus slowing the rate of transmission and reducing congestion. For a connectionless protocol such as UDP, RED can also help preserve the packet buffer resources of the AP. Instead of simply dropping the tail end of the queue when the maximum queue size is reached, RED drops packets with increasing probability as the queue fills up, immediately freeing resources.

CONFIGURING RED

RED relies on two parameters, *q-elasticity-factor* and *red-minimum-queue-value*, to create a maximum and minimum range for the queue.

The *q-elasticity-factor* is the maximum number of packets that a particular class may queue for transmission. For example, if the *q-elasticity-factor* is set to the value 50, the class can queue up to 50 packets, regardless of the size of the packets in bytes. After the maximum number of packets is queued, any additional packets are dropped until the queue size is less than 50 packets. Remember that the queue size diminishes over time as the class transmits packets and removes them from the head of the queue. For additional information about the *q-elasticity-factor*, see "Setting the Q-elasticity Factor" on Page 290.

The *red-minimum-queue-value* activates RED queue management for a particular CBQ class. If it is set to 0, the default, RED is turned off. Configure this parameter to have a numerical value between 1 and the value of the *q-elasticity-factor* for the class on which you want to activate RED. When the number

of packets queued for the class is between the *red-minimum-queue-value* and the *q-elasticity-factor*, packets arriving at the class are randomly dropped.

To determine the proper *red-minimum-queue-value* for a particular class, think about the types of traffic and packet sizes likely to be passing through the class. Then balance these two metrics against the configured bandwidth allocation of the class.

For example, the following command configures traffic-class ftp-app to control FTP. While FTP typically uses large packet sizes, class ftp-app allocates it a low bandwidth. Using the *q-elasticity-factor*, the command sets a maximum queue size of 10 packets.

```
> config cbq.1 traffic-class.ftp-app application ftp bandwidth-allocation \
  16000 q-elasticity-factor 10
```

To share bandwidth equally among all connections passing through this class, while maintaining a small queue size, this command sets the *red-minimum-queue-value* to 3.

```
> config cbq.1 traffic-class.ftp-app red-minimum-queue-value 3
```

RED will randomly drop incoming packets with increasing probability after the average queue size reaches and then surpasses 3. Because of the random manner in which RED drops packets, the heaviest users of the connection will have more packets dropped than lighter users, thus sharing bandwidth.

As the average queue size increases, so does the probability that RED will drop a given packet that arrives at the class at a time when the class is queueing. The probability increases as the average queue size approaches the maximum queue size specified by the *q-elasticity-factor*. When the average queue size is equal to the maximum queue size, all arriving packets are dropped.

MONITORING RED

RED provides three parameters to monitor CBQ queues:.

Parameter	Description
red-average-queue-size	Average number of packets queued by this class.
red-dropped-packets	Number of RED mandated packet drops by this class. This counter does not include packets dropped because of a lack of space on the queue. That is, packets are dropped when the number of packets queued equals the class's configured *q-elasticity-factor* because at that point no space is available to store them.
red-probability	Current probability that the next packet arriving at this class will be dropped if the class is queueing when the packet arrives. The *red-probability* ranges between 2^0 - 2^16. The higher the value, the higher the probability that a packet will be dropped. A probability of 2^16 (65535) indicates that every packet will be dropped rather than added to the queue that corresponds to this class.

You can display RED statistics about a particular class as follows:

```
> show cbq.1 traffic-class.ftp-app red

Red Minimum Queue Value* = 3
        Red Probability  = 0
 Red Average Queue Size  = 0
    Red Dropped Packets  = 0
```

CLASSIFYING ON BGP COMMUNITY

You can use CBQ to classify packets based on the BGP Community attribute. The Community attribute groups destinations that have a common property. A community has no boundaries and is neither restricted to one AS nor one network. The Community attribute allows you to arbitrarily group advertised routes.

Additionally, the BGP Community attribute lets you associate an identifier with a route so you can assign an identifier to a set of routes for which you want the same treatment. This categorization of routes is called route coloring.

To classify packets based on the BGP Community, use the following source and destination parameters to match routes.

- src-bgp-community

- dest-bgp-community

When the src-bgp-community is configured, the source IP address of the packet is looked up in the routing table. If found, and if the route was added to the routing table by BGP (that is, it is a BGP route), the configured community attribute is compared to the classes' src-bgp-community value that corresponds to the route. Dest-bgp-community works in the same manner, but uses the destination IP address of the packet.

BGP Communities are 32-bits values expressed as two 16-bit values separated by a colon. Usually, the first 16-bit value identifies the autonomous system. A provider (AS256), for example, who wants to define a private community called "peer-routers" could use the community 256:1 represented in decimal notation. This translates to the 32-bit decimal value 16777217, or 0x1000001.

Some community values are reserved for well-known communities. The NO_EXPORT (0xffffff01) community indicates that routes carrying this community value should not be advertised to peers outside a confederation. The NO_ADVERTISE (0xffffff02) community indicates that routes should not be advertised to any BGP peer.

The following command matches the BGP Community value of 2:100:

```
> config cbq.1 traffic-class.community-source src-bgp-community 2:100 \
  parent root-input-tree
```

The following command shows the flow definition for community-source:

```
> show cbq.1 traffic-class.community-source flow-def
```

```
        Src Ip Addresses* = ""                    Operator* = and
       Dest Ip Addresses* = ""          Icmp Msgs Allowed* = ""
              Src Ports* = ""       Src Bgp AS Expression* = ""
             Dest Ports* = ""      Dest Bgp AS Expression* = ""
              Protocols* = ""          Src Bgp Community* = 2:100
           Applications* = ""         Dest Bgp Community* = 0:0
      Classification Tos* = ""    Datalink Traffic Indices* = ""
Classification TOS Mask* = 0x00     Datalink Traffic Mask* = ""
        Src Domain Names* = ""          Forwarding Profile* = ""
       Dest Domain Names* = ""
```

CLASSIFYING ON BGP AS PATH

You can use CBQ to classify packets based on BGP's AS_Path attribute. The AS_Path is a sequence of autonomous system (AS) numbers that indicate the ASs a packet has traversed to reach its destination. Classifying packets based on the AS_Path provides a way to treat separate flows with different policies depending on the route the flow will take to reach its destination.

To classify packets based on BGP AS_PATH, use limited regular expressions and the following source and destination parameters to match routes.

- src-bgp-as-expression
- dest-bgp-as-expression

When src-bgp-as-expression is configured, the source IP address of the packet is looked up in the routing table. If found, and if the route was added to the routing table by BGP (that is, it is a BGP route), the configured regular expression is checked against the AS Path that corresponds to the route. Dest-bgp-as-expression works in the same manner, but uses the destination IP address of the packet.

By using regular expressions, you can match patterns, letting you cast a wide net in classifying particular types of traffic. The following table shows these expressions and their meaning.

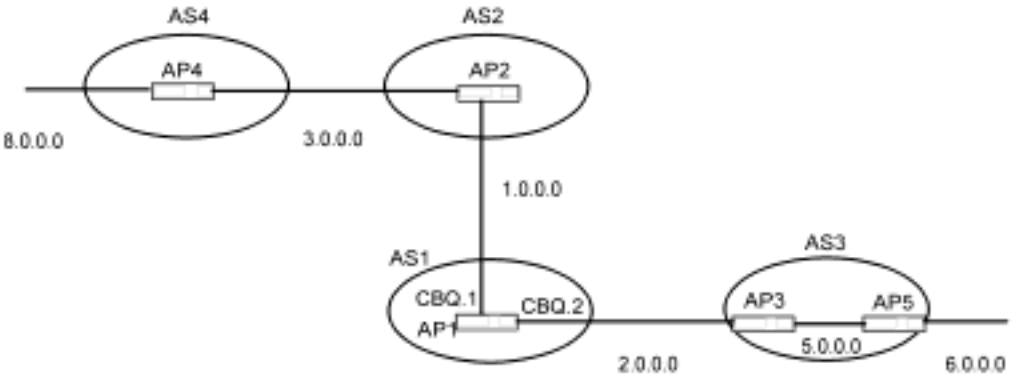
AS Path String	Matches
^$	Local routes only. All traffic is within the AS, so there is no AS Path.
*	All paths, including all transit traffic.
^*xx*$	A path containing exactly the path *xx* (an adjacent AS). Also defined as routes that originated from directly connected customers.
^*xx*	A path starting with *xx*.
xx$	A path ending with *xx*.
xx	A path containing *xx* anywhere in the path.
x.y	A path containing *x*, followed by *y*. The dot provides a one character wildcard match, including white space.

NOTE *Cisco Systems expressions and Access Point system*

expressions differ in the use of the underscore. With Cisco, the underscore, such as in _200$, limits the AS number to being exactly 200 and not 1200 or 2200. In the AP, the underscore is implied and optional.

To illustrate how to classify traffic using BGP's AS_Path attribute, we'll use the sample configuration shown in Figure 32. The configuration uses five routers (AP1-AP5) and four autonomous systems (AS1-AS4).

Figure 32 BGP AS_Path Sample Configuration



The following commands, based on the configuration illustrated in Figure 32, show how to configure CBQ to classify traffic from the perspective of AP1. (Note that for both the source and the destination, the AS_Path is the path from this AP to the source or destination, and the sequence of AS path numbers reverses order depending on if the packet is incoming or outgoing.)

In this example, CBQ.1 input on AP1 connects to AP2 over network 1.0.0.0, while CBQ.2 on AP1 connects to AP3 over network 2.0.0.0. The goal is to classify traffic on AP1's CBQ.1 and CBQ.2 input trees.

The following command classifies all traffic destined to directly connected networks:

```
> config cbq.1 traffic-class.in-local dest-bgp-as-expression ^$ parent \
  root-input-tree
```

The following command classifies all traffic destined to AS3:

```
> config cbq.1 traffic-class.in-transit-to-as3 dest-bgp-as-expression ^3 \
  parent root-input-tree
```

The following command classifies all other traffic that is not local, nor destined to AS3:

```
> config cbq.1 traffic-class.in-transit dest-bgp-as-expression * parent \
  root-input-tree
```

The following command classifies all traffic destined to directly connected networks:

```
> config cbq.2 traffic-class.in-local dest-bgp-as-expression ^$ parent \
  root-input-tree
```

The following command classifies all traffic destined to AS2:

```
> config cbq.2 traffic-class.transit-to-as2 dest-bgp-as-expression ^2$ \
  parent root-input-tree
```

The following command classifies all traffic passing through AS2, but not destined to AS2 (that is, all traffic going through AS2 to AS4). Class order is important in this series of commands. To accomplish this classification, class transit-to-as2 must precede this class.

```
> config cbq.2 traffic-class.transit-through-as2 dest-bgp-as-expression 2 \
  parent root-input-tree
```

After you have classified your AS paths, you can use CBQ to further classify traffic. For instance, you can apply bandwidth management to control transit traffic beyond AS3.

```
> config cbq.1 traffic-class.in-transit bandwidth-allocation 10000000 parent \
  root-input-tree
```

The CBQ tree for this configuration is shown below. Although destination AS Path classification is used in this example, source AS Path classification could also be used.

```
> show cbq.* traffic-class-tree

                        CBQ.* TRAFFIC-CLASS-TREE

    INTERFACE| ROW|                          Class Hierarchy  |
-------------|----|-------------------------------------------|
        cbq.1|   1|root-input-tree           Rate=100000000   |
        cbq.1|   2|  in-local                Rate=        0   |
        cbq.1|   3|  in-transit-to-as3       Rate=        0   |
        cbq.1|   4|  in-transit              Rate= 10000000   |
        cbq.1|   5|  in-default              Rate=        0   |
        cbq.2|   1|root-input-tree           Rate=100000000   |
        cbq.2|   2|  in-local                Rate=        0   |
        cbq.2|   5|  transit-to-as2          Rate=        0   |
        cbq.2|   7|  transit-through-as2     Rate=        0   |
        cbq.2|   9|  in-default              Rate=        0   |
```

CLASSIFYING ON DATALINK

You can use CBQ to classify packets based on datalink information. For example, you can enforce the same priorities you've set up for your VLAN traffic for traffic being sent over the WAN. Using CBQ, you can classify the WAN-bound traffic based on Tag Control Information (TCI) within the Ethernet Encoded Tag Headers. Datalink classification allows you to classify traffic based on incoming 802.1p priority bits and 802.1q VLAN identifier in the TCI or both.

CBQ classification on the TCI uses a range string and a mask value. To classify packets based on the datalink traffic, use the following parameters to match on priority or VLANs.

- datalink-traffic-indices

  The datalink-traffic-indices is a range list of 16-bit TCI values expressed in hexadecimal (optionally using the "0x" prefix notation).

- datalink-traffic-mask

  The datalink-traffic-mask is a mask which is applied to the TCI and each value in the datalink-traffic-indices range. The mask is also expressed in hexadecimal.

The TCI contains a three-bit priority value (802.1p), a one-bit canonical format indicator (CFI), and a 12-bit VLAN identifier (802.1q).

CLASSIFYING ON 802.1P

802.1p defines the three uppermost bits of the TCI as illustrated below.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
user priority			CFI	VLAN Id											

Using datalink-traffic-indices and datalink-traffic-mask, you can classify using just the 802.1p bits and ignore the remaining bits in the TCI. For example, you can individually classify priorities 0-3, and classify priorities 4-7 into a single class as shown in the following commands:

```
> config cbq.3 traffic-class.in-priority-0 datalink-traffic-class-indices 0 \
  datalink-traffic-mask 0xe000 parent root-input-tree

> config cbq.3 traffic-class.in-priority-1 datalink-traffic-indices 0x2000 \
  datalink-traffic-mask 0xe000 parent root-input-tree

> config cbq.3 traffic-class.in-priority-2 datalink-traffic-indices 0x4000 \
  datalink-traffic-mask 0xe000 parent root-input-tree

> config cbq.3 traffic-class.in-priority-3 datalink-traffic-indices 0x6000 \
  datalink-traffic-mask 0xe000 parent root-input-tree

> config cbq.3 traffic-class.in-priority-4-through-7 datalink-traffic-indices \
  0x8000-0xe000 datalink-traffic-mask 0xe000 parent root-input-tree
```

The CBQ tree for this configuration is shown below:

```
> show cbq.3 traffic-class-tree

INTERFACE| ROW|                                              Class Hierarchy|
---------|----|-------------------------------------------------------------|
    cbq.3|   1|root-input-tree                          Rate=100000000   |
    cbq.3|   2|  in-allowed-vlans                       Rate=         0   |
    cbq.3|   3|  in-priority-0                          Rate=         0   |
    cbq.3|   4|  in-priority-1                          Rate=         0   |
    cbq.3|   5|    datalink-2000-dst-ip-4.4.4.3-src-i>Rate=         0   |
    cbq.3|   6|  in-priority-2                          Rate=         0   |
    cbq.3|   7|    src-ip-11.11.11.7-in-priority-2      Rate=         0   |
    cbq.3|   8|      dst-ip-4.4.4.3-src-ip-11.11.11.7>Rate=         0   |
    cbq.3|   9|        datalink-4000-dst-ip-4.4.4.3-s>Rate=         0   |
    cbq.3|  10|  in-priority-3                          Rate=         0   |
    cbq.3|  11|  in-priority-4-through-7                Rate=         0   |
```

CLASSIFYING ON 802.1Q (VLAN)

802.1q defines the 12 lowest bits of the TCI as illustrated below.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
user priority			CFI	VLAN Id											

Using datalink-traffic-indices and datalink-traffic-mask, you can classify using just the 802.1q bits and ignore the remaining bits in the TCI. For example, you can classify all frames received on an Ethernet interface which contains VLAN IDs 64-80 and 128, without regard to 802.1p priority or the CFI bit as shown in the following command:

```
> config cbq.3 traffic-class.in-allowed-vlans parent root-input-tree \
  peer-classification-order 5 datalink-traffic-indices 0x40-0x50, 80 \
  datalink-traffic-mask 0x0fff
```

USING TRAFFIC PARAMETERS IN COMBINATION

When defining a traffic class, you can use any of the available CBQ parameters in combination. As an example, you might want to create a narrowly defined traffic class that specifies a specific host range, a particular protocol, and a single service.

In this case, the protocol and service specified are TCP and HTTP. A class like this might be used to allocate a fixed amount of bandwidth (for Web browsing) to a specific set of host machines.

```
> config cbq.1 traffic-class.finance-web src-ip-addresses \
  198.2.3.4-198.2.3.15 protocols tcp src-ports 80 bandwidth-allocation \
  50000 bounded false max-bandwidth 750000 parent departments
```

This definition applies only to IP packets that match all three of the criteria specified: that is, to packets that originate in a particular host range, are TCP-based, and are using HTTP (e.g., Web browsing).

CONFIGURING PEER CLASSIFICATION ORDER

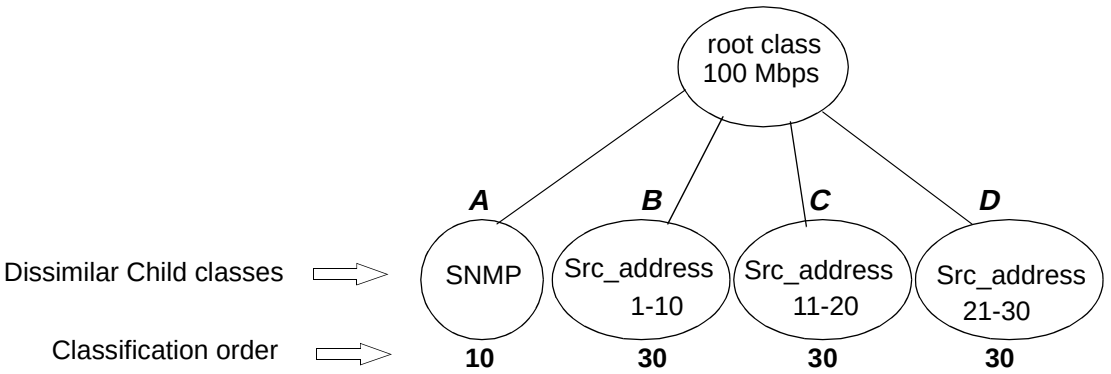
You can use the peer-classification-order parameter to specify the order in which CBQ processes packets. Specifying a classification order among peer classes is especially useful for providing a filtering mechanism within the CBQ tree. It also makes configuration easier, since it allows you configure classes with overlapping policies, without introducing processing ambiguity.

By default, all CBQ classes have a peer-classification-order of 100. When different classification orders exist among siblings, CBQ always processes classes with the lowest classification order first, and default classes last. This provides a handy way to filter out certain types of traffic, and enhances the security provided by classifying packets on input.

EXAMPLE: FILTERING SPECIFIC TRAFFIC TYPES

As an example, consider the tree shown in Figure 33, which shows a root class and four siblings. Class A is designed to match SNMP packets. Classes B, C, and D are designed to match packets coming from specific IP address ranges (1-10, 11-20, and 21-30).

Figure 33 Using Peer Classification Order



The classification order parameter for class A is 10. The other classes have a classification order of 30. This means CBQ will always attempt to match packets seen on the interface to class A before its siblings. In this case, all SNMP packets will be dropped because the entire definition for the class is as follows:

```
> config cbq.1 traffic_class.A_snmp src-ports snmp parent root-output-tree \
  peer-classification-order 10 bandwidth-allocation 0 bounded true
```

This traffic class definition guarantees that all SNMP packets will be dropped, since:

- *Class A will be matched against packets before its sibling classes*
- *Packets that match the class will be given zero bandwidth*
- *These packets will not be permitted to borrow additional bandwidth*

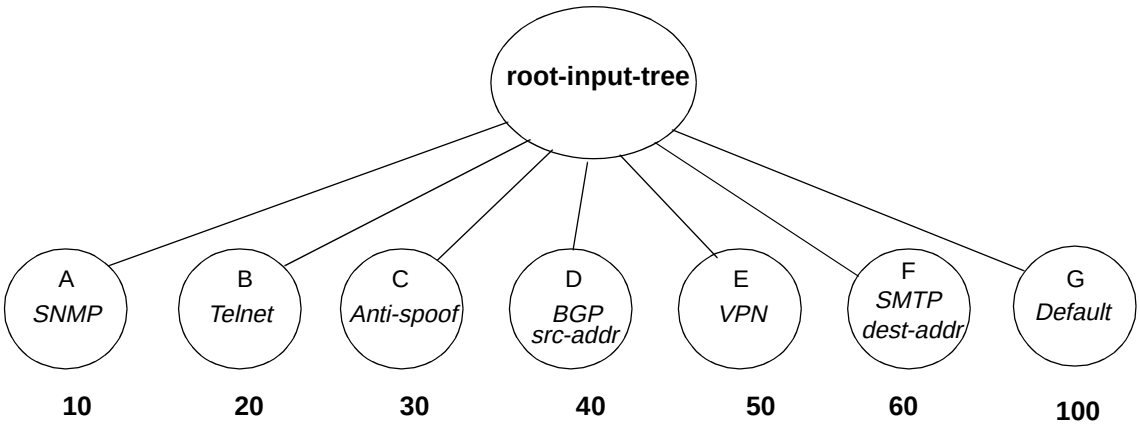
In sum, assigning a peer classification order of 10 to the SNMP class instructs CBQ to look for matches to this class first. This efficiently removes these packets before they are matched against other classes, and prevents the packets from being presented to any other AP interface.

EXAMPLE: USING PEER CLASSIFICATION TO ENHANCE SECURITY

The foregoing example shows how to use peer classification in a simple scenario. This feature can just as easily be used to specify a precise classification order for a large number of siblings.

In the sample tree shown in Figure 34, peer classification is used to enhance site security. Of the seven sibling classes attached to root-input-tree, three are filtered using a combination of peer classification and zero-bandwidth allocation. Three sibling classes are processed in a specific order. Another class (the default) exists to pass packets that do not match any of the classes.

Figure 34 Using Peer Classification to Enhance Security



PREVENTING AMBIGUITY

Because of the peer-classification-order assigned to these classes, they are processed in alphabetical order. When a packet appears on the interface, CBQ first checks to see if it matches to class A, then to class B, and so on.

This processing order prevents any ambiguity in processing that might otherwise arise. For instance, suppose an SNMP packet appears on the interface that matches classes A and E. Without a defined classification order, this would produce unpredictable results: some packets would get classified by A, and filtered; others would be classified by E, and routed into the VPN tunnel. Defining a processing order removes this possibility, by requiring CBQ to match every packet to class A first.

FILTERING UNWANTED TRAFFIC

As in the preceding example, the first three classes are designed to capture and filter unwanted packets. Each is a zero-bandwidth, bounded class: each captures and filters a specific type of IP packet.

- *Class A and B are designed to filter all SNMP and Telnet packets*

  As in the preceding example, these classes accomplish this by classifying the packets before any of their siblings, and by scheduling zero-bandwidth and no borrowing privileges.

- *Class C is designed as an anti-spoofing mechanism*

  This class works by matching and filtering out packets with a source address that matches to the site's internal network (1.2.3.0 -1.2.3.255). This class might be configured as follows:

  ```
  > config cbq.1 traffic-class.anti-spoof parent root-input-tree \
    peer-classification-order 30 bandwidth-allocation 0 bounded true \
    src-ip-addresses 1.2.3.0-1.2.3.255
  ```

- *Class D is designed to pass BGP packets from a specific source address*

  The only BGP packets classified and scheduled are from a specific peer BGP router (IP address 2.3.4.5). Class D allocates 50 Kb to this purpose, but can borrow additional bandwidth if needed (and available). A sample configuration for this class is:

  ```
  > config cbq.1 traffic-class.BGP-updates parent root-input-tree \
    peer-classification-order 40 bandwidth-allocation 50000 \
    src-ip-addresses 2.3.4.5 src-ports bgp
  ```

Note that the default operator (AND) is used in this operation. If an OR operator were used, the class would match packets for any of the classification parameters specified and would not filter the BGP packets as intended because BGP routes from any source address would be classified by D.

See Page 288 for more information on using the operator parameter.

Class E is designed for packets that belong to a configured VPN tunnel

The destination address for tunnel traffic is 1.2.3.7. Packets destined for this address are sent into the tunnel, provided they have not been matched by classes A, B, C or D. The class has 1 Mb of bandwidth, and can borrow additional as needed (bounded is NOT set to true).

```
> config cbq.1 traffic-class.VPN-traffic parent root-input-tree \
  peer-classification-order 50 bandwidth-allocation 1000000 \
  dest-ip-addresses 1.2.3.7
```

Note that the processing order in this example has ramifications for tunnel traffic. For example, this configuration would prevent SNMP and Telnet packets from passing through the tunnel. It is important to assess the effect of classification order on traffic passing through the AP.

Class F is designed to classify and schedule SMTP (mail) packets

This class's whole mission is to match and schedule SMTP packets destined for the mail server (IP address 1.2.3.25). Only packets that match both of these criteria are delivered.

```
> config cbq.1 traffic-class.smtp-mail parent root-input-tree \
  peer-classification-order 60 bandwidth-allocation 500000 \
  dest-ip-addresses 1.2.3.25 src-ports smtp
```

Class G is the Default

This class is designed to pass along all traffic that does not meet the match criteria configured for it siblings. By default, CBQ processes all default classes last, so there is no need to specify a peer classification order in this case. In no circumstances will G be processed before any of its siblings.

```
> config cbq.1 traffic-class.all-other-default parent root-input-tree \
  bandwidth-allocation 0
```

Note that, unlike classes A, B, and C, class G is unbounded. This means that G is able to borrow bandwidth from its parent as needed. This is a key distinction, since lack of bounded means that the class will pass all packets

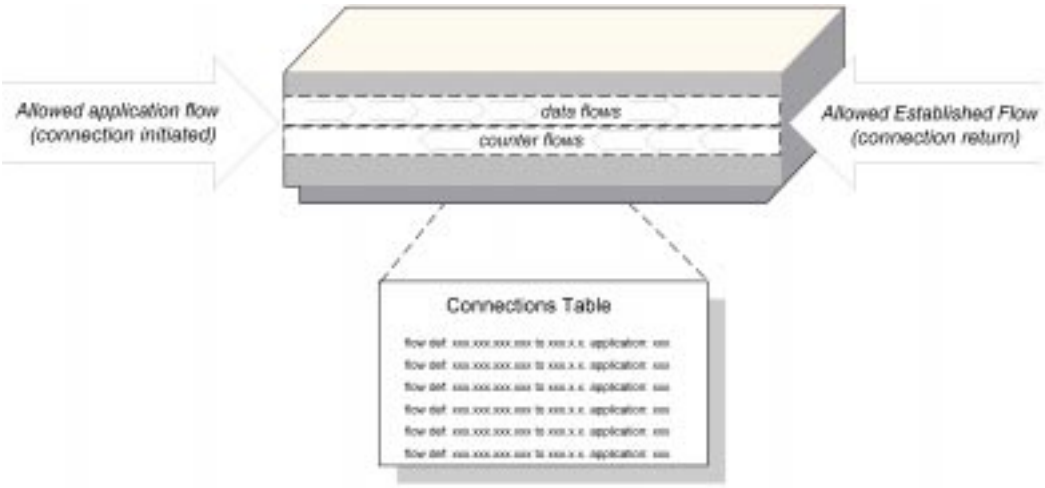
on a best-effort basis. In contrast, classes A, B and C have no bandwidth to offer from any source.

USING STATELESS AND STATEFUL CLASSIFICATION

CBQ allows for both stateless and stateful classification. Stateful classification provides additional firewall capability, and allows easier classification of commonly used services. For detailed information on setting up a firewall, see Chapter 8, "Configuring Firewall Parameters."

You can think of stateful application classification as dynamically creating and deleting temporary *flows* and *counterflows* through the CBQ firewall. The simplified picture in Figure 35 illustrates the use of flows and counterflows to create a stateful connection through the AP.

Figure 35 Stateful Connection Through the AP



You can use CBQ to statefully classify application flows through the AP. Each flow definition incorporates a complementary pair of CBQ classes. Each stateful pair functions like two ends of a narrowly defined corridor. This stateful corridor expressly allows connections and return flows using a specific application between designated hosts, networks, or domains. Other connection attempts using the same application are denied. The AP maintains detailed flow information in a connections table, allowing for visibility into connections into and out of the secured network.

When a packet initiates a traffic flow, a temporary memory location is created to allow return traffic via a counterflow. When the traffic flow stops, this tem-

porary memory location is deleted and CBQ shuts down the counterflows that depend on this location. To provide stateful classification for a service, you must create one class for the outbound traffic flow, and one class for the counterflow of this service.

How Stateless Classification Works

Stateless classification is the default, and most commonly used manner of classifying IP packets. This method works by matching packets to a traffic class based on the values contained in the packet's IP header. An example of this is a class called 'ALL-WEB' configured to match packets with the following characteristics:

- Source addresses: 1.2.3.0 - 1.2.3.255
- Protocol: TCP
- Application: Web

For each packet seen on the interface, CBQ checks for matches to these characteristics. Any packets that match all three criteria are matched to the ALL-WEB traffic class. These packets receive the bandwidth guarantees and borrowing privileges (if any) configured for this class.

Stateless classification is *static*, in the sense that CBQ uses the classification policy of a class to look for exact matches to packets flowing through an interface. (You can use an OR operator when configuring traffic classes to cause CBQ to look for at least one of the classification parameters listed, instead of matching all the classification parameters. For more on using the OR operator, see Page 288).

About Stateful Classification

Stateful classification works by opening up a port for a service in one direction (usually, outbound to the WAN), and creating a return path for the same service. We refer to these two paths as *flow* and *counterflow*, respectively. Once a flow definition is created in one direction, a return flow is established for return data.

There are two benefits to using stateful classification. First, this method tends to make it easier to configure certain well-known services, such as FTP. Second, stateful classification provides for additional firewalling capability.



For an example that shows how to use stateful classification along with CBQ's firewall capabilities, see "Sample Configurations" on Page 310.



See "Configuring Stateful Classification of FTP" on Page 313 for an example that shows how to build a stateful classification policy for a service that uses multiple port numbers for return of data.

ADVANTAGES OF STATEFUL CLASSIFICATION

Stateless classification depends on an exact match for packets flowing out, and flowing back through the firewall. For service types that use the same port for control and data, this poses no problem. However, some services use one port for control information, and can use a range of port numbers for the return of data. FTP provides a classic instance of this.

While FTP uses port 21 for control information and typically uses port 20 for data flows, the two end points (client and server) can switch to a high-numbered pair of ports for data flows. When this occurs, the high-numbered ports are chosen at random by the end points. This makes stateless configuration difficult, since the administrator must open up all conceivable return ports for return flow of FTP data. Doing this is inconvenient, from the perspective of configuring the service. Perhaps more importantly, it is inherently insecure: opening up all high-numbered ports to allow for a single service creates a wide avenue of entry that might be exploited by unwanted or malicious packets.

In contrast, stateful classification allows you to create a single class for return flow of TCP or UDP traffic. Classification of return traffic is based on the service type, rather than on the port. This makes it possible to aggregate all possible port pairings into a single class.

CONFIGURING STATEFUL APPLICATIONS

The AP can statefully classify the applications in Table 19. Doing so requires creation of one class for traffic flow to initiate a connection, and another for return flow of packets on that connection.

Table 19 Statefully Classifiable Applications

allTcp allTcpEstablished	tftp tftpEstablished	ldapSSL ldapSSLEstablished
allUdp allUdpEstablished	snmp snmpEstablished	nntp nntpEstablished
allIcmp allIcmpEstablished	httpSSL httpSSLEstablished	ntp ntpEstablished
ftp ftpEstablished	smtp smtpEstablished	pop3 pop3Established
telnet telnetEstablished	bgp bgpEstablished	radius radiusEstablished
http httpEstablished	slaProbe slaProbeEstablished	isakmp isakmpEstablished
dns dnsEstablished	ldap ldapEstablished	h323 h323Established
rtsp rtspEstablished	l2tp l2tpEstablished	ssh sshEstablished

The names listed here refer to the traffic flow used to initiate a connection. The return flow for each of these classes uses the suffix *Established*. So, to statefully classify allICMP packets, you would create a class as follows:

```
> config cbq.1 traffic-class.allicmp-out applications allIcmp parent \
  root-output-tree
```

The return flow for this class would use an *Established* application, as follows:

```
> config cbq.1 traffic-class.allicmp-back applications allIcmpEstablished \
  parent root-input-tree
```

While you can use a stateful class (such as nntp) to create an initial traffic flow, without using the appropriate *Established* class (in this case, nntpEstablished) for a return flow, return traffic will not be statefully classified. The AP requires

an instance of each class to statefully classify the application.

STATEFULLY CLASSIFYING OTHER APPLICATIONS

Generic client/server applications that are not in the list of supported applications can also be statefully classified. You can accomplish this by specifying the appropriate source and destination port, and transport application (allTcp, allUDP, or allIcmp). An example that uses this method to statefully classify Telnet is shown here.

```
>config cbq.1 traffic-class.tricky-telnet parent root-input-tree dest-ports 23
  applications allTcp

>config cbq.2 traffic-class.tricky-telnet-e parent root-input-tree src-ports 23
  applications allTcpEstablished
```

Doing this is, of course, unnecessary for Telnet since stateful classification for telnet is explicitly supported by CBQ. This is to say that CBQ is aware that the applications telnet and telnetEstablished use port 23, and are transported by TCP.

However, you can use this method to statefully classify other generic client/server applications that are not explicitly supported by CBQ. To do this, combine the source and destination port numbers with the appropriate transport protocol applications (either allTcp, allUdp or allIcmp).

For example, you can statefully classify the syslog protocol, a generic client/server application which uses UDP port number 514, as follows:

```
> add cbq.1 traffic-class.syslog parent root-input-tree applications allUdp
  dest-ports 514

> add cbq.2 traffic-class.syslog-e parent root-input-tree applications \
  allUdpestablished src-ports 514
```

SAMPLE CONFIGURATIONS

You can use the *applications* parameter to use ordinary names (rather than port numbers) when classifying many commonly-used applications. The syntax of this parameter is as follows:

```
> config cbq.3 traffic-class.<class name> applications <application type>
```

Configuring stateful classification for any application requires two classes: one for the application flow in one direction, and one for the return flow. Return flow classes are always distinguished by the suffix *Established*.

For example, providing stateful classification for Telnet requires two classes on separate CBQ trees. Assuming that Telnet packets will be sent from the LAN to some location on the WAN, you would create two classes:

- *One class for Telnet sessions on the input tree of the LAN interface*
- *One class for return of Telnet data on the input tree of the WAN interface. Alternatively, this second class could be configured on the output tree of the LAN interface. However, doing so would not offer any protection for service daemons within the AP.*

These two classes are used to accommodate the standard request-response paradigm used by most TCP-based services. The first class (telnet) initiates a connection; the second (telnetEstablished) captures state information from the connection request.

POSITIONING THE STATEFUL CLASSES TO PROTECT END SYSTEMS

For stateful classification to work properly, you must position ordinary classes (like telnet) and established classes (telnetEstablished, in this example) in proper relation to one another.

The AP shown in Figure 36 has two interfaces, and two instances of CBQ: CBQ.1 and CBQ.2. Each of these interfaces has a root-input-tree and a root-output-tree. This adds up to a possible total of four independent CBQ trees on the device. The question is, where should the administrator implement stateful telnet classes?

This is a key question, with important consequences. At bottom, CBQ is a classification and scheduling mechanism which allows you to provide guaranteed levels of service to certain classes. However, CBQ can also be configured as a denier of service, or firewall. The existence of independent CBQ trees for

input and output makes it possible to use CBQ with or without this built-in firewall capability.

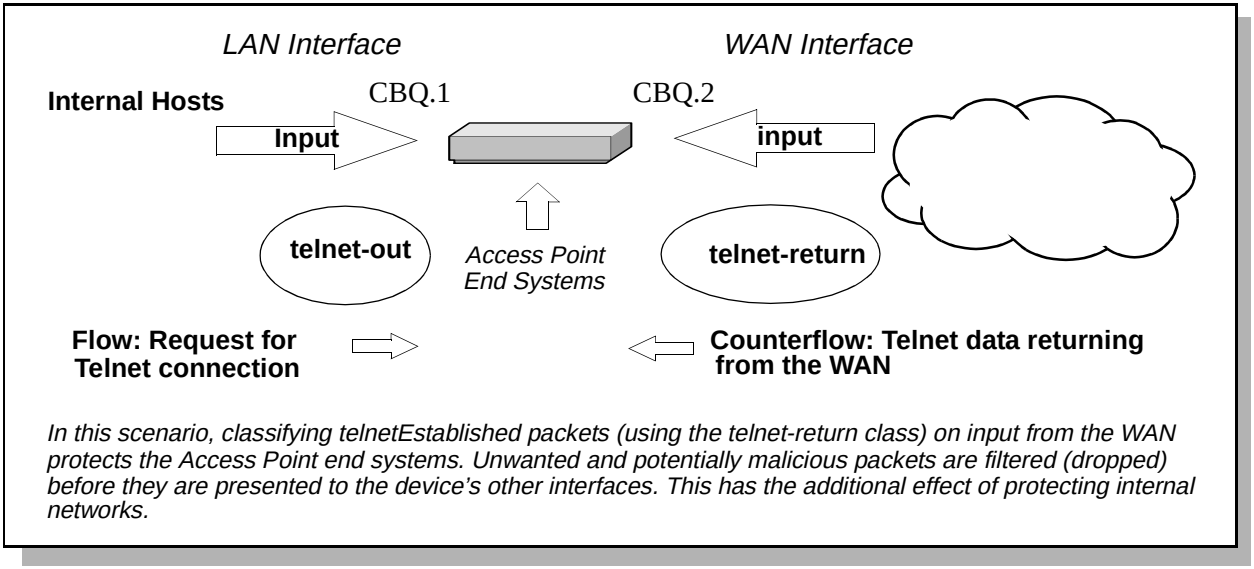
The two scenarios that follow are provided as illustrations. The first scenario leverages CBQ's firewall capability, while providing stateful classification of a particular type of service (Telnet). The second scenario does not leverage this capability, but uses two CBQ classes to provide stateful classification for the same service.

Generally, it is best to employ security measures wherever possible. CBQ's flexible implementation supports this. For this reason, scenario one is strongly preferred.

SCENARIO 1: STATEFUL CLASSIFICATION WITH FIREWALL CAPABILITY

The example shown in Figure 36 provides one answer. In this example, class telnet-out is a Telnet application on the root input tree of CBQ.1. The return data flow is placed on the root input tree of CBQ.2.

Figure 36 Sample Stateful Classification for Telnet



In this scenario, classifying telnetEstablished packets (using the telnet-return class) on input from the WAN protects the Access Point end systems. Unwanted and potentially malicious packets are filtered (dropped) before they are presented to the device's other interfaces. This has the additional effect of protecting internal networks.

Placing these classes in this relation to one another accomplishes two important objectives, one relating to QoS, the other to security:

1. It provides an overall classification policy for Telnet access to the WAN (QoS).
2. It protects end systems in the AP stack, and internal hosts, from infiltration by unwanted or malicious return data (Security).

Classifying packets *on input to CBQ.2* fulfills the second of these objectives. Doing so assures that only desired packets (those matching the telnet-return class are an example) are classified and scheduled. The AP drops all other packets without ever presenting them to the IP forwarders associated with the device's other interfaces.

Here is a sample definition for the outgoing (LAN to WAN) telnet class:

```
> config cbq.1 traffic-class.telnet-out applications telnet parent root-input-tree
  bandwidth-allocation 50000
```

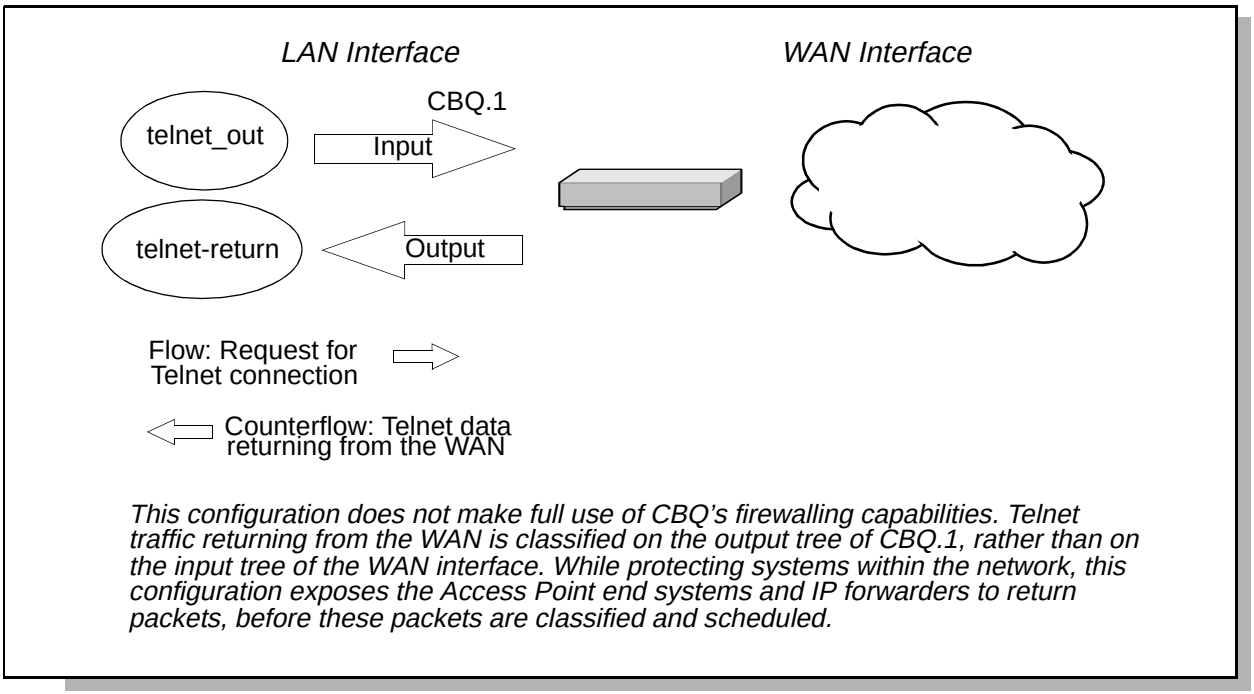
Here is a sample definition for the return (WAN to LAN) class. Note that rather than telnet, the application specified (using the applications argument) is telnetEstablished. An established class is required for the return flow of all statefully classified traffic.

```
> config cbq.2 traffic-class.telnet-return applications telnetEstablished parent
  root-input-tree bandwidth-allocation 50000
```

SCENARIO 2: STATEFUL CLASSIFICATION WITHOUT FIREWALL

Figure 37 illustrates an alternative configuration. In this scenario, Telnet is still statefully classified, however the AP end systems are not protected. What is notable about this case is that CBQ's firewall capabilities are not being used. Because this configuration classifies packets on the *output tree* of CBQ.1, the AP's end systems are exposed to all packets *before* these packets are classified by CBQ.

Figure 37 Sample Stateful Classification, Without Firewall



This configuration does not make full use of CBQ's firewalling capabilities. Telnet traffic returning from the WAN is classified on the output tree of CBQ.1, rather than on the input tree of the WAN interface. While protecting systems within the network, this configuration exposes the Access Point end systems and IP forwarders to return packets, before these packets are classified and scheduled.

CONFIGURING STATEFUL CLASSIFICATION OF FTP

The FTP protocol uses different ports for control information and data. In addition, FTP can use other high-numbered ports for return data flows as needed. Use of all of these ports adds complexity to configuring stateless FTP classes because a class must exist for every port that might be used by the protocol. This poses two difficulties:

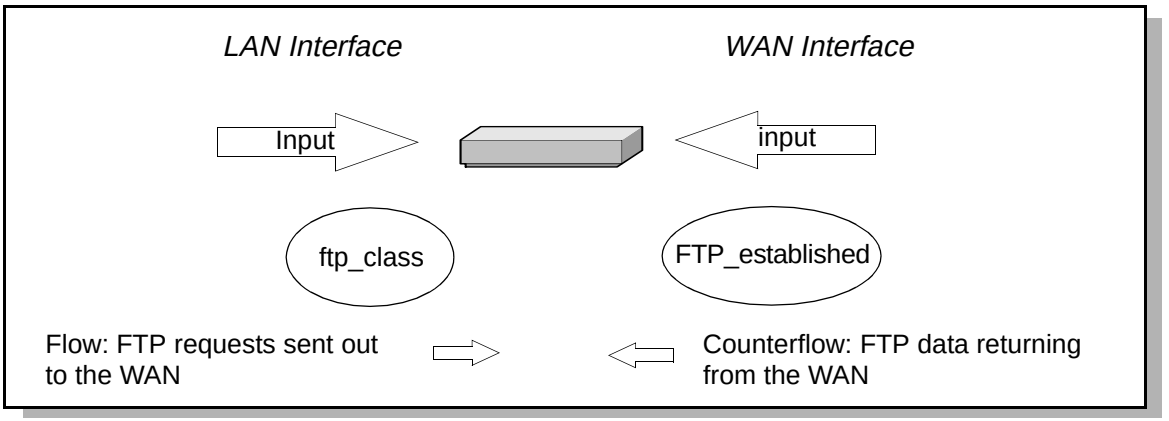
1. It is difficult to determine which high-numbered ports FTP will use for return data flows. The only solution is to create a class for the return flow on all high-numbered ports.
2. Opening all high-numbered ports for return traffic flows raises security concerns.

Using stateful classification for FTP solves this problem, by allowing a single class for all outgoing (request) flows, and a single class for all return flows. In addition, return flows can be filtered on input, allowing CBQ to drop unwanted packets on the interface (while allowing return data) before these packets

move up the protocol stack.

Figure 38 shows a sample implementation of stateful classification for FTP. In this example, a traffic class (called ftp_class) has been configured on the Input tree of the LAN side. This class is designed to allow FTP requests from certain LAN-side hosts. Another traffic class (called FTP_established) has been configured on the input tree of the WAN side.

Figure 38 Sample Stateful Classification for FTP



The purpose of the FTP_established class is to enable FTP data to return to the systems that have requested it. This traffic can arrive on port 21 or on a high-numbered port: FTP_ established matches whatever return port is chosen by the protocol. This makes it unnecessary to configure a whole group of ports for return data.

Table 19 lists applications that can be statefully classified by the AP.

STATEFUL CLASSIFICATION FOR TCP AND UDP APPLICATIONS

TCP is a connection-oriented protocol; UDP applications are inherently connectionless. Owing to this, stateful application classification works differently for TCP and UDP applications.

For applications running over TCP:

- Segments originating on the internal network are inspected to see if a TCP connection is being initiated (SYN+!ACK). If the packet is permitted, it is transmitted and a temporary entry is created to allow return traffic. Otherwise, the packet is dropped.

- Segments originating on the external network are inspected to see if they are part of an existing connection. If so, they are allowed to pass; otherwise, they are dropped.
- Some application segments are inspected to see if they are about to open data channels (e.g., FTP) or transition to high-numbered ports. If so, a temporary entry is created to allow the data channel traffic to pass.
- Segments are inspected to see if the connection is being closed (FIN). If the segments are about to close, the temporary entry is marked "close pending" and will be deleted after a short timeout (known as the TCP 2MSL wait) to allow both ends of the connection to complete negotiation of the close when faced with lost segments. This allows the "holes" in the firewall to be closed as soon as possible.

For UDP-based applications, stateful classification simulates UDP sessions as follows:

- Datagrams originating on the internal network cause a temporary entry to be created to allow return traffic.
- Datagrams originating on the external network are inspected to see if they are part of the return traffic and, if so, are allowed to pass; otherwise, they are dropped.
- Some application datagrams are inspected to see if they are about to open data channels or transition to high-numbered ports (e.g., TFTP). If so, a temporary entry is created to allow the data channel traffic to pass.
- After a period of inactivity, the temporary memory location representing the simulated UDP session's return traffic is deleted.

CONFIGURING STATEFUL CLASSIFICATION FOR MULTIMEDIA

The Real Time Streaming Protocol (RTSP) and the H.323 standard enable real-time delivery of multimedia — voice, video, data — over IP-based connections. The AP supports:

- RTSP (for applications such as RealNetworks' RealPlayer G2)
- H.323 (for applications such as Microsoft Windows' Netmeeting or Intel's Videophone)

Both RTSP and the H.323 use multiple ports and streams when transferring data. Statefully classifying RTSP and H.323 as applications eliminates the

complexity of dealing with the proliferation of ports and streams by allowing a single class for all outgoing flows, and a single class for all return flows. In addition, stateful classification provides firewalling capabilities by filtering returning packets on input, and dropping unwanted packets before they can move on to other interfaces.

CONFIGURING STATEFUL CLASSIFICATION OF RTSP

To statefully classify RTSP traffic, use the following commands:

```
> config cbq.1 traffic-class.rtsp-out parent root-input-tree applications rtsp \
  bandwidth-allocation 30000000

> config cbq.2 traffic-class.rtsp-return parent root-input-tree \
  applications rtspEstablished bandwidth-allocation 30000000
```

RTSP uses DNS, HTTP, and ICMP for the initial setup between the client and server. Therefore, you must statefully classify DNS, HTTP, and ICMP as shown in the following sample commands.

To statefully classify DNS:

```
> config cbq.1 traffic-class.dns-out parent root-input-tree applications dns

> config cbq.2 traffic-class.dns-return parent root-input-tree \
  applications dnsEstablished
```

To statefully classify HTTP:

```
> config cbq.1 traffic-class.http-out parent root-input-tree applications http

> config cbq.2 traffic-class.http-return parent root-input-tree \
  applications httpEstablished
```

To statefully classify ICMP:

```
> config cbq.1 traffic-class.allIcmp-out parent root-input-tree applications \
  allIcmp

> config cbq.2 traffic-class.allIcmp-return parent root-input-tree \
  applications allIcmpEstablished
```

CONFIGURING STATEFUL CLASSIFICATION OF H.323

To statefully classify H.323 traffic, use the following commands. If you classify h323 on the CBQ input interface, be sure that you classify h323Established on input for the other CBQ interface.

```
> config cbq.1 traffic-class.h323-out parent root-input-tree applications h323 \
  bandwidth-allocation 30000000

> config cbq.2 traffic-class.h323-return parent root-input-tree applications \
  h323Established bandwidth-allocation 30000000
```

Note that all streams related to the H.323 session are treated as a single entity, including voice, video, and data streams.

MANAGING CBQ

The AP provides a number of parameters for managing CBQ. Some of these are specific to an entire CBQ interface; some are used with specific traffic classes.

SETTING INTERFACE-WIDE CBQ PARAMETERS

CBQ includes a number of interface-wide parameters. You can use these parameters to exert global control over all classes on a specific CBQ instance. The parameters relate to the following areas:

- ☑ *Automatically removing AutoClasses at specified intervals*
- ☑ *Placing an upper-limit on the number of classes that can exist on the interface*
- ☑ *Gathering statistics on class utilization*

AGING OUT AUTOCLASSES

You can cause AutoClass children to age out of existence using two parameters. If you do not set an age-out parameter for AutoClasses, CBQ will continue to add new classes until the number of classes allowed on the interface has been reached. The two parameters used to age out AutoClasses are:

- cbq-autoclass-age-out
- cbq-autoclass-age-time

Basically, these two parameters work together to execute a specified action each time a specified amount of time has elapsed. The cbq-autoclass-age-time parameter specifies the time in seconds that the system must wait before taking action on *idle classes*. In this context, idle classes are those that have not transmitted packets during the period specified by cbq-autoclass-age-time. The

action taken could be to remove the class, age the class once, or do nothing. The cbq-autoclass-age-out parameter specifies the action to take. The values for cbq-autoclass-age-out are:

- doNotAge

  This action (or non-action) is the default. When set, doNotAge specifies that AutoClasses are not to be removed (aged) from the CBQ tree.

- ageOnce

  Specifies that idle classes will be removed only once, when the value of cbq-autoclass-age-time has been reached. After this, classes are not removed.

- agePeriodically

  Removes idle classes each time cbq-autoclass-age-time has elapsed. The timer runs continuously to remove idle classes on an ongoing basis.

EXAMPLES: AGING AUTOCLASSES

The default value for cbq-autoclass-age-out is doNotAge — a parameter which translates accurately as: leave all AutoClasses alone. However, you may want to prune your population of AutoClasses periodically, particularly if the number of classes attached to a CBQ instance has grown beyond several hundred.

The default for cbq-autoclass-age-time is 100 seconds. Minimum and maximum values for this parameter are 10 and 36000 (ten hours), respectively.

To establish 15 minutes as the cbq-autoclass-age-time parameter for cbq.1, you could use the following command:

```
> config cbq.1 cbq-autoclass-age-time 900 cbq-autoclass-age-out\
  agePeriodically
```

The age time specified here (900) is in seconds. This command assures that idle classes (those that have not transmitted packets for 15 minutes) will be removed from cbq.1.

Aging out classes periodically is a very common practice. However, you may want to age out classes only once, to remove a number of old idle classes. To accomplish this, use a cbq-autoclass-age-out value of ageOnce, as in this example:

```
> config cbq.1 cbq-autoclass-age-time 10 cbq-autoclass-age-out ageOnce
```

Entering this command removes all classes that have been idle for ten seconds (that is, classes that have not transmitted any packets for that period). After this one-time cleanup, AutoClasses continue to accrue until the maximum number allowed on the interface has been reached.

LIMITING THE NUMBER OF CLASSES ON AN INTERFACE

Turning AutoClass on causes CBQ to spawn classes automatically, based on information in IP packets that pass through a parent class. You can limit the total number of classes the AP creates on any interface using the cbq-maximum-classes-on-this-interface parameter. The default value for this parameter is 250; minimum and maximum values are 1 and 1,000 classes, respectively.

The form of this command is as follows:

```
>config cbq.<instance> cbq-maximum-classes-on-this-interface <number>
```

The number you specify enforces a limit on the aggregate class number. Once this limit has been reached, the AP begins deleting the oldest existing classes as new classes are added.

NOTE *It is strongly recommended that you not set cbq-maximum-classes-on-this-interface higher than 1,000 to avoid consumption of resources that might be needed by other software components.*

COLLECTING STATISTICS FOR TRAFFIC CLASSES

The AP provides a flexible mechanism for collecting CBQ statistics on a per-class or per-interface basis. You can configure the rate at which a common set of CBQ statistics are collected, and the number of different samples to keep on the box.

All statistics are stored in RAM and are not preserved across sessions. The sections that follow explain how to collect statistics on a global or per-class basis. The AP provides detailed information on all traffic that passes through CBQ. For information on showing statistics, see “Showing Statistics” on Page 321.

ENABLING GLOBAL COLLECTION OF STATISTICS

By default, cbq-history-status is disabled. To collect statistics for every class on the box, set this parameter to enabled, as follows:

```
> config cbq.<instance> cbq-history-status enabled
```

ENABLING STATISTICS COLLECTION FOR SELECTED CLASSES

You can also gather statistics on selected classes. You can accomplish this by enabling the history-status parameter for selected classes. To gather statistics on the classes all-out and all-web, enter this command:

```
> config cbq.1 traffic-class.all-out history-status enabled
```

```
> config cbq.1 traffic-class.all-web history-status enabled
```

The rate at which statistics are collected is controlled by the setting of cbq-history-timer as described below.

CONTROLLING THE RATE OF STATISTICS COLLECTION

The cbq-history-timer parameter controls the sampling rate for statistics collection. The default sampling rate is 5000 milliseconds (5 seconds). Acceptable values range from 50 milliseconds to 72,000,000 milliseconds (20 hours).

This command specifies a polling rate of 10 seconds on cbq.1:

```
> config cbq.1 cbq-history-timer 10000
```

NOTE *When collecting statistics, it is best to avoid setting the sampling rate higher than necessary. Doing so for a large number of classes may impact overall system performance. As a rule of thumb, rely on the default sampling rate unless your needs dictate otherwise.*

CONTROLLING COLLECTION OF STATISTICS

Use the cbq-history-table-size parameter to specify the size of the table that is created by gathering CBQ statistics. The default table size is 100. To set the table to another value (in this case, 250), use this command:

```
>config cbq.1 cbq-history-table-size 250
```

Each entry in the table represents one statistical sample. When the table is full, the most recent sample supersedes the oldest one it contains. The table may be sized at between 10 and 1000 entries.

SHOWING STATISTICS

You can show statistics on an interface-wide level, or for individual traffic classes. To show statistics for an entire interface, use this command:

```
> show cbq.1 traffic-classes history [aggregates | cbq-stats | flow]
```

To show statistics for a specific traffic class, use this command:

```
> show cbq.1 traffic-class.<class name> history [aggregates | cbq-stats | flow]
```

where

aggregates	Provides tabular information on: • aggregate bytes • aggregate packets
cbq-stats	Provides tabular information on: • High water • Idle • Queued packets • Over limits • Borrow attempts • Throttles • Borrowed bytes
flow	Provides tabular information on: • bytes • packets • dropped bytes • dropped packets

The short sections that follow describe each of these characteristics.

BYTES, AGGREGATE BYTES

The number of bytes sent by this class to date.

For interior classes, this counter represents the aggregate bytes sent by all child classes of this class. As such, this statistic is recursive, and the root class bytes counter will represent all bytes sent by all classes on the interface.

DROPPED BYTES

Closely tied to the dropped packets counter, dropped bytes specifies the aggregate total bytes dropped.

PACKETS, AGGREGATE PACKETS

Represents the number of packets sent by this class to date.

For interior classes, this counter represents the aggregate packets sent by all child classes of this class.

DROPPED PACKETS

If a packet is received for transmission by a class which is queueing, and cannot queue any more, that packet is dropped. The dropped packets counter represents the total number of packets dropped under these circumstances.

HIGH WATER

This statistic represents the highest number of packets that has been queued for this class since the last time the class was reset.

IDLE

This statistic represents the average idle time, or derivation from the interpacket gap that is expected given the bandwidth allocation. The statistic is expressed in tens of nanoseconds. For instance, if you have a 1 Kbps class, that class may transmit 125 bytes per second. At a rate of 125 bytes per second, it takes 8000000 nanoseconds to transmit each byte. Therefore, the class is eligible to transmit a byte every 8000000 nanoseconds (every 0.008 second).

If the class is sending 125 bytes per second, then the idle would be 0, as the actual rate of transmitting bytes is exactly 8000000 nanoseconds. However, if the class is sending 2 Kbps (250 bytes per second), then the rate would be a byte every 4000000 nanoseconds, which is (8000000 - 4000000) a total of 4000000 nanoseconds difference (and negative, as the class is overlimit) from the expected rate. Therefore, the idle would be -4000000.

QUEUED PACKETS

The number of packets currently queued and awaiting transmission.

OVER LIMITS

This statistic records instances of a class crossing the line between a positive (underlimit) and negative (overlimit) average idle. When a class is transmitting perfectly at its rate, it has an average idle of zero, and a limit status of at limit.

- The overlimit counter increments each time a packet is transmitted from a

class which has been recently transmitting more than its allocated rate. (The class was able to borrow, but still has used more than its allocation.)

- The overlimit counter also increments each time a packet cannot be transmitted because the class to which its flow belongs is overlimit and cannot borrow.

Borrow Attempts

This statistic records instances of a class trying to borrow. A class may attempt to borrow when it receives a packet and is overlimit, or when it is dequeuing and has gone back overlimit itself and wishes to borrow to continue to drain its queue.

Note that all of these statistics apply recursively, that is, interior and leaf classes both keep these statistics.

Throttles

This statistic represents an occurrence of a request to transmit that is denied. This may happen when a class has been borrowing bandwidth and desires to borrow more bandwidth to transmit a packet, but cannot borrow because no ancestors have bandwidth to lend. This statistic is also incremented when a class is being delayed, and a packet arrives at the class wanting to be transmitted, but is queued instead.

Borrowed Bytes

This statistic records the number of bytes a class borrowed.

USING AUTOCLASS

The AutoClass feature allows traffic flows to be automatically classified and scheduled by AP interfaces. When AutoClass is turned on for a traffic class (auto-class-enabled=true), child classes are classified (created) and scheduled (assigned bandwidth) based on the packets that match the parameters configured for the AutoClass parent.

Running AutoClass produces a volatile traffic classification hierarchy. The hierarchy can be made part of the AP's configuration by saving it to CMEM. If not saved, the hierarchy is lost on reboot. If saved, the hierarchy can be manually edited and refined by the administrator to meet existing business goals.

Parameters associated with AutoClass are:

- auto-class-enabled (true/false)
- auto-class-child-bounded (true/false)
- auto-class-sub-tree-depth (integer, 0-3)
- auto-class-child-bw-allocation (integer, number in bps)
- save-auto-class-children (true/false)
- auto-class-child-max-bw-alloc (integer)
- auto-class-child-src-ip-divisor
- auto-class-child-dest-ip-divisor

SAMPLE AUTOCLASS CONFIGURATION

Configuring AutoClasses is a straightforward process that involves using a set of specialized parameters. When configuring a class at the command line, you can use these parameters in any order. The example developed below shows how to use these parameters to create and develop an AutoClass definition.

To configure AutoClass, use the auto-class-enabled parameter in conjunction with other relevant parameters. As an example, let's consider this sample definition for a class called *outbound-hosts*.

```
> config cbq.1 traffic-class.outbound-hosts dest-ip-addresses \
  125.2.4.18-125.2.4.50 parent root-output-tree bandwidth-allocation
  10000000
```

This definition declares a class (outbound-hosts) with a bandwidth allocation of 10 Mbps and a parent of root-output-tree. To turn this class into an AutoClass parent, set the auto-class-enabled parameter to true as shown:

```
> config cbq.1 traffic-class.outbound-hosts auto-class-enabled true\
  auto-class-child-bw-allocation 50000 dest-ip-addresses 125.2.4.18-
  125.2.4.50 parent root-output-tree bandwidth-allocation 10000000
```

Enabling AutoClass causes the class outbound-hosts becomes an AutoClass parent whose children are created in accordance with specific classification parameters. In this case, the classification parameters are limited to a destination IP address range, a bandwidth allocation for the AutoClass parent (10 Mbps), and a separate bandwidth allocation for AutoClass children (50 Kbps).

Now, packets seen on CBQ.1 that match to this address range automatically generate AutoClass children. Moreover, each child class has bandwidth allocated to it, and can borrow additional bandwidth on the link as needed.

CONTROLLING AUTOCLASS CHILD MAX BANDWIDTH ALLOCATION

The auto-class-child-max-bw-alloc parameter allows you to specify the maximum bandwidth that can be allocated to an AutoClass child. To use the parameter, proceed as shown here.

```
> config cbq.1 traffic-class.outbound-hosts auto-class-enabled true\
  auto-class-child-bw-allocation 50000 auto-class-child-max-bw-alloc 60000
  dest-ip-addresses 125.2.4.18-125.2.4.50 parent root-output-tree\
  bandwidth-allocation 10000000
```

USING THE AUTOCLASS CHILD IP DIVISOR

Two parameters exist to allow autoclass child packets to be divided, by source or destination IP address, into ranges. The two parameters used to specify the type and number of ranges are:

- auto-class-child-src-ip-divisor
- auto-class-child-dest-ip-divisor

The IP divisor number provided can be any non-zero number. Here is a sample command using an AutoClass child source IP divisor of 4 on a class called HTTP-in.

```
> config cbq.1 traffic-class.http-in auto-class-enabled true src-ip-addresses
  198.200.0.0-198.200.255.255 parent root-input-tree auto-class-child-src-ip-
  divisor 4
```

Use of the divisor segregates packets that match HTTP-in into four dynamically created and equal ranges. In this case, the ranges are:

198.200.0.0 - 198.200.0.63

198.200.0.64 - 198.200.0.127

198.200.0.128 - 198.200.0.191

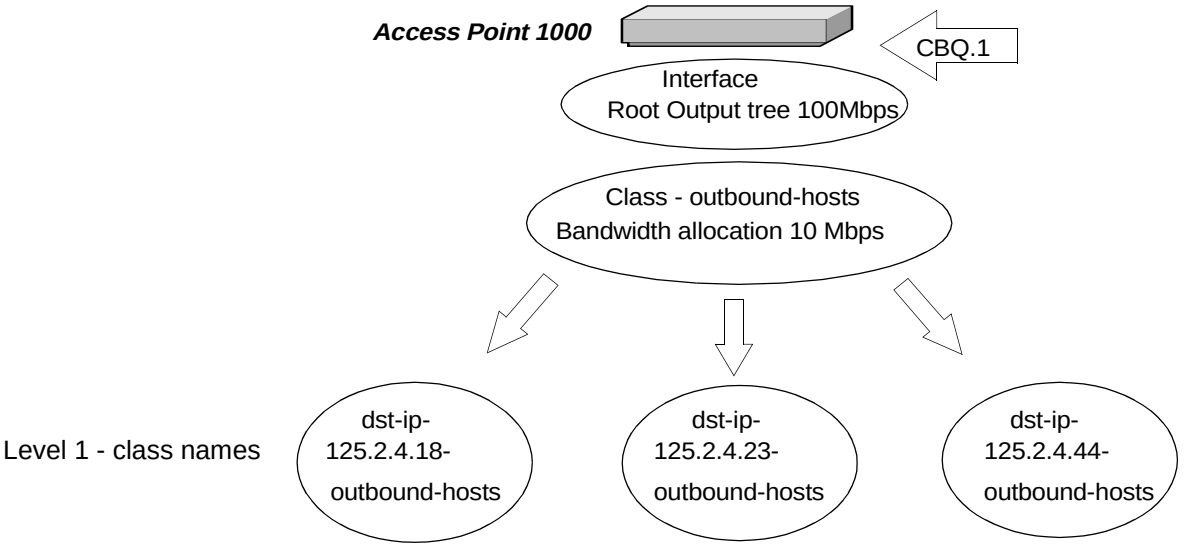
198.200.0.192 - 198.200.0.255

An IP source address of 198.200.0.18 will fall into the range of 198.200.0.0 - 198.200.0.63, and so on.

THE RESULTS OF TURNING ON AUTOCLASS

Figure 39 shows a possible result of traffic classification based on the class *outbound-hosts*, configured earlier (see Sample AutoClass Configuration on Page 325). A burst of packets destined for IP addresses 125.2.4.18, 23, and 44 arrive on CBQ.1. The CBQ classifier matches these packets against classification parameters established for outbound-hosts. The packets flow through outbound-hosts accordingly, and three distinct classes are automatically generated.

Figure 39 AutoClass Generation, 1 Level Subtree



SPECIFYING SUBTREE DEPTH

Subtree depth is controlled by the auto-class-sub-tree-depth parameter, which can be set from 1 - 3 levels deep. The default depth for any subtree is 1. This is the depth of the AutoClass subtree illustrated in Figure 39.

In the case of this example, only one classification parameter — destination IP address — is in use. Hence, the subtree generated can only be one level deep. However, the addition of additional parameters could generate up to two levels of child classes beneath a primary AutoClass child. In this case, the AutoClass tree could be up to three levels deep. The actual depth of the subtree depends, however, on the setting of auto-class-sub-tree-depth. No matter how many classification parameters are supplied, the subtree's depth will always be determined by the setting of this parameter.

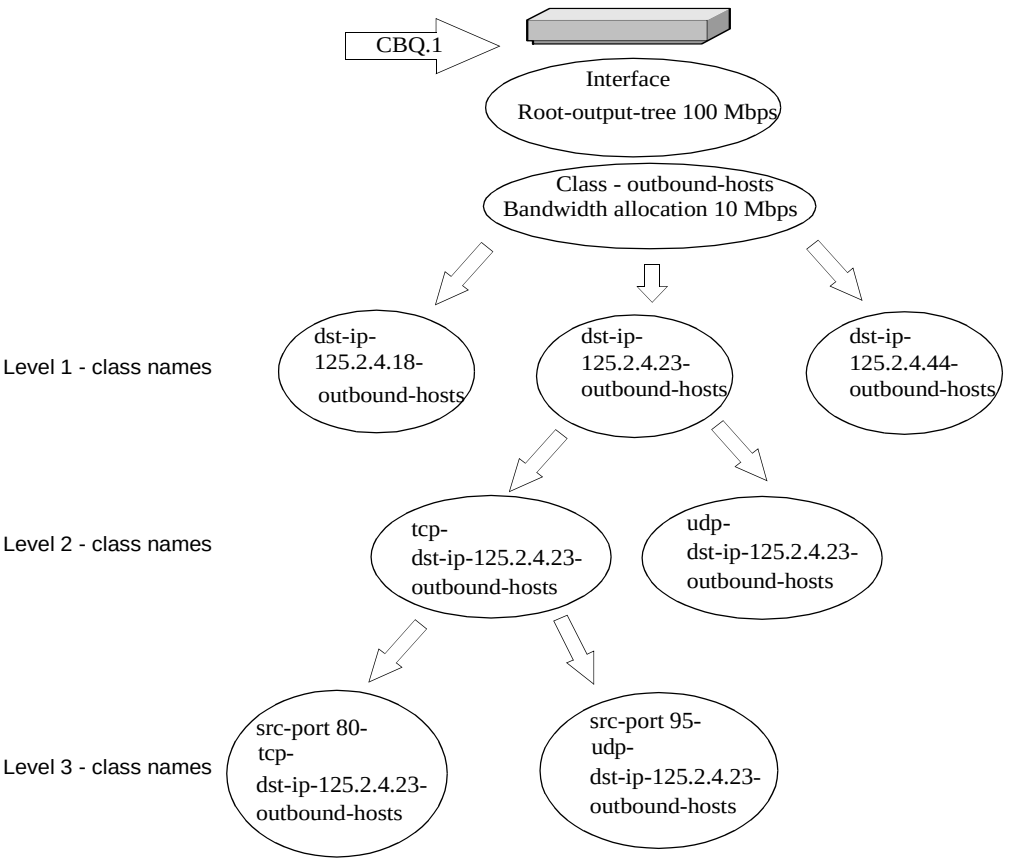
GENERATING A THREE-LEVEL SUBTREE

To illustrate, consider an amplification of the example. The addition of two new classification parameters and an auto-class-sub-tree-depth parameter of 3 will affect the way the subtree grows, and the depth of the subtree.

```
> config cbq.1 traffic-class.outbound-hosts auto-class-enabled true\
  auto-class-child-bw-allocation 50000 auto-class-child-bounded false\
  auto-class-sub-tree-depth 3 dest-ip-addresses 125.2.4.18-125.2.4.50 \
  protocols tcp-udp src-ports http-100 bounded false parent root-output-tree \
  bandwidth-allocation 10000000
```

Provided matching packets are seen on the interface, these additions will deepen the AutoClass subtree and spawn two levels of additional classes. The picture of a new, hypothetical tree shown in Figure 40 illustrates this point.

Figure 40 AutoClass Generation, 3 Level Subtree



AUTOCLASS NAMING CONVENTIONS

As packets are matched against the AutoClass outbound-hosts, an AutoClass subtree grows down three levels to accommodate the number of classification parameters provided and the auto-class-sub-tree-depth setting.

As classes are created, each receives a unique name that identifies its parentage and position in the tree. At each level in the subtree, AutoClass derives child class names from two factors:

- The name of the classification parameter operative at that level
- The name of the AutoClass parent, appended as a suffix

Examine the class names generated at level 1 of the tree shown in. The names of the three child classes are:

- dst-ip-125.2.4.18-outbound-hosts
- dst-ip-125.2.4.23-outbound-hosts
- dst-ip-125.2.4.44-outbound-hosts

For each name, the destination IP address is followed by the name of the parent class (outbound-hosts). Adding the parent class's name as a suffix is the way all AutoClass names are derived. For instance, at Level 2, one AutoClass child is named:

- tcp-dst-ip-125.2.4.23-outbound-hosts

The name is formed by adding the parent name as a dashed suffix to the Level 2 classification parameter. Level 3 classes are generated in the same way, as shown in Figure 40.

AUTOCLASS NAME RESTRICTIONS

As discussed earlier, AutoClass generates the names of child classes automatically, based on two factors:

- The name of the AutoClass parent
- Information in transmitted IP packets

For this reason, certain strings are reserved and cannot be used. These strings are the tokens CBQ uses to generate AutoClass names. Some tokens (such as tcp-) derive from the names of CBQ parameters; others (such as ftp-data-) derive from the name of services. The restriction is necessary to prevent two or more classes with the same name from appearing in the CBQ tree.

For example, a CBQ class named src-ip-10.2.3.4 could cause a name conflict. This conflict would arise if a packet that actually has this source IP were to spawn an AutoClass.

Table 20 lists the tokens that cannot be used in creating CBQ class names. Notice that each prefix ends with a dash. Altering any prefix by removing the trailing dash (such as by substituting ftp for ftp- in a class name) makes it acceptable for use in a class name.

Table 20 Invalid CBQ-Class Name Prefixes

src-ip-	icmp-	udp-	src-port-	dst-port-
dest-ip-	tcp-	ospf-	proto-	fs-
http-	systat-	time-	kerberos-	sunrpc-
ftp-data-	daytime-	name-	supdup-	uucp-path-
ftp-	netstat-	whois-	hostnames-	nntp-
telnet-	qotd-	bootps-	iso-tsap-	news-
tftp-	msp-	bootpc-	x400-	snmp-
tcpmux-	chargen-	rje-	x400-snd-	snmptrap-
echo-	smtp-	finger-	csnet-ns-	bgp-
discard-	nsw-fe-	link-	pop-2-	clearcase-
ldap-	netcp-	login-	printer-	talk-
rip-	timed-	courier-	uucp-	new-rwho-
rmonitor-	monitor-	pcserver-	doom-	kerberos-adm-
rfile-	ufsd-	xaudio-	ingreslock-	listen-
nfsd-	lockd-	dtspc-		



Tip: Naming CBQ Classes

Because each of the restricted tokens shown in Table 20 uses at least one dash, not using this character ensures that no conflicts arise between class names you create, and those generated by AutoClass.

MORE ON AUTOCLASS SUBTREE DEPTH

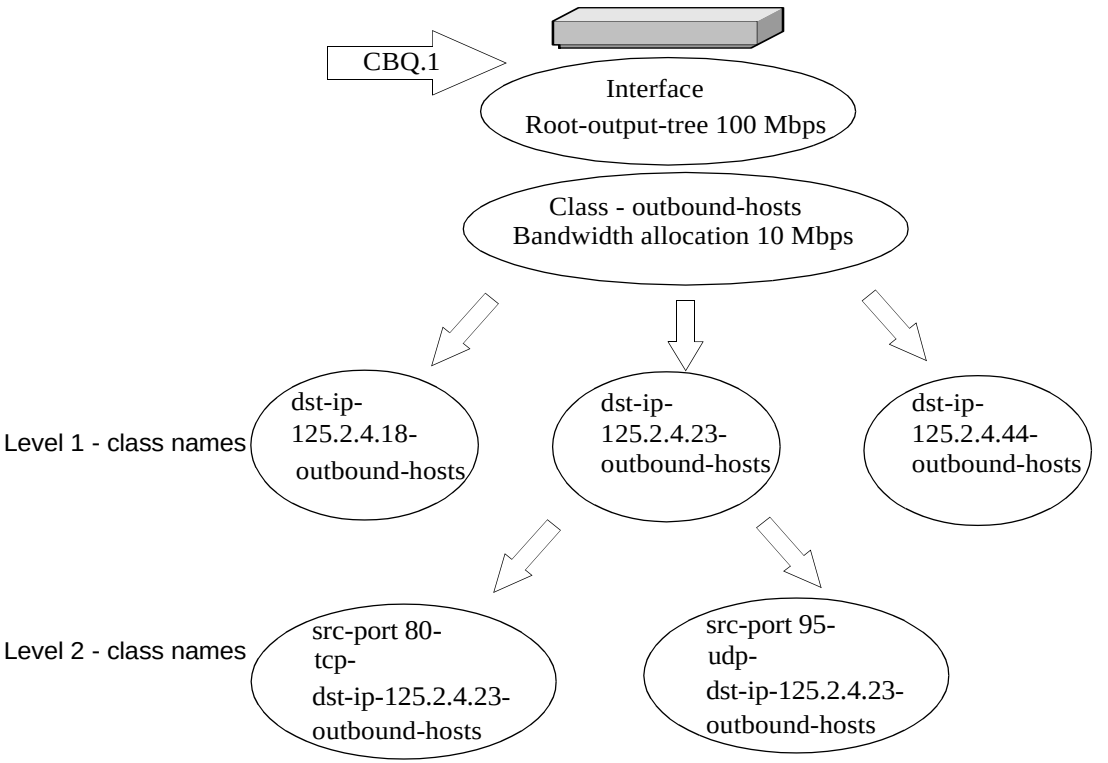
You can set auto-class-sub-tree-depth to 1 (the default) or 2, and still use three classification parameters. The resulting subtree will be as deep as you specify, and individual class names will be adjusted accordingly.

For example, you could change the subtree depth in the preceding example by issuing an additional command:

```
> config cbq.1 traffic-class.outbound-hosts auto-class-sub-tree-depth 2
```

The resulting tree would be two rather than three levels deep. Class names would be generated in the same manner. In this case, classes that had existed at Level 3 would move to Level 2. Figure 41 illustrates this with a modified version of the picture shown in Figure 40.

Figure 41 AutoClass Generation, 2 Level Subtree



As this picture shows, changing the auto-class-sub-tree-depth for outbound hosts has the effect of removing the Level 2 class shown in Figure 40. All classification parameters are preserved within the child class names at the (new) Level 2, as are their AutoClass names. Each name defines its class' position in the subtree, and its parentage.

SAVING AUTOCLASS CHILDREN

To save AutoClass children, use the save-auto-class-children parameter together with a parameter that matches to the depth of your subtree. Supported completion values are:

- oneLevel
- twoLevels
- threeLevels

A sample command is shown below.

```
> config cbq.1 traffic-class.outbound-hosts save-auto-class-children \
  twoLevels
```

This command saves the AutoClass subtree, including all AutoClass child classes, to the permanent configuration memory of the AP.

EDITING AN AUTOCLASS SUBTREE

After saving AutoClass children, you can selectively edit the CBQ tree to create a structure that meets your business needs. Editing the tree can encompass any of a variety of operations, including:

- Increasing the bandwidth of classes that are operating at near peak capacity
- Pruning away classes that had very low utilization within a broad period of time
- Adding borrowing capabilities for classes with bursty traffic characteristics

To edit class definitions at the command line, use any of the following commands to see class characteristics.

```
> show cbq-interfaces traffic-class-tree
```

```
> show cbq-interfaces traffic-classes flow-def
```

You can edit the definition of any class using *config* commands in conjunction with the class name. For instance, to change bounding or bandwidth allocation for a class, enter one of the following commands.

```
> config cbq.1 traffic-class.src-port 80-tcp-dst-ip-125.2.4.23-outbound-hosts\
  bandwidth-allocation 200000

> config cbq.1 traffic-class.src-port 80-tcp-dst-ip-125.2.4.23-outbound-hosts\
  bounded true
```

You can enter new configuration parameters on the same line, rather than in separate commands as shown here.

To remove a class (and, by definition, all of its attached children), use the row-status-destroy parameter:

```
> config cbq.1 traffic-class.src-port 80-tcp-dst-ip-125.2.4.23-outbound-hosts\
  row-status destroy
```

A Few More Words on AutoClass...

When using AutoClass, bear the following facts and limitations in mind:

- AutoClass children and the subtree in which they exist are not part of the AP's configuration memory. (They can be saved to the AP's permanent configuration, as described in "Saving AutoClass Children" on Page 333.)
- Changing the definition of an AutoClass parent deletes all of its children.
- ToS and domain name classification parameters cannot be used in conjunction with AutoClass.
- The max-bandwidth parameter is not supported for AutoClasses. See "Using the Max Bandwidth Parameter" on Page 290.

USING THE TOS PARAMETERS TO IMPLEMENT DIFF-SERV

Access Point systems support configuration of the ToS (Type of Service) byte in IP packet headers so administrators can read or write Diff-Serv markings to support Service Level Agreements between network providers (such as ISPs) and their customers.

Two categories of actions are supported: Classifying based on ToS, and writing to the ToS byte of specified packets.

CLASSIFYING AND SCHEDULING BASED ON TOS VALUE

The AP uses two parameters to classify and schedule packets based on their ToS values. You can use these parameters to set up a traffic class that incorporates ToS values assigned to IP packets. Entry values for each command are expressed in hex.

- classification-tos

  The hex values (converted to binary in the tos field) that specify the start and end of the range of bit patterns that CBQ will match against the masked area of IP packets. Enter two hexadecimal values separated by a hyphen to specify a bit range.

- classification-tos-mask

  This hex value (converted to binary in the tos field) specifies the bits in the ToS byte of IP packets that will be read and matched against by CBQ.

The following example shows how to configure a traffic class (named tos1). The mask indicates the only bits CBQ will match against. The remaining bits (0-3 and 7) are ignored. The start and end range specifies the range of bit patterns that will yield a match.

```
> config cbq.1 traffic-class.tos1 classification-tos 0-30 \
  classification-tos-mask 70
```

A simple picture of the ToS byte and the values set within it by each command

illustrates how the commands work.

ToS Command Values (read)	Bits Set in ToS Byte							
Bit Number	7	6	5	4	3	2	1	0
classification-tos start = 0	0	0	0	0	0	0	0	0
classification-tos end = 0x30	0	0	1	1	0	0	0	0
classification-tos-mask = 0x70	0	1	1	1	0	0	0	0

In this case, only IP packets containing one of the four following values in bits 4 - 6 of the ToS field of IP packets are classified (matched) as belonging to the class tos1.

000, 001, 010, 011

Note that the values 110 (0x60) and 111 (0x70) are not included among matches, even though they fall within the specified mask. Each exceeds the classification-tos end value (0x30).

WRITING TO THE TOS FIELD OF SPECIFIED PACKETS

The ability to write to the ToS field of specific types of IP packets allows the AP to mark these packets for special treatment by core routers in ToS-aware networks. The parameters to use to write to the ToS field are:

- tos-value

  For non-borrowing packets, this hex value specifies the value to be written to the designated tos-mask. It only applies to packets running at or below specified bandwidth allocation for the class.

- tos-mask

  For non-borrowing packets, this hex value specifies the bits in the ToS byte into which the tos-value will be written by CBQ. It only applies to packets running at or below specified bandwidth allocation for the class.

- tos-borrow-value

  For packets borrowing idle bandwidth, this hex value specifies the value to be written to the designated tos-mask. It only applies to classes for which bounded has been set to false.

- tos-borrow-mask

  For packets borrowing idle bandwidth, this hex value specifies the bits in the ToS byte into which the tos-value will be written by CBQ. It only applies to classes for which bounded has been set to false.

You can use ToS values to distinguish between packets belonging to non-borrowing and borrowing classes, and set different ToS byte values for them. As in the case of reading ToS values, the mask parameter (tos-mask) indicates which bits are to be acted upon.

SETTING A TOS VALUE ON NON-BORROWING PACKETS

In this context, non-borrowing packets fall into either of two categories.

- Packets belonging to a bounded class (e.g., a class prohibited from borrowing additional bandwidth)
- Packets belonging to a class whose bandwidth utilization is at or below its allocation

You can use this feature to write a specific ToS value into all non-borrowing packets with specific classification parameters. For example, the command that follows sets a ToS mask in bits 4 - 6, and a ToS value of 0x10 for all packets destined for hosts in the range 1.2.3.4 - 1.2.3.255.

```
> config cbq.1 traffic-class.angel dest-ip-addresses 1.2.3.4-1.2.3.255 \
  tos-value 10 tos-mask 70 bandwidth-allocation 500000 \
  parent root-output-tree
```

Note that while this simple example writes to the ToS field on the basis of a single classification parameter (destination IP address range), multiple parameters can be used in combination. The ToS bits set by this command are shown

below.

ToS Parameter Values (write)	Bits Set in ToS Byte							
Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
tos-value = 0x10	0	0	0	1	0	0	0	0
tos-mask = 0x70	0	1	1	1	0	0	0	0

SETTING A TOS VALUE ON PACKETS THAT ARE BORROWING

Packets that are borrowing bandwidth can be marked differently from traffic running at or under the limit established for their class.

As an example of this, we can add to the definition established for the traffic-class Angel. For this class, we can set a tos-borrow-mask of 0x80 for over-limit packets destined for the specified subnet, and set bit 7 to 1. As a result, packets belonging to this class are marked differently according to whether they are within or below the bandwidth allocation established for the class (0.5 Mbps).

```
> config cbq.1 traffic-class.angel dest-ip-addresses 1.2.3.4-1.2.3.255 \
  tos-value 10 tos-mask 70 tos-borrow-value 80 tos-borrow-mask 80\
  bandwidth-allocation 500000 parent root-output-tree
```

ToS bits set by this command are shown below.

ToS Parameter Values (write)	Bits Set in ToS Byte							
Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
tos-borrow-value = 0x80	1	0	0	0	0	0	0	0
tos-borrow-mask = 0x80	1	0	0	0	0	0	0	0

8

CONFIGURING CBQ AS A FIREWALL

This chapter explains how to use CBQ to configure a firewall on Access Point interfaces. It contains the following sections.

About the CBQ Firewall *340*

Sample Firewall Configurations *345*

Monitoring Stateful Connections *361*

Configuring Connection Accounting *365*

Specialized Firewall Capabilities *366*

This chapter explains how to use CBQ to create and manage a robust set of security policies in representative access networks. Its examples rely directly on the stateful classification capability of CBQ. Before reading this chapter, see Chapter 7 for general information on configuring traffic classes, and specific instruction on configuring stateful classification.

ABOUT THE CBQ FIREWALL

In this chapter, the word firewall is often used to refer to CBQ's filtering capabilities. CBQ is designed to function as a flexible general-purpose firewall for IP-based applications. You can also configure CBQ to protect against specific attacks, such as those that incorporate spoofing or use ICMP fragments.

Bandwidth management and firewalling are complementary aspects of CBQ. When CBQ is used as a filtering mechanism, we speak of CBQ as *firewalling* traffic or as functioning as a firewall with respect to certain classes of traffic. In practice, most CBQ trees embody a mix of bandwidth management and firewalling capabilities.

STATEFUL CLASSIFICATION AND THE CBQ FIREWALL

As explained in Chapter 7, stateful classification is a core capability of CBQ, and is also the foundation of the AP's firewall capabilities. Statefully classifying services allows the firewall to maintain state information on data flows and counterflows through the firewall. This enables CBQ (in its role of firewall) to make careful distinctions about the data flows (and counterflows) to allow, and those to filter.

IMPORTANT!

The examples contained in this chapter assume that you understand how to configure stateful traffic classes. This area is covered in detail in Chapter 7. It is important to read this material before attempting to configure CBQ as a firewall.

MANAGING DATA FLOWS AND COUNTERFLOWS: HOW IT WORKS

To take a simple example, suppose you want to allow users external to your site to access a specific Web server on your network. While allowing this access, you need to prevent data from flowing out of your network on any other connections.

You can accomplish these goals by configuring a single stateful application pair for http. Each stateful application pair consists of two classes: one for an initiated flow, and one for a counterflow of data. In the discussions that follow, we refer to these classes as *initiated* and *established* classes.

USING INITIATED AND ESTABLISHED CLASSES

In this example, the initiated flow comes from outside your network. You would therefore configure a CBQ class on the input side of your WAN interface, as follows:

```
> config cbq.1 traffic-class.web-inbound dest-ip-addresses 10.2.3.4 \
  applications http parent root-input-tree bandwidth-allocation 25000
```

This class allows HTTP traffic to a single destination within your site — host 10.2.3.4. Moreover, the class allocates 25 Kbps of the link's total bandwidth to this class.

However, class *web-inbound* is only one part of the stateful application pair CBQ needs to secure access to the Web server. The other class is configured on the input tree of the LAN side of the network. Called *web-return*, this class allows data to flow back to external hosts who have connected using web-inbound. Class web-return provides 100 Kbps of bandwidth for return flows. Its definition follows:

```
> config cbq.2 traffic-class.web-return src-ip-addresses 10.2.3.4 applica-
  tions httpEstablished parent root-input-tree bandwidth-allocation 100000
```

The presence of these two classes works like two doors on either side of the AP. You can think of connections through these doors as a narrow corridor opened by the firewall.

While the above two classes allow inbound connections to a specific server, this stateful pair allows outbound connections by all internal users on the 10.0.0.0 network to sites on the Internet.

```
> config cbq.2 traffic-class.web-to-external-sites src-ip-addresses 10.0.0.0-
  10.0.0.255 applications http parent root-input-tree bandwidth allocation
  25000

> config cbq.1 traffic-class.web-to-external-sites-return src-ip-addresses
  10.0.0.0-10.0.0.255 applications httpEstablished parent root-input-tree
  bandwidth allocation 100000
```

The use of a stateful application pair for LAN users browsing external sites limits return flows to those returning on an initiated flow. The combination of these two stateful classification pairs accomplishes these goals:

- External users are permitted to access Web server 10.2.3.4, and receive return data; attempted accesses to other systems are dropped.

- Internal users are permitted to access Web servers on the Internet; return HTTP flows on these accesses are permitted. All other traffic from the Internet to internal users is dropped.

Only packets that match the profile of the initiated class (Web traffic to a specific server) are allowed through the corridor; only return flows on this initiated class are allowed through in the opposite direction. As explained in Chapter 7, the classes work together to support a specific bidirectional flow of data.

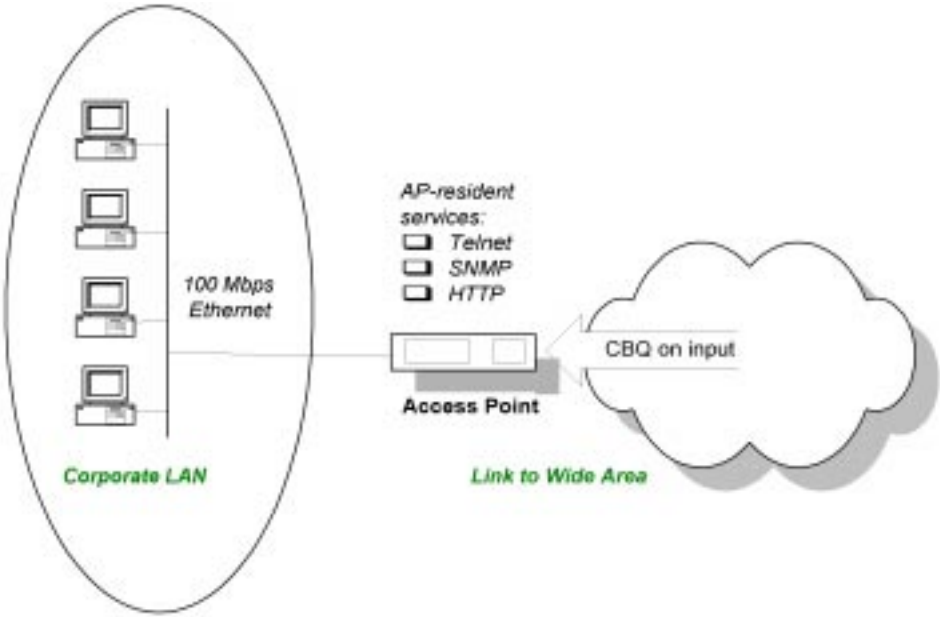
EXAMPLES: QOS PLUS SECURITY

You can configure CBQ to combine firewall and bandwidth-management capabilities in the same CBQ tree. CBQ's firewalling capabilities enable you to deny all but specifically allowed types of traffic, and also to create filtering classes that match against (and drop) unwanted traffic types. You can use these firewalling capabilities to assure that:

- ☑ *AP-resident services (like Telnet, SNMP and HTTP) are never exposed to unwanted and potentially malicious packets.*
- ☑ *Unwanted packets are not presented to the IP forwarder of any other AP interfaces.*

Protecting the AP's resident services and internal (LAN) networks is a vital security consideration when handling traffic received from the WAN as shown in Figure 42.

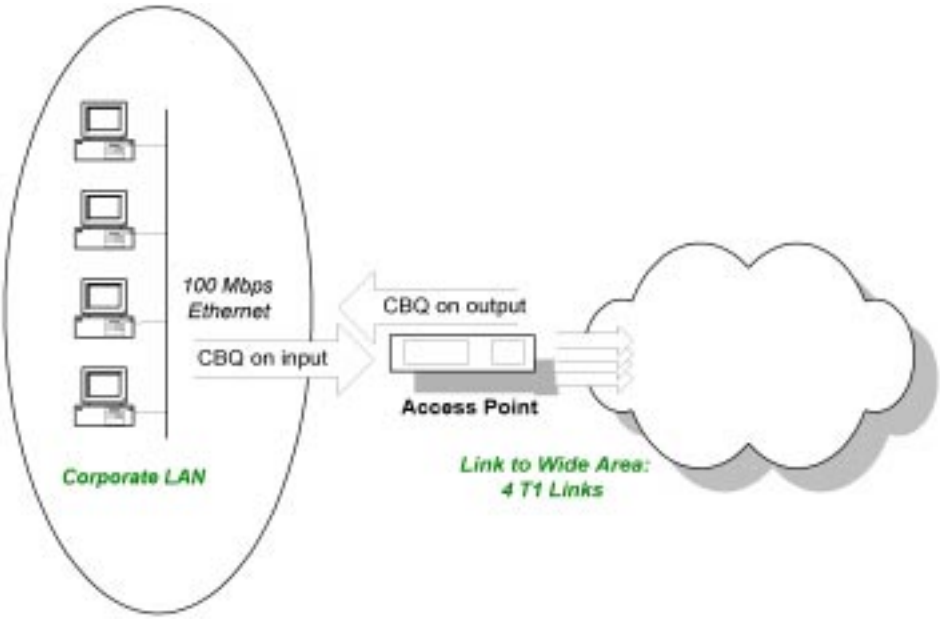
Figure 42 CBQ on Input: Single WAN to LAN Scenario



In this scenario, CBQ is implemented on the root-input-tree of the WAN interface to protect the AP's resident services, and also to allow selective access for traffic originating on the public network. Traffic shaping can be applied within traffic classes that control traffic entering the LAN from the public network. Traffic not specifically allowed on the LAN are dropped by CBQ on its external (WAN) interface.

While the security measures outlined for Figure 42 are designed to protect a LAN (and the AP) from external traffic, firewalling is also important in LAN to WAN scenarios. A typical case is shown in Figure 43. This figure shows a LAN to multiple WAN interface configuration. In this case, you can configure CBQ to perform both firewall and QoS functions on the LAN interface for traffic bound for the public network.

Figure 43 CBQ on Input: LAN to Multiple WAN Scenario



In Figure 43, CBQ is implemented in a stateful scenario. CBQ is used *on input* to classify traffic passing through the AP before the traffic is multiplexed on T1 interfaces. For return traffic, CBQ is used *on output.* Placing a single CBQ class on the root-output-tree of the LAN interface makes it possible to demultiplex the traffic returning on four WAN interfaces. In this case, using stateful classification offers distinct advantages:

- *Using CBQ on input on the LAN side makes it unnecessary to place separate CBQ trees on each outgoing interface.*
- *Using CBQ on output (again on the LAN side) allows traffic returning to the LAN to be demultiplexed using a single traffic class, rather than with a separate class on each T1 interface.*
- *Firewalling can be performed on both the input and output of the LAN interface.*

SAMPLE FIREWALL CONFIGURATIONS

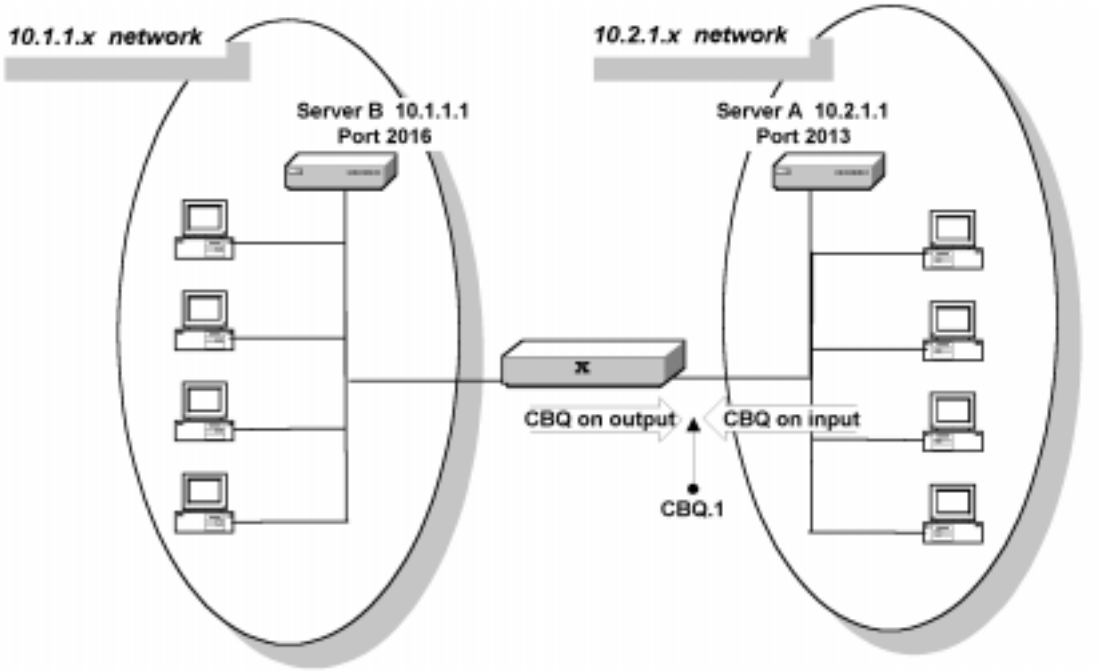
Following are three firewall configurations. Each configuration is based on a representative network scenario, and a set of access and security considerations specific to it.

- Configuration 1 - LAN to LAN Scenario
- Configuration 2 - LAN to WAN Scenario
- Configuration 3 - Providing Security for the DMZ

CONFIGURATION 1- LAN TO LAN SCENARIO

The example that follows shows how to configure CBQ on the input and output trees of the same interface. The goal is to allow users on private networks on each side of the AP to access a server on the other side. The configuration is built around the scenario shown in Figure 44.

Figure 44 Configuration 1 - LAN to LAN Scenario



The scenario shown here uses CBQ trees on the input and output side of the same interface (CBQ.1) to manage access to different servers. Users on the 10.1.1.x network are provided access to port 2013 on server A, and can receive return data from this server. Users on the 10.2.1.x network can access server B and receive return data. For enhanced security and ease-of-configuration, CBQ is configured to statefully classify all traffic exchanged between the two LANs.

This example is fairly simple. Its intent is to show how CBQ input and CBQ output trees can be combined to provide the following:

- *Access and bandwidth management for intended users*
- *A stateful firewall to prevent access by unintended users*

Users on the 10.1.1.x network need access to a specific service on host 10.2.1.1. Users on the 10.2.1.x network need access to a specific service on host 10.1.1.1. These goals can be met by configuring two CBQ trees. A CBQ-on-input tree (built below class root-input-tree) allows access to the 10.1.1.x net. Another tree, in the output direction, provides access to the 10.2.1.x net.

CBQ AS A FIREWALL

As you step through this example, notice that use of stateful classification both *enables* and *constrains* traffic flows between the two networks. The use of stateful classification provides a strict many-to-one communications channel between two groups of users and a specific server/port combination on an adjacent LAN. Attempts to pass packets not defined within this framework are dropped by CBQ, which functions as a firewall with respect to them.

In effect, CBQ enforces a set of rules that define access between users on the two networks. These rules can be expressed as follows:

- Users on network 10.1.1.x can only access a particular service (port 2013) on Server A.
- Users on network 10.2.1.x can only access a particular service (port 2016) on Server B.

All other communications attempted between hosts on these two networks are dropped, since CBQ (as configured here) provides no class that allows them to pass.

In this instance, the CBQ firewall is provided by using stateful classification of all TCP services. This assures that the only TCP traffic allowed is to a specific server and port in one direction, and that all traffic must return on an alternate return traffic flow established by CBQ. Stateful classification is also available for specific services, such as HTTP, FTP, and Telnet.

FOR ACCESS TO SERVER A FROM NET 10.1.1.X

Begin by configuring a tree to classify outbound traffic to server A. The parent class for the tree is root-output-tree. The class matches against all TCP traffic destined for port 2013 on server A (address 10.2.1.1).

```
> config cbq.1 traffic-class.out-to-server-A src-ip-addresses 10.1.1.1-
  10.1.1.255 dest-ip-addresses 10.2.1.1 dest-ports 2013 applications allTcp
  bandwidth-allocation 1000000 parent root-output-tree
```

Return traffic for this class is handled on the root-input-tree. The source port is port 2013 — the same as the destination port specified in the above command. The application chosen for return is allTcpEstablished.

Here is the definition for the return class:

```
> config cbq.1 traffic-class.return-from-server-A src-ports 2013 \
  src-ip-addresses 10.2.1.1 dest-ip-addresses 10.1.1.1-10.1.1.255 \
  dest-ports 2013 applications allTcpEstablished bandwidth-allocation
  1000000 parent root-input-tree
```

FOR ACCESS TO SERVER B FROM NET 10.2.1.X

Providing access to server B from network 10.2.1.x can be accomplished with the following commands:

```
> config cbq.1 traffic-class.out-to-server-B src-ip-addresses 10.2.1.1-
  10.2.1.255 dest-ip-addresses 10.1.1.1 dest-ports 2016 applications allTcp
  bandwidth-allocation 1000000 parent root-input-tree
```

This class provides for traffic from all users on network 10.2.1.x to the specified port on server 10.1.1.1. The class that follows provides for a return flow of information from that server (and port) to accessing users:

```
> config cbq.1 traffic-class.return-from-server-B src-ports 2016 \
  src-ip-addresses 10.1.1.1 dest-ip-addresses 10.2.1.1-10.2.1.255 \
  dest-ports 2016 applications allTcpEstablished bandwidth-allocation
  1000000 parent root-output-tree
```

CONFIGURATION 2 - LAN TO WAN SCENARIO

This example shows how to configure CBQ to implement a specific security profile. This security profile provides certain services to internal clients, and more restrictive services to external users.

Internal clients are allowed the following services:

- Telnet through firewall to external networks.
- FTP through firewall to external networks.
- HTTP and SSL access through firewall to external networks.
- SMTP mail through firewall to external networks.
- DNS - external DNS information must be made available to internal clients.

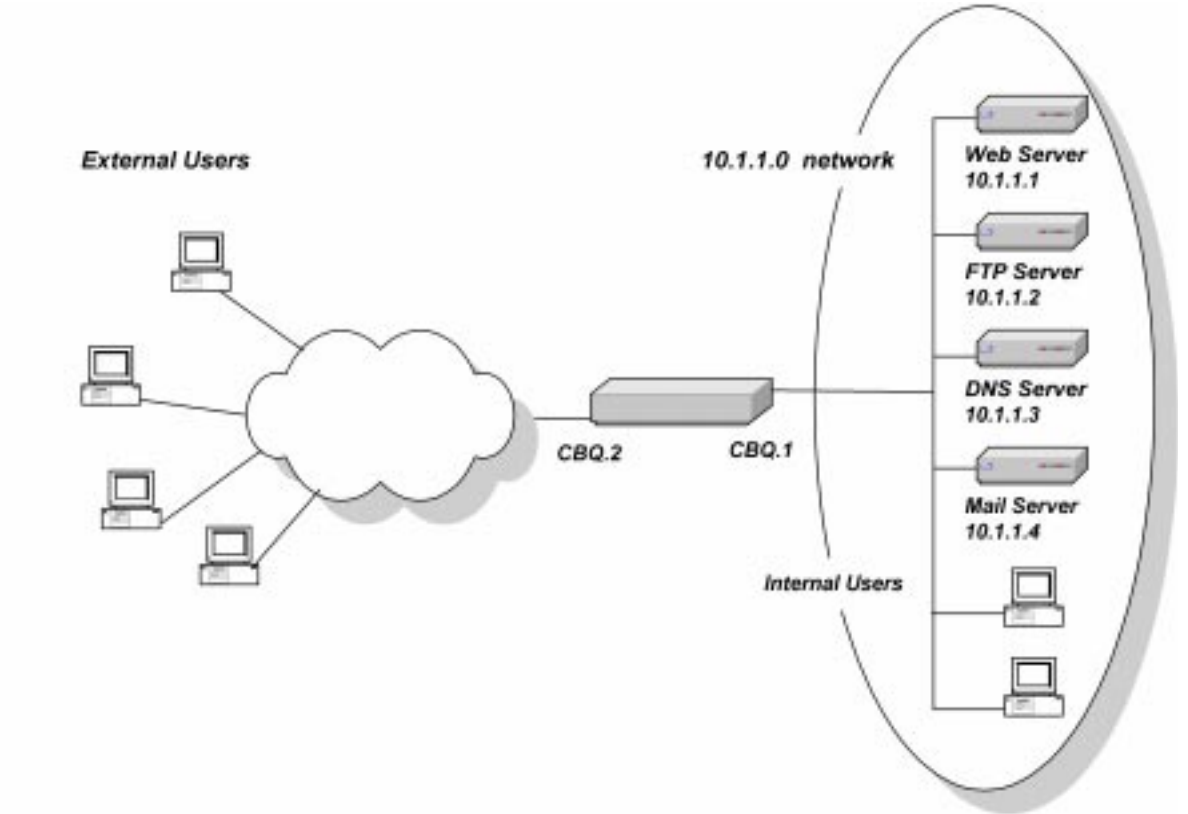
For external users, access to the internal network is limited to these services:

- FTP access to an FTP server located on the internal network.
- HTTP and SSL access to a Web server located on the internal network.

- SMTP - mail must be deliverable to clients on internal networks through an internal mail server.
- DNS - external access to an internal DNS server.

The example is based on the configuration shown in Figure 45.

Figure 45 Configuration 2 - LAN to WAN Scenario



GOALS OF THE CONFIGURATION

This configuration uses a set of stateful classes on the root-input-tree of CBQ.1 and CBQ.2. Classes placed on the CBQ.1 input tree allow access for certain services (like SNMP, Telnet, and HTTP) to internal users on the 10.1.1.0 network. Return classes for these services are placed on CBQ.2. Each return class explicitly allows data to return from external hosts, provided these data match one of the traffic flows initiated by an internal client.

As a result, the configuration ensures that only a limited set of services are

allowed onto the LAN, and only if these services match to a flow previously initiated by an internal client. This arrangement both secures the internal LAN, and provides necessary services to internal users.

This configuration assumes that root classes exist on each CBQ tree, but are inactive. After adding stateful classes as shown here, complete the configuration by activating the root class for each CBQ tree as explained on Page 279 in Chapter 7.

CLASSES FOR INTERNAL SYSTEMS

These classes provide specific services for internal systems. In addition, the classes are configured to insulate the internal network from unwanted access.

DENY SPOOFING

First, create an anti-spoofing mechanism. Spoofing is used in many types of attack, and is easy to foil. The traffic class deny-spoofers is designed to filter all accesses from the outside that appear to originate from the internal network (10.1.1.0).

```
> config cbq.2 traffic-class.deny-spoofers src-ip-addresses 10.1.1.0-
  10.255.255.255 parent root-input-tree bandwidth-allocation 0 bounded
  true
```

This filter works by identifying and dropping all spoof packets using a zero bandwidth bounded class (called deny-spoofers).

ALLOW INTERNAL USERS TO TELNET THROUGH THE AP

Configure stateful telnet classification on the internal and external interfaces to meet this goal. The class on CBQ.1 classifies Telnet traffic as it enters the AP. The class on CBQ.2 only allows Telnet that the AP recognizes as a return flow for traffic initiated from the internal network. The class ignores traffic that is not generated by an internal user.

```
> config cbq.1 traffic-class.telnet-external parent root-input-tree \
  applications telnet

> config cbq.2 traffic-class.telnet-return parent root-input-tree \
  applications telnetEstablished
```

ALLOW INTERNAL USERS TO FTP THROUGH THE AP

Provide stateful FTP classes to allow internal users to FTP to external hosts. Configuration of these classes is the same as for Telnet.

```
> config cbq.1 traffic-class.ftp-external parent root-input-tree applications
  ftp

> config cbq.2 traffic-class.ftp-return parent root-input-tree applications
  ftpEstablished
```

ALLOW WEB ACCESS TO EXTERNAL NETWORKS

Next, configure stateful HTTP classes. These classes allow HTTPSSL traffic, along with plain HTTP, so internal users will be able to access secure sites on the Web.

```
> config cbq.1 traffic-class.http-out parent root-input-tree applications http

> config cbq.2 traffic-class.http-return parent root-input-tree applications
  httpEstablished
```

ALLOW SMTP TO EXTERNAL NETWORKS

The stateful SMTP classes configured here allow internal clients to send and receive SMTP packets to outside destinations.

```
> config cbq.1 traffic-class.smtp-out parent root-input-tree applications
  smtp

> config cbq.2 traffic-class.smtp-return parent root-input-tree applications
  smtpEstablished
```

PROVIDE EXTERNAL DNS INFORMATION TO INTERNAL CLIENTS

This class makes DNS information available to internal clients. The class allows for DNS resolution (which is usually performed over UDP) and zone transfers (usually performed over TCP).

```
> config cbq.1 traffic-class.dns-external parent root-input-tree applications
  dns

> config cbq.2 traffic-class.dns-external-return parent root-input-tree \
  applications dnsEstablished
```

CLASSES FOR EXTERNAL SYSTEMS

These classes allow selective access to internal sites by systems on the Internet. The classes are designed to allow specific and limited access.

ALLOW EXTERNAL CLIENTS TO ACCESS INTERNAL FTP AND HTTP SERVERS

You can use stateful or stateless classification to allow external hosts access to Web and FTP servers (10.1.1.1 and 10.1.1.2). Here is the stateful application pair that allows external access to the Web server:

```
> config cbq.2 traffic-class.web-internal parent root-input-tree \
  dest-ip-addresses 10.1.1.1 applications http

> config cbq.1 traffic-class.web-internal-return parent root-input-tree \
  src-ip-addresses 10.1.1.1 applications httpEstablished
```

The class on CBQ.2 (web-internal) allows Web traffic from external sites to access the internal Web server. The class on CBQ.1 (web-internal-return) requires return HTTP traffic to be a counterflow to traffic initiated from the external network. Note that these classes implicitly allow HTTPSSL traffic to be passed in either direction, along with ordinary HTTP traffic.

Similarly, this stateful pair allows external systems to access the internal FTP server:

```
> config cbq.2 traffic-class.ftp-internal parent root-input-tree \
  dest-ip-addresses 10.1.1.2 applications ftp

> config cbq.1 traffic-class.ftp-return parent root-input-tree src-ip-addresses
  10.1.1.2 applications ftpEstablished
```

Note that if the FTP server in question has a domain name and the AP DNS client is configured, you could use src-domain-names rather than src-ip-addresses.

ALLOW EXTERNAL HOSTS TO ACCESS AN INTERNAL MAIL SERVER

This stateful pair allows external hosts to access the internal mail server. At the same time, the AP denies SMTP access to other internal sites.

```
> config cbq.2 traffic-class.mail-internal parent root-input-tree \
  dest-ip-addresses 10.1.1.4 applications smtp

> config cbq.1 traffic-class.mail-return parent root-input-tree \
  src-ip-addresses 10.1.1.4 applications smtpEstablished
```

ALLOW EXTERNAL HOSTS TO ACCESS AN INTERNAL DNS SERVER

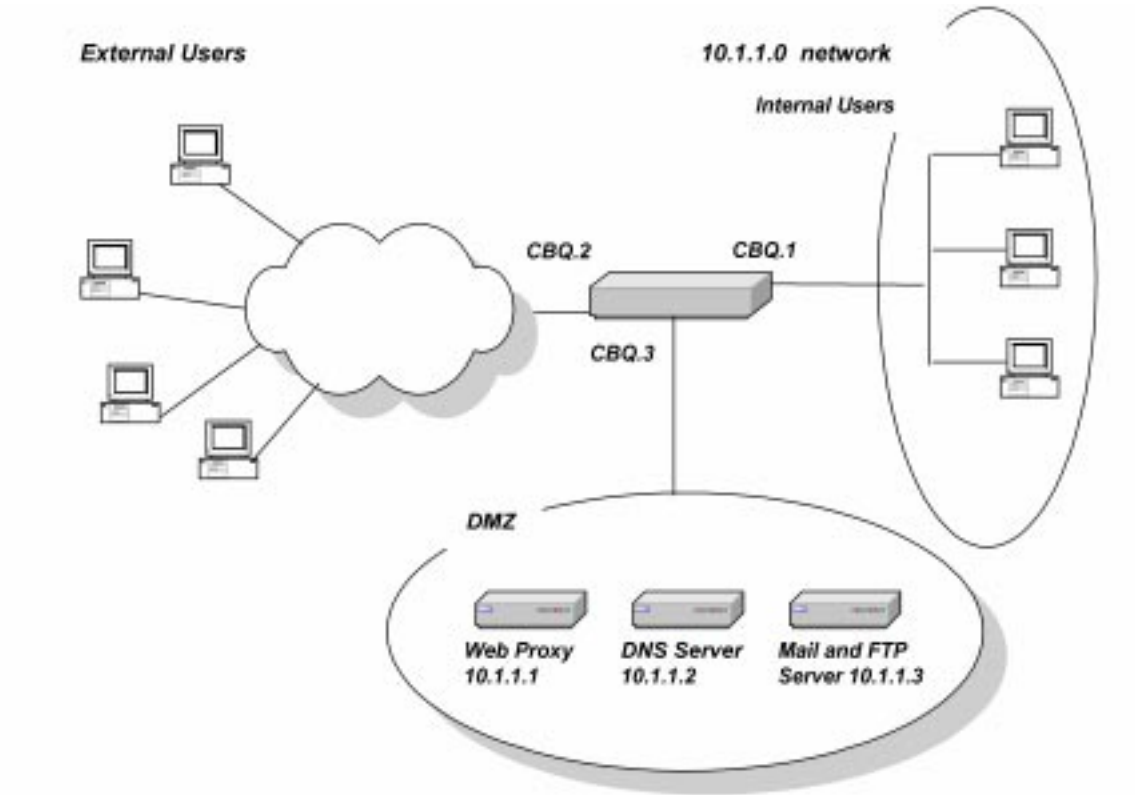
This stateful pair allows external hosts to access information on the internal DNS server.

```
> config cbq.2 traffic-class.dns-internal parent root-input-tree dest-ip-
  addresses 10.1.1.3 applications dns

> config cbq.1 traffic-class.dns-return parent root-input-tree src-ip-
  addresses 10.1.1.3 applications dnsEstablished
```

CONFIGURATION 3 - SECURING THE DMZ

This example shows how to configure a highly secure firewall for systems that are exposed to Internet traffic and situated outside of a private network. The example that follows is based on the configuration shown in Figure 46.

Figure 46 Configuration 3 - Securing Hosts on the DMZ



Often referred to as a DMZ (demilitarized zone), the perimeter network shown here exists between an organization's private network, and the public network. Providing secure, selective access to hosts on a DMZ is important, since both internal services (Web proxy services, for example) and external services (such as a corporate Web server) are often provided on a DMZ.

The configuration outlined here provides services to internal users. Users on the public network are provided restrictive access to systems on the DMZ. Since this example's purpose is to illustrate how to use CBQ to build a security framework, bandwidth has not been assigned when creating the sample classes that follow.

This configuration provides secure services to internal and external users:

Services Provided to Internal Users

- Telnet through firewall to a bastion host on the DMZ.
- FTP through firewall to anonymous FTP proxy server on the DMZ.
- HTTP through firewall to an HTTP proxy server on the DMZ.
- SSL through the firewall to an HTTP proxy server on the DMZ.
- SMTP mail through firewall to external networks.
- DNS using a domain name server on the DMZ.

Services Provided to External Users

- FTP access to an anonymous FTP server located on the DMZ.
- HTTP access to a Web server located on the DMZ.
- SSL access to a Web server located on the DMZ.
- SMTP access to a mail server located on the DMZ.
- DNS using a name server on the DMZ.

MAKE ROOT-INPUT-TREES ACTIVE

By default, a root-input-tree and root-output-tree (each with a 100 Mbps) is created on each CBQ instance. The classes created for this example live beneath the root-input-trees of CBQ.1, CBQ.2, and CBQ.3. The root-input-tree for each CBQ instance is the parent class for the classes created beneath it. However, you must activate (that is, turn on) this parent class, otherwise its child classes will not become active.

Activate the root-input-tree of each instance with these commands:

```
> config cbq.1 traffic-class.root-input-tree row-status active

> config cbq.2 traffic-class.root-input-tree row-status active

> config cbq.3 traffic-class.root-input-tree row-status active
```

After activating these root classes, you can add classes as shown below. As mentioned earlier, specific bandwidth allocations have not been provided in the sample classes that follow.

ADD TRAFFIC CLASSES FOR INTERNAL SYSTEMS

These classes provide specific services for internal systems. In addition, the classes are configured to secure the internal network from unwanted access.

ALLOW USERS TO TELNET TO THE WEB PROXY ON THE DMZ

Internal users can telnet to 10.1.1.1 and from this host to hosts on the Internet. Similarly, internal users can surf external locations using an HTTP proxy on this host.

```
> config cbq.1 traffic-class.telnet-to-bastion-host-on-dmz dest-addresses
  10.1.1.1 parent root-input-tree applications telnet

> config cbq.3 traffic-class.telnet-to-bastion-host-on-dmz-return src-
  addresses 10.1.1.1 parent root-input-tree applications telnetEstablished
```

ALLOW HTTP, HTTPSSL, AND FTP ACCESS TO SYSTEMS ON THE DMZ

The stateful HTTP classes shown here allow both HTTP and HTTPSSL access to the Web proxy server.

```
> config cbq.1 traffic-class.http-and-httpssl-to-proxy-on-dmz \
  dest-addresses 10.1.1.1 parent root-input-tree applications http

> config cbq.3 traffic-class.http-and-httpssl-to-proxy-on-dmz-return \
```

```
src-addresses 10.1.1.1 parent root-input-tree applications httpEstablished
```

This stateful pair allows access to the anonymous FTP server.

```
> config cbq.1 traffic-class.ftp-to-anon-ftp-server-on-dmz dest-addresses
  10.1.1.3 parent root-input-tree applications ftp

> config cbq.3 traffic-class.ftp-to-anon-ftp-server-on-dmz-return \
  src-addresses 10.1.1.3 parent root-input-tree applications ftpEstablished
```

ALLOW SMTP MAIL TO THE DMZ MAIL SERVER

Internal systems can send and receive mail from the mail server.

```
> config cbq.1 traffic-class.smtp-to-mail-server-on-dmz dest-addresses
  10.1.1.3 parent root-input-tree applications smtp

> config cbq.3 traffic-class.smtp-to-mail-server-on-dmz-return src-
  addresses 10.1.1.3 parent root-input-tree applications smtpEstablished
```

ALLOW ACCESS TO THE DNS SERVER

These stateful classes enable internal systems to access the DNS server for name resolution.

```
> config cbq.1 traffic-class.dns-to-domain-server-on-dmz dest-addresses
  10.1.1.2 parent root-input-tree applications dns

> config cbq.3 traffic-class.dns-to-domain-server-on-dmz-return src-
  addresses 10.1.1.2 parent root-input-tree applications dnsEstablished
```

CREATE A DEFAULT DENY CLASS FOR THE LAN

The purpose of this class is to deny (drop) all services other than those specifically allowed. The AP views classes ending with the suffix *-default* as default classes. These classes match all packets that do not match to any other class.

```
> config cbq.1 traffic-class.lan-deny-default bandwidth-allocation 0
  bounded true
```

In this case, a specific default class is not necessary. CBQ automatically drops all packets that do not match any traffic class. However, creating a specific default class has the advantage of allowing you to use counters to see how many bytes are dropped by this class. See Chapter 7 for detailed information on using default classes.

ADD CLASSES FOR EXTERNAL SYSTEMS

DENY SPOOFERS FROM EXTERNAL NET

This class denies packets designed to appear as if they come from internal systems. A single traffic class makes it easy to defend against this type of intrusion.

```
> config cbq.2 traffic-class.deny-spoofers-on-external-net src-addresses
  10.1.1.0-10.255.255.255 bandwidth-allocation 0 bounded true
```

ALLOW HTTP, HTTPSSL, AND FTP ACCESS TO SYSTEMS ON THE DMZ

Allow external systems to access the corporate Web server and the anonymous FTP server on the DMZ.

```
> config cbq.2 traffic-class.http-and-httpssl-to-corp-web-server \
  dest-addresses 10.1.1.1 parent root-input-tree applications http

> config cbq.3 traffic-class.http-and-httpssl-to-corp-web-server-return \
  src-addresses 10.1.1.1 parent root-input-tree applications httpEstablished
```

This stateful pair allows external systems to access the anonymous FTP server.

```
> config cbq.2 traffic-class.ftp-to-bastion-host-on-dmz dest-addresses
  10.1.1.3 parent root-input-tree applications ftp

> config cbq.3 traffic-class.ftp-to-bastion-host-on-dmz-return src-addresses
  10.1.1.3 parent root-input-tree applications ftpEstablished
```

ALLOW HTTP AND HTTPSSL FROM THE DMZ TO THE WAN

This stateful pair allows users on the Web proxy server to access sites on the WAN.

```
> config cbq.3 traffic-class.http-and-httpssl-from-proxy-to-wan \
  src-addresses 10.1.1.1 parent root-input-tree applications http

> config cbq.2 traffic-class.http-and-httpssl-from-proxy-to-wan-return dest-
  addresses 10.1.1.1 parent root-input-tree applications httpEstablished
```

ALLOW THE DMZ MAIL SERVER TO ACCESS EXTERNAL MAIL SERVERS

This stateful pair allows the DMZ mail server to contact and send mail to mail servers on external networks.

```
> config cbq.3 traffic-class.smtp-from-mail-server-to-wan src-addresses
  10.1.1.3 parent root-input-tree applications smtp

> config cbq.2 traffic-class.smtp-from-mail-server-to-wan-return src-
  addresses 10.1.1.3 parent root-input-tree applications smtpEstablished
```

ALLOW EXTERNAL MAIL SERVERS TO ACCESS THE DMZ MAIL SERVER

This stateful pair allows external mail servers to send mail to the mail server on the DMZ.

```
> config cbq.2 traffic-class.smtp-to-bastion-host-on-dmz dest-addresses
  10.1.1.3 parent root-input-tree applications smtp

> config cbq.3 traffic-class.smtp-to-bastion-host-on-dmz-return src-
  addresses 10.1.1.3 parent root-input-tree applications smtpEstablished
```

ALLOW USERS FROM THE DMZ TO FTP TO THE INTERNET

```
> config cbq.3 traffic-class.ftp-from-anon-ftp-server-to-wan src-addresses
  10.1.1.3 parent root-input-tree applications ftp

> config cbq.2 traffic-class.ftp-from-anon-ftp-server-to-wan-return dest-
  addresses 10.1.1.3 parent root-input-tree applications ftpEstablished
```

ALLOW ACCESS TO THE NAME SERVER ON THE DMZ

Several stateful application pairs are required to allow secure access to the bastion DNS server.

The pair shown here allows the DNS server to query other DNS servers on the public network. This class is necessary, as it enables the DNS server to resolve IP address information by contacting other well-known servers.

```
> config cbq.3 traffic-class.dns-from-domain-server-to-wan src-addresses
  10.1.1.2 parent root-input-tree applications dns

> config cbq.2 traffic-class.dns-from-domain-server-to-wan-return dest-
  addresses 10.1.1.2 parent root-input-tree applications dnsEstablished
```

In contrast, this stateful pair allows external DNS servers to access the server on the DMZ.

```
> config cbq.2 traffic-class.dns-to-bastion-host-on-dmz dest-addresses
  10.1.1.2 parent root-input-tree applications dns

> config cbq.3 traffic-class.dns-to-bastion-host-on-dmz-return \
  src-addresses 10.1.1.2 parent root-input-tree applications dnsEstablished
```

ALLOW TELNET TO THE WAN

This stateful pair allows internal users to telnet from host 10.1.1.1 to the WAN.

```
> config cbq.3 traffic-class.telnet-from-bastion-to-wan src-addresses
  10.1.1.1 parent root-input-tree applications telnet

> config cbq.2 traffic-class.telnet-from-bastion-host-to-wan-return dest-
  addresses 10.1.1.1 parent root-input-tree applications telnetEstablished
```

CREATE A DEFAULT DENY CLASS FOR THE WAN

The purpose of this class is to deny (drop) all services other than those specifically allowed. The AP views classes ending with the suffix *-default* as default classes. These classes match all packets that do not match to any other class.

```
> config cbq.2 traffic-class.wan-deny-default bandwidth-allocation 0
  bounded true parent root-input-tree
```

CREATE A DEFAULT DENY CLASS FOR THE DMZ

This class drops all traffic exiting the DMZ that does not match any configured traffic class.

```
> config cbq.3 traffic-class.dmz-deny-default bandwidth-allocation 0
  bounded true parent root-input-tree
```

Figure 47 Traffic Class Tree: DMZ Example

CBQ.* TRAFFIC-CLASS-TREE

INTERFACE	ROW	Class Hierarchy	
cbq.1	1	root-input-tree	Rate=1000000000
cbq.1	2	telnet-to-bastion-hot-on-dmz	Rate= 0
cbq.1	3	http-and-httpssl-to-proxy-on-dmz	Rate= 0
cbq.1	4	ftp-to-anon-ftp-server-on-dmz	Rate= 0
cbq.1	5	smtp-to-mail-server-on-dmz	Rate= 0
cbq.1	6	dns-to-domain-server-on-dmz	Rate= 0
cbq.1	7	lan-deny-default	Filter Class
cbq.2	1	root-input-tree	Rate=1000000000
cbq.2	2	http-and-httpssl-to-corp-web-server	Rate= 0
cbq.2	3	ftp-to-bastion-host-on-dmz	Rate= 0
cbq.2	4	ftp-from-anon-ftp-server-to-wan-retu>	Rate= 0
cbq.2	5	dns-from-domain-server-to-wan-return	Rate= 0
cbq.2	6	dns-to-bastion-host-on-dmz	Rate= 0
cbq.2	7	deny-spoofers-on-external-net	Rate= 0
cbq.2	8	telnet-from-bastion-to-wan-return	Rate= 0
cbq.2	9	http-and-httpssl-from-proxy-to-wan-r>	Rate= 0
cbq.2	10	smtp-from-mail-server-to-wan-return	Rate= 0
cbq.2	11	smtp-to-bastion-host-on-dmz	Rate= 0
cbq.2	12	wan-deny-default	Filter Class
cbq.3	1	root-input-tree	Rate=1000000000
cbq.3	2	telnet-to-bastion-host-on-dmz-return	Rate= 0
cbq.3	3	ftp-to-anon-ftp-server-on-dmz-return	Rate= 0
cbq.3	4	smtp-to-mail-server-on-dmz-return	Rate= 0
cbq.3	5	dns-to-domain-server-on-dmz-return	Rate= 0
cbq.3	6	http-and-httpssl-to-corp-web-server->	Rate= 0
cbq.3	7	ftp-to-bastion-host-on-dmz-return	Rate= 0
cbq.3	8	smtp-from-mail-server-to-wan-return	Rate= 0
cbq.3	9	ftp-from-anon-ftp-server-to-wan	Rate= 0
cbq.3	10	dns-from-domain-server-to-wan	Rate= 0
cbq.3	11	dns-to-bastion-host-on-dmz-return	Rate= 0
cbq.3	12	telnet-from-bastion-to-wan	Rate= 0
cbq.3	13	http-and-httpssl-from-proxy-to-wan	Rate= 0
cbq.3	14	smtp-to-bastion-host-on-dmz-return	Rate= 0
cbq.3	15	http-and-httpssl-to-proxy-on-dmz-ret>	Rate= 0
cbq.3	16	dmz-deny-default	Rate= 0

MONITORING STATEFUL CONNECTIONS

The AP implements connection monitoring to provide visibility into the firewall. You can view all connections going through the AP, or summary information on connections going through a particular CBQ class.

VIEWING INFORMATION ON STATEFUL CONNECTIONS

Connection monitoring is only available for statefully configured applications. As discussed previously, this requires use of a stateful application pair (such as tcp and tcpEstablished or http and httpEstablished) to create two classes on separate AP interfaces. You can use stateful application pairs for all TCP, all UDP or all ICMP connections, or for specific protocols like HTTP. To make monitoring possible for UDP and ICMP-based connections, the AP models them as connection-oriented protocols.

You can view connections by flow or by time.

ABOUT CONNECTION STATE INFORMATION

Information contained in the CBQ Connections tables (sorted by flow or by time) is self-explanatory. However, two areas deserve further comment. The system time in the table begins in 1970 (1970/0/0 hh:mm:sec) if a system date and time have not been assigned. See "Configuring the System Name, Date and Time" on Page 46 for instructions on setting the AP date and time.

Each connection captured in the table has a state associated with it. Supported states are *Established*, *Connecting*, *Disconnecting*, *OneWay* and *TwoWay*. Of these, *OneWay* and *TwoWay* are reserved for ICMP and UDP, which the AP models as connection-oriented protocols. A state of *OneWay* signifies that a connection has been initiated; a state of *TwoWay* indicates that a response to this connection attempt has been received.

For TCP-based connections, possible states are *Established*, *Connecting* and *Disconnecting*. *Established* means that a three-way handshake has been completed, and that bidirectional flow of data has been established between the AP and another system. *Connecting* signifies that a connection attempt has been made, either by the AP or by a system attempting to connect to it. *Disconnecting* signifies that a TCP FIN or Reset has been sent, either by the AP or by the system with which it had been communicating.

VIEWING CONNECTION INFORMATION BY FLOW

To view connections by flow, use this command:

```
> show cbq connections-by-flow
```

```
                                CBQ CONNECTIONS-BY-FLOW

    SRC IP ADDR|    DEST IP ADDR| PROT| SRC PORT| DEST PORT| APPLICATION|        Connect Time|         Elapsed Time|
               |                |     |         |          |            |              State|          Src Packets|
               |                |     |         |          |            |          Src Octets|  Src Retrans Packets|
               |                |     |         |          |            |  Src Retrans Octets|         Dest Packets|
               |                |     |         |          |            |         Dest Octets| Dest Retrans Packets|
               |                |     |         |          |            | Dest Retrans Octets|      Kill Connection|
---------------|----------------|-----|---------|----------|------------|--------------------|---------------------|
198.202.232.58|    216.33.87.17|  tcp|     4205|      http|         tcp|1999/12/15 08:13:45>|         00:00:48.577|
               |                |     |         |          |            |       Disconnecting|                   10|
               |                |     |         |          |            |                 803|                    0|
               |                |     |         |          |            |                   0|                   13|
               |                |     |         |          |            |               12355|                    2|
               |                |     |         |          |            |                3000|                false|
198.202.232.58|    216.33.87.17|  tcp|     4207|      http|         tcp|1999/12/15 08:13:45>|         00:00:48.553|
               |                |     |         |          |            |       Disconnecting|                   23|
               |                |     |         |          |            |                3481|                    0|
               |                |     |         |          |            |                   0|                   23|
               |                |     |         |          |            |               24453|                    3|
               |                |     |         |          |            |                4500|                false|
```

Viewing connections by flow provides complete information on all current connections through the firewall. However, the multi-column format requires you to set your host's column width to 100 or greater. You can set column width to 100 with this command:

```
> colwidth 100
```

SORTING CONNECTION INFORMATION BY TIME

You can also sort connection information by time using this command:

```
> show cbq connections-by-time
```

A sample display of information sorted by time is shown here:

```
                                CBQ CONNECTIONS-BY-TIME

INDEX|          Src Ip Addr|         Dest Ip Addr|                 Prot|
     |             Src Port|            Dest Port|          Application|
     |         Connect Time|         Elapsed Time|        Src Class Name|
     |      Dest Class Name|                State|           Src Packets|
     |           Src Octets| Src Retrans Packets|    Src Retrans Octets|
     |         Dest Packets|          Dest Octets| Dest Retrans Packets|
     |  Dest Retrans Octets|      Kill Connection|                      |
-----|---------------------|---------------------|----------------------|
   9|       198.202.232.87|         209.3.82.85|                   tcp|
     |                1976|                 http|                   tcp|
     |1999/12/15 09:20:17>|         00:31:33.495|              cbq.3:tcp|
     |          cbq.3:tcp-i|          Established|                  1173|
     |                47256|                    0|                    0|
     |                 1172|               318579|                    0|
     |                    0|                false|                      |
 462|       198.202.232.38|        205.188.5.178|                   tcp|
     |                3065|                 5190|                   tcp|
     |1999/12/15 09:29:31>|         00:22:19.781|              cbq.3:tcp|
     |          cbq.3:tcp-i|          Established|                    16|
     |                1721|                    0|                    0|
     |                  14|                 2639|                    2|
     |                  80|                false|                      |
```

When you sort by time, information is displayed from earliest to latest flow. For instance, in the display shown here, the connect time of the first entry is 9:20:31 am, and that of the second is 9:29:31.

USING THE WHERE COMMAND TO SORT CONNECTION INFORMATION

When used in combination with connection-by-time information, the *where* command provides you with a way to zero in on specific connections. For instance, you may want information only on connections associated with a particular source address. You can get this information as follows:

```
> show cbq connection-by-time.* summary where src-up-addr == <ip
  address>
```

Here is a sample of the output from this command:

```
CBQ CONNECTION-BY-TIME.* SUMMARY WHERE SRC-IP-ADDR == 198.202.232.70

INDEX|      Src Ip Addr|         Dest Ip Addr|                Prot|         Src Port|
     |        Dest Port|          Application|        Connect Time|     Elapsed Time|
     |   Src Class Name|      Dest Class Name|               State|      Src Packets|
     |       Src Octets|  Src Retrans Packets|  Src Retrans Octets|     Dest Packets|
     |      Dest Octets| Dest Retrans Packets| Dest Retrans Octets|  Kill Connection|
------|-----------------|---------------------|--------------------|-----------------|
   566|   198.202.232.70|         205.188.4.243|                 tcp|             1072|
     |             5190|                  tcp|1999/12/15 14:40:36>|  01:02:54.763|
     |        cbq.3:tcp|          cbq.3:tcp-i|         Established|               32|
     |             2482|                    0|                   0|               30|
     |             3373|                   10|                 400|            false|
```

As another example, you may want to get information on connections with a duration of 1 hour or more. In this case, use the *where* command as follows:

```
> show cbq connection-by-time.* summary where elapsed-time >=
  01:00:00
```

Here is a sample of the output from this command:

```
CBQ CONNECTION-BY-TIME.* SUMMARY WHERE ELAPSED-TIME >= 01:00:00

INDEX|      Src Ip Addr|         Dest Ip Addr|                Prot|         Src Port|
     |        Dest Port|          Application|        Connect Time|     Elapsed Time|
     |   Src Class Name|      Dest Class Name|               State|      Src Packets|
     |       Src Octets|  Src Retrans Packets|  Src Retrans Octets|     Dest Packets|
     |      Dest Octets| Dest Retrans Packets| Dest Retrans Octets|  Kill Connection|
------|-----------------|---------------------|--------------------|-----------------|
   188|   198.202.232.87|          209.3.82.85|                 tcp|             3806|
     |             http|                  tcp|1999/12/15 14:37:13>|  01:03:38.352|
     |        cbq.3:tcp|          cbq.3:tcp-i|         Established|             2534|
     |           101696|                    0|                   0|             2558|
     |           921588|                    0|                   0|            false|
   217|  198.202.232.193|         207.251.73.28|                 tcp|            56290|
     |             http|                  tcp|1999/12/15 14:37:35>|  01:03:16.988|
     |        cbq.3:tcp|          cbq.3:tcp-i|         Established|             2529|
     |           101576|                    0|                   0|             2542|
     |           870282|                    0|                   0|            false|
   222|   198.202.232.34|         205.188.5.50|                 tcp|             3875|
     |             5190|                  tcp|1999/12/15 14:37:52>|  01:03:00.355|
     |        cbq.3:tcp|          cbq.3:tcp-i|         Established|               71|
     |             6607|                    0|                   0|               66|
     |            10889|                    4|                 160|            false|
```

VIEWING SUMMARY INFORMATION ABOUT CONNECTIONS

Use this command to view summary information on connections.

```
> show cbq connection-summary
```

```
                                CBQ CONNECTION-SUMMARY

                      Current Connected  = 0
                    Current Unconnected  = 1
                      Total Connected    = 0
                  Total Connect Failures = 1182
          Total Killed Due To Inactivity = 0
                 Total Killed By Management = 0
    Total Killed Due To Reclassification = 0
             Total Out Of Window Rejects = 0
               Total No Resource Rejects = 0
               Total Syn Protect Timeouts = 0
                  Syn Protect Timeout= = 30 seconds
```

CONFIGURING CONNECTION ACCOUNTING

You can configure the AP to capture detailed connection information for stateful classes. To do this, you must enable connection accounting on a class-by-class basis. For instance, this command enabled connection accounting for the class finance-in:

```
> config cbq.1 traffic-class.finance-in connection-accounting enabled
```

After enabling accounting for a class, you can view detailed accounting information for it using this command:

```
> show cbq.1 traffic-class.finance-in class-connections summary
```

```
                CBQ.1 TRAFFIC-CLASS.FINANCE-IN CLASS-CONNECTIONS

  Current Unconnected  = 0     Total Killed Due To Reclassification  = 0
    Current Connected  = 0               Total Out Of Window Rejects  = 0
    Total Connections  = 0                 Total No Resource Rejects  = 0
Total Connect Failures = 0 Total Killed Due To Syn Flood Protection  = 0
 Total Killed Inactive = 0                        Retransmitted Packets  = 0
  Total Killed By Mgmt = 0                        Retransmitted Octets   = 0
```

SPECIALIZED FIREWALL CAPABILITIES

In addition to providing a generalized and powerful firewalling capabilities, the AP implements specialized controls for certain types of applications and protocols. These include control over the following:

- ICMP packets
- IP Fragment classification
- SYN flood

CLASSIFYING ICMP PACKETS

ICMP is a message control and error-reporting protocol widely used for gathering information relating to IP packet processing. Protocols like ping and traceroute — both based on ICMP — are commonly used for trouble-shooting and diagnostics.

ICMP presents some security risks, partly because of its usefulness in information gathering. Some forms of attack involve use of certain types of ICMP packets. Moreover, ICMP provides a virtual window to outside parties attempting to discover host names, addresses, and the overall topology of a private network. For these reasons, the AP makes it easy to selectively allow or deny certain ICMP packet types. This allows you to set up a security profile that provides internal users with the use they would expect from the protocol, while denying hackers the information ICMP messages might otherwise provide.

STATEFULLY CLASSIFYING ICMP PACKETS

To provide stateful classification for ICMP messages, you can use the applications types allIcmp and allIcmpEstablished. Here, for example, are two classes used to provide stateful ICMP classification on the input and output trees of CBQ.1 and CBQ.2. The two classes specify a flow and a return flow, respectively. They work together in the same manner as other stateful classes supported by the AP.

```
> config cbq.1 traffic-class.stateful-icmp applications allIcmp parent \
  root-input-tree

> config cbq.2 traffic-class.stateful-icmp-return applications \
  allIcmpEstablished parent root-output-tree
```

CONTROLLING ICMP PACKETS

In addition, the classification parameter *icmp-msgs-allowed* lets you control the types of ICMP packets that are allowed in and out of a private network. You can use this parameter in combination with applications types allIcmp and allIcmpEstablished, or with protocols type ICMP. Using any of the above traffic types ensures that CBQ will identify ICMP packets. Following this initial identification, the *icmp-msgs-allowed* parameter allows CBQ to make much finer distinctions about which packets should be allowed. The completions available for this parameter allow you to specify exactly which packet types are allowed by CBQ. These include the following:

echoReply	destUnreachable	sourceQuench
redirect	echoRequest	ttlExceeded
paramProblem	timeRequest	timeReply
infoRequest	infoReply	maskRequest
maskReply		

Here is a sample traffic class configuration in which the *icmp-msgs-allowed* parameter is used to allow just three ICMP packet types: timeRequest, timeReply, and echoReply.

```
> config cbq.1 traffic-class.sample-icmp-class applications allIcmp \
  icmp-msgs-allowed timeRequest+timeReply+echoReply
```

HOW IT WORKS

By default, when you classify packets as allIcmp, allIcmpEstablished, or as protocol icmp, CBQ allows all ICMP packet types. However, if you use the *icmp-msgs-allowed* parameter to designate one or more specific packet types, these are the only ones CBQ will allow. For example, the class created with this command allows all ICMP packet types:

```
> config cbq.1 traffic-class.all-icmp-types applications allIcmp \
  parent root-input-class row-status active
```

In contrast, this class allows only packets of type infoRequest or infoReply.

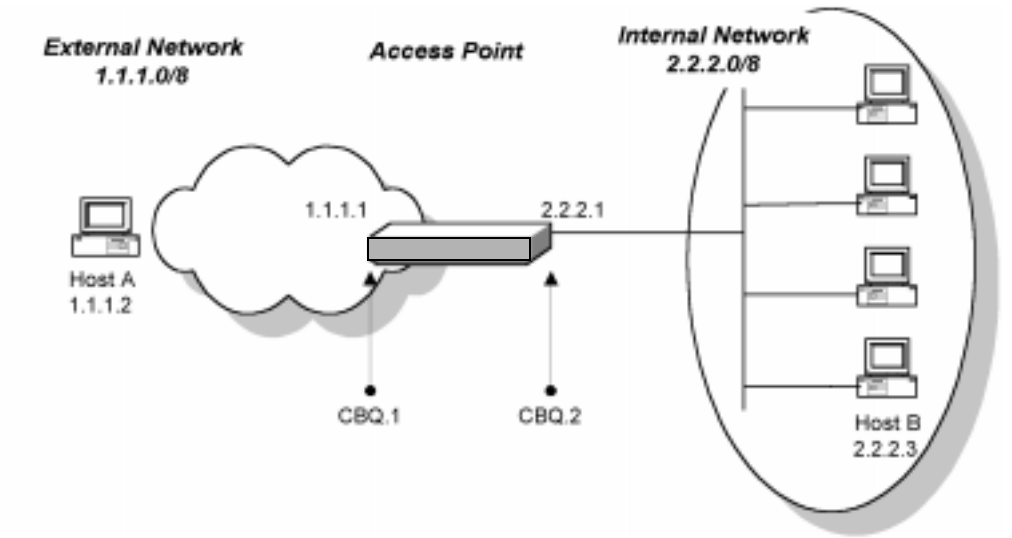
```
> config cbq.1 traffic-class.all-icmp-types applications allIcmp \
  icmp-msgs-allowed infoRequest+infoReply parent root-input-class \
  row-status active
```

Here, as in other cases, specifying particular packet types means these are the only types that CBQ will match against. Hence, these are the only ICMP packet types that will be allowed to pass through the firewall.

EXAMPLE: ICMP PACKET FILTERING

The examples that follow are based on the configuration shown in Figure 48. In this scenario, the AP sits on the edge of private network 2.2.2.0. The router's ICMP filtering capabilities are utilized to selectively allow and deny different types of ICMP packets into and out of this network.

Figure 48 Sample Configuration: ICMP Fragment Filtering



The example that follows explains how to configure the AP firewall to control ICMP traffic flows to meet these goals.

1. Allow external users to ping the AP external interface (1.1.1.1).
2. Allow external hosts to send ICMP redirects to the AP.
3. Allow internal users (such as Host B) to ping external hosts.
4. Allow internal users to receive these ICMP packets:
   - destUnreachable
   - sourceQuench
   - ttlExceeded

- paramProblem

5 Prevent internal users from receiving the following ICMP messages:

- redirect
- echoRequest
- timeRequest
- timeReply
- infoRequest
- infoReply
- maskRequest
- maskReply

ALLOWING EXTERNAL USERS TO PING THE AP EXTERNAL INTERFACE

To support goals 1 and 2, the protocol is set to ICMP and the *icmp-msgs-allowed* parameter specifies redirect and echo requests. Since the AP's replies to ping (echoReply) do not go through a CBQ interface, specifying echoReply is not necessary.

```
> config cbq.1 traffic-class.router-icmp protocols icmp icmp-msgs-allowed \
  redirect+echoRequest src-ip-addresses 0.0.0.0-255.255.255.255 \
  dest-ip-addresses 1.1.1.1,2.2.2.1 parent root-input-tree
```

Here is the flow definition for traffic class router-icmp. Note that only ICMP redirects and echo requests are allowed by CBQ.1. All other ICMP packet types are dropped.

```
> show cbq.1 traffic-class.router-icmp flow-def
```

```
                CBQ.1 TRAFFIC-CLASS.ROUTER-ICMP FLOW-DEF

      Src Ip Addresses* = 0.0.0.0-255.255.255.255
     Dest Ip Addresses* = 1.1.1.1, 2.2.2.1
             Src Ports* = ""
            Dest Ports* = ""
              Protocols* = icmp
         Applications* = ""
   Classification Tos* = ""
Classification TOS Mask* = 0x00
     Src Domain Names* = ""
    Dest Domain Names* = ""
              Operator* = and
          Icmp Filter* = redirect+echoRequest
```

ALLOWING INTERNAL USERS TO PING EXTERNAL HOSTS

To allow internal users to ping external hosts, you can configure stateful classification of ICMP. This requires two classes: one for outbound ping attempts, and one the return of information. The first class, called icmp-outbound, is placed on the root-output-tree of CBQ.1. The protocol chosen is ICMP. Since the icmp-msgs-allowed parameter is not included in this class definition, all types of ICMP are allowed by the class.

```
> config cbq.1 traffic-class icmp-outbound applications allIcmp parent
  root-output-tree bandwidth-allocation 50000 row-status createAndGo
```

The flow definition for this class is shown below.

```
> show cbq.1 traffic-class.icmp-outbound flow-def
```

```
                    CBQ.1 TRAFFIC-CLASS.ICMP-OUTBOUND FLOW-DEF

  Src Ip Addresses* = ""                  Classification Tos* = ""
 Dest Ip Addresses* = ""            Classification TOS Mask* = 0x00
         Src Ports* = ""                    Src Domain Names* = ""
        Dest Ports* = ""                   Dest Domain Names* = ""
         Protocols* = ""                            Operator* = and
    Applications* = allIcmp                     Icmp Filter* = ""
```

Called icmp-inbound, the return class is placed on the input tree of CBQ.1. It is important to note that this class uses the applications classification parameter allIcmpEstablished, rather than ICMP. The class uses the *icmp-msgs-allowed* parameter to specify echoReply. Since this is the only type of packet specified, it is the only type that will be allowed in by the class. All other types of ICMP traffic are dropped.

```
> config cbq.1 traffic-class.icmp-inbound applications allIcmpEstablished
  icmp-msgs-allowed echoReply parent root-input-tree
```

Note that this return class (icmp-inbound) could have been configured using protocol ICMP and an icmp-msgs-allowed match to echoReply. However, using stateful classification provides an extra level of security, by preventing an attacker from flooding the internal network with echoReplies. This type of nuisance attack is feasible, though not common. It is prevented by the chosen configuration, because stateful application ensures that only echoReply packets returning on the original, outbound traffic flow are allowed.

ALLOWING INBOUND USERS TO RECEIVE CERTAIN ICMP PACKETS

To allow internal users to receive certain types of ICMP packets, we can modify the flow definition of class icmp-inbound. Including only the ICMP packet types we want to allow into the network ensures that all other packets are dropped. The modified icmp-inbound class is shown below.

```
> config cbq.1 traffic-class icmp-inbound applications allIcmpEstablished \
  icmp-msgs-allowed echoReply+destUnreachable+sourceQuench+\
  ttlExceeded+paramProblem parent root-input-tree
```

BLOCKING ICMP MESSAGES FROM THE INTERNAL NETWORK

CBQ class all-systems-filter (also on the CBQ.1 root-input-tree) blocks a number of ICMP packet types, as specified in goal 5. This class blocks packets by identifying them — based on a match to any of the types specified by icmp-msgs-allowed — providing zero bandwidth, and setting bounded to true. Since zero bandwidth is provided, and none can be borrowed, none of the matched packets will get through. Here is the class definition:

```
> config cbq.1 traffic-class.all-systems-filter protocols icmp \
  icmp-msgs-allowed redirect+echoRequest +timeRequest+\
  timeReply+infoRequest+infoReply+maskRequest+maskReply \
  dest-ip-addresses 0.0.0.0-255.255.255.255 parent root-input-tree \
  bandwidth-allocation 0 bounded true
```

Locating this class on the input tree drops the unwanted packets *before* they are processed by the AP. This makes sense from a security standpoint. The flow definition for all-systems-filter is shown below.

```
> show cbq.1 traffic-class.all-systems-filter flow-def
```

```
                CBQ.1 TRAFFIC-CLASS.ALL-SYSTEMS-FILTER FLOW-DEF

      Src Ip Addresses* = ""
     Dest Ip Addresses* = 0.0.0.0-255.255.255.255
             Src Ports* = ""
            Dest Ports* = ""
             Protocols* = icmp
          Applications* = ""
    Classification Tos* = ""
Classification TOS Mask* = 0x00
      Src Domain Names* = ""
     Dest Domain Names* = ""
              Operator* = and
            Icmp Filter* = redirect+echoRequest+timeRequest+timeReply+infoReque\

                           st+infoReply+maskRequest+maskReply
```

CLASSIFYING IP FRAGMENTS

IP Fragment Classification allows you to configure the Access Point system to forward, filter, or classify IP fragments. You can use this feature to protect internal systems from certain attacks. The latter include denial of service attacks, which typically flood certain hosts with more requests than they can handle, and "ping of death" type attacks which crash systems by sending packets too large for them to reassemble.

HOW FRAGMENT CLASSIFICATION WORKS

The AP uses CBQ to classify fragments by capturing state information on the first (initial) fragment seen on an interface, and comparing information captured from subsequent and final fragments to this information.

HANDLING INITIAL FRAGMENTS

CBQ captures state information on the first fragment that appears on the interface. This information includes:

- Source and destination IP address
- IP protocol type
- The 16-bit IP identification
- The time at which the fragment was seen
- Whether a final fragment was seen
- Protocol-specific information, including TCP and UDP port numbers, if the packet is TCP or UDP, or the ICMP message identifier and sequence number

HANDLING SUBSEQUENT FRAGMENTS

CBQ captures the state information contained in subsequent fragments, and compares it to existing state information. If the source and destination IP address, IP protocol, and IP identification information does not match state information derived from the initial packet, the AP drops the fragment. However, if the fragment information is a match, the classifier is modified to contain protocol specific state information such as the TCP or UDP port numbers.

The packet is then passed to the classifier. If the time between the first fragment and any subsequent fragment exceeds the value of Inactive Timeout,

all state information is deleted and subsequent packets of the flow are dropped.

FINAL FRAGMENTS

The AP handles final fragments like subsequent fragments, except that it times out more quickly. Prior to receiving the final fragment, the time that elapses between fragments must not be greater than Inactive Timeout. Otherwise, the flow times out. Following reception of the final fragment, the AP deletes state information for the flow Last Timeout seconds later.

Setting the Last Timeout delay to something less than Inactive Timeout allows state information to be deleted more quickly than normal, while still allowing time for out of order fragments. That is, the IP packet can still be assembled when a misordered subsequent fragment is received after a final fragment. (Users can require final fragments be received in order by setting the value of Last Timeout to zero.)

Table 21 provides information on the parameters used in classifying IP fragments.

DEFENDING AGAINST DENIAL OF SERVICE ATTACKS

The AP foils denial of service attacks by allowing you to configure a limit to the size of fragments passing through CBQ. Port access attacks are handled by enhanced classification of IP fragments. All fragments with an offset of zero are treated as initial fragments. Each subsequent 0-offset packet is treated as a separate flow and classified using the packet's header information. If this packet is part of a denied service (like Telnet) it fails classification.

Another attack relies on extremely small packets that force TCP header information into separate packet fragments. This attack is designed to circumvent filtering rules in which a security device examines the first packet in a flow and allows subsequent packets to pass. The AP repels this attack by requiring a minimum length for initial fragments. For example, initial fragments containing UDP are required to be at least 8 octets in length.

CONTROLLING FRAGMENTS

The AP controls and monitors fragments on a per-interface basis using the IP Fragment Table. You can access this table using this command:

```
> show cbq.* ip-fragment-table
```

A sample table is shown below.

CBQ.* IP-FRAGMENT-TABLE

INTERFACE	Ip Fragment Disposition Ip Fragmented Pkts Ip Filtered Packets	Ip Fragment Max Size Ip Fragmented Octets Ip Filtered Octets
cbq.1	forward 0 0	16384 0 0
cbq.2	forward 0 0	16384 0 0
cbq.3	forward 0 0	16384 0 0
cbq.4	forward 0 0	16384 0 0
cbq.5	forward 0 0	16384 0 0
cbq.6	forward 0	16384 0

The IP Fragment Table contains information about IP fragments seen on each interface. The AP maintains packet and octets counters for the number of fragments seen and filtered on the interface. Fragments that are not filtered initially are forwarded, or may pass to the classifier where they are subject for further filtering.

GLOBAL IP FRAGMENT CONFIGURATION

You can configure the following limits for IP fragment handling. The values you specify apply to all CBQ instances.

To configure the maximum number of outstanding fragmented packets, use this command:

```
> config cbq ip-fragments max-outstanding
```

You can set this value between 0 and 1000.

To configure the amount of time that can elapse between fragments before

timeout, use this command:

```
> config cbq ip-fragments inactive-timeout <time in seconds>
```

You can set this value between 15 - and 3600 seconds (1 hour).

To configure the amount of time that can elapse after the last fragment before timeout, use this command:

```
> config cbq ip-fragments last-timeout <time in seconds>
```

You can set this value between 15 - and 3600 seconds (1 hour).

CONFIGURING CBQ TO DROP ALL FRAGMENTS

You may want to configure the AP to drop all IP fragments on an interface. To do this, configure fragment disposition to filter, as follows:

```
> config cbq.2 ip-fragment-disposition filter
```

Setting this parameter to filter causes CBQ to drop all fragments on the interface, including initial, subsequent, and final fragments.

CONFIGURING CBQ TO FORWARD OR CLASSIFY FRAGMENTS

You can configure the AP to forward or classify fragments, subject to a maximum size. Fragment forwarding is the default mode of behavior. Use this command to specify what to do with IP fragments.

```
> config cbq.2 ip-fragment-disposition <forward|filter|classify>
```

To prevent "ping of death" type attacks, you can set a maximum allowed size for a PDU. PDUs meeting or exceeding this size are dropped by CBQ. To configure a maximum PDU size, use this command:

```
> config cbq.1 ip-fragment-max-size <0-16384>
```

The default value for ip-fragment-max-size is 16384. See “How Fragment Classification Works” on Page 372 for detailed information on fragment classification.

To show information on IP fragments, use this command:

> show cbq ip-fragments

```
                                CBQ IP-FRAGMENTS

                  Inactive Timeout* = 30 seconds
                     Last Timeout* = 15 seconds
                    Max Outstanding* = 100
               Current Outstanding  = 0
                  Most Outstanding  = 0
          Total No Buckets Available  = 0
       Total Aged Due To Inactivity  = 0
  Total Aged Due To Final Timeout  = 0
```

Table 21 Parameters: IP Fragment Classification

Name	Description
Inactive Timeout	The amount of time that elapses, in seconds, before inactive fragmented flows time out.
Last Timeout	The amount of time, in seconds, to wait after a final fragment is seen before disallowing subsequent IP fragments. This timeout should be between 0 and the value of Inactive Timeout. A value of zero disallows further fragments if the final fragments are delivered out of order. A value greater than Inactive Timeout has no effect. A value less than Inactive Timeout allows state information to be aged more quickly after the final fragment is seen.
Max Outstanding	The maximum number of outstanding fragmented packets allowed at any one time. When this value is decreased, no new fragments will be allowed to be 'outstanding' until the current number of fragments outstanding drops below this value. In other words, if there are 175 outstanding fragments and this value is set to 100, no new fragments will be outstanding until 76 entries are aged out.
Current Outstanding	The current number of outstanding fragmented packets.
Most Outstanding	The largest value of CurrentOutStanding.
Total No Resources Available	The number of initial fragments dropped because no resources were available.
Total Aged Due to Inactivity	The number of state entries aged using Inactive Timeout.
Total Aged Due to Final Timeout	The number of state entries aged using Last Timeout.

SYN Flood Protection

TCP SYN flooding is a common denial-of-service attack which works by sending SYN requests to a host, and then failing to acknowledge the host's SYN+ACK response with an ACK. As a result, the 3-way handshake for each potential TCP session never completes. The attacked host waits for the ACK needed to clear its backlog queue. The ACK never comes, and the host's backlog queue fills up until it cannot accept any new TCP connection attempts.

For some hosts, as few as six-ten half-opened sessions can cause a crash. For this reason, it is recommmended that you implement SYN flood protection for hosts exposed to unprotected traffic. Corporate Web servers or mail servers on a DMZ are prime examples. Conversely, it is not ordinarily necessary to protect hosts on internal networks from this type of attack.

The AP implements both passive and active modes of defense against SYN flooding. Each of these modes works on a per-class basis, and each must be implemented in conjunction with the *Established* class of a statefully classified application.

Active Mode

Active Mode works by sending back a preemptive ACK to the host under attack. This completes the 3-way handshake and clears the host's SYN queue, effectively foiling the attack.

The sample command shown here creates an httpEstablished class (server-a-established), with SYN flood protection set to intervene:

```
> config cbq.1 traffic-class.server-a-established applications \
  httpEstablished syn-protect-mode intervene
```

This class is designed to listen for ACKs sent out to an external host by server A. Whenever an ACK is classified by the *Established* class, the AP automatically sends a SYN+ACK back to server A. This completes the 3-way handshake, leaving server A free to accept new connection attempts. If the initial ACK to A is not part of an attack, a duplicate ACK arrives and is discarded upon receipt.

PASSIVE MODE

Passive mode provides another, less vigilant means of protecting against SYN flood attacks. In this mode, the AP monitors SYN+ACKs passing through an established class. If an ACK does not arrive to complete the handshake in a set period of time, the AP sends an ACK back to the host, clearing its SYN queue.

To configure a SYN protect interval, use this command.

```
> config cbq connection-summary syn-protect-timeout <time in seconds>
```

The default interval is 30 seconds.

9

CONFIGURING QVPN TUNNELS

Access Point QVPN supports secure remote connections for corporate extranets and client connections. Based on the IPSec security standard, Access Point QVPN defines procedures for establishing, managing, and terminating secure VPN tunnels.

In addition, the AP can be configured to function as an L2TP Network Server (LNS), and support L2TP connections from L2TP area concentrators (LACs) or L2TP clients (such as MS-Windows® 2000).

This chapter explains how to set up secure QVPNs across public networks. It contains the following sections.

Introduction to VPNs 380
About Access Point QVPN 383
Configuring IPSec Site-to-Site Tunnels 399
Configuring IPSec Client Tunnels 404
Monitoring IPSec Tunnels 409
Using the IKE Keepalive Protocol 414
Configuring IPSec Transport Mode Connections 417
Configuring L2TP Tunnels 424
Authenticating Dial-in Users 437
Configuring PKI 446

INTRODUCTION TO VPNS

As use of the Internet for business-class applications continues to grow, so has the importance of Virtual Private Networking (VPN) technologies. As an operational definition, VPNs are any type of network that use a combination of *encryption*, *authentication* and *tunneling* techniques to create secure pathways, or *tunnels*, between geographically separated hosts over a *public* network infrastructure.

WHAT ARE VPNS?

It is easiest to define what VPNs are by contrasting them with a system of owned or leased lines like those provided by telephone carriers. The telephony infrastructure provides users with a truly private network. When a call is initiated between separate locations, a circuit is set up for the duration of the communication. During this communication, the circuit is *real*: data flow back and forth over the same physical path. The connection is also entirely *private*. Voice and other data flowing over it are entirely owned by the communicating parties and — theoretically at least — inaccessible to outside parties.

VPNs work differently. Data passed through a secure VPN pathway, or *virtual tunnel*, do not traverse the same physical path, but may take any number of paths. If any of these paths go down while data is transiting between parties, the data find another way to reach their goal. The tunnel between endpoints is *virtual*, rather than actual. Moreover, though the total communication is fragmented in transit, it is also entirely private. In fact, the virtual private network can incorporate sophisticated encryption and authentication techniques, that make data transfers far more secure than other methods allow.

ADVANTAGES OF VPNS

The advantages of VPNs over "real" end-to-end connections rests on three factors: cost, security, and scalability. The most significant problem with the use of switched circuit connections is that owned and leased lines are expensive. In general, the expense of a call is proportionate to the physical distance between parties, and its overall duration.

For local calls, the expense is minimal. Circuits established between business branch offices in the same city are not terribly expensive. Conversely, setting up and maintaining an actual, open circuit between Boston and Bangkok requires network resources to be reserved at each step between the locations.

For national or global businesses needing to establish such circuits on a regular or even constant basis, the cost is prohibitive. One way businesses have addressed this is by leasing high-speed (T1 or higher speed lines) between diverse geographic locations. However, this too is expensive. Moreover, it is an expense that grows along with the increases in the volume of transactions it makes possible.

Low Cost of Public, Packet-Switched Networks

VPNs differ from circuit-switched networks in two fundamental respects. First, and most significantly, VPNs are packet-switched rather than circuit-switched networks. The cost of data transfer over public, packet-switched networks is significantly less than transfer via owned or leased-lines. When one considers the increasing reliance of businesses and organizations on data communications, this cost differential is truly enormous.

Data sent over VPNs are sent in myriad individual packets, each one of which contains some portion of the overall message. Rather than traveling over a dedicated circuit, these packets can take multiple paths over the public network infrastructure. The actual message is reassembled when packets reach their destination. Hence, the appearance of circuit (a virtual circuit) is maintained between the hosts communicating at either end.

Increased Security

The tunneling protocols and security procedures incorporated within VPNs make communications far less susceptible to compromise. In addition to cost reductions, all VPNs address a set of fundamental concerns. Foremost among these are:

- *Privacy*
- *Authenticity/data integrity*
- *User authentication*

Circuit-switched networks do not address these considerations. They lack the protections afforded by VPN protocols, and are inherently porous. Telephone circuits can be bugged, and their data siphoned off by unintended parties. Moreover, once an actual circuit is established between parties, data flowing over it travels *in the clear*. They are not scrambled or encrypted in any way that would make them difficult to decipher. This makes these data vulnerable to compromise by complex subterfuge, or by comparatively simple social engineering (assuming the identity of another).

SCALABILITY

Businesses dependent on owned or leased lines (truly private networks) face a staggering fact: the cost of data transfer increases in rough proportion to their use of the infrastructure.

In contrast, VPNs comprise a *virtual* infrastructure: no fixed or permanent circuit exists between communicating VPN clients and servers, or cooperating VPN gateways. And while this infrastructure leverages the superior economy of public, packet-switched networks, it is also easy to build and maintain. This is because VPN protocols like IPSec incorporate automated management of keys and security associations, allow rapid establishment and management of thousands of virtual connections, without increasing administrative costs.

COMMON VPN APPLICATIONS

The potential applications of VPN tunneling are practically as broad as exist for conventional telephony. A set of distinct categories have emerged in response to organizational demands. It is reasonable to group 90% of VPNs within one or the other of these categories.

- *Remote access for client connections*
- *Business-to-business, for site-to-site communication between different locations*
- *Extranets between businesses, their customers and suppliers.*

Access Point QVPN supports each of these categories with a robust, flexible implementation that provides QoS guarantees for tunneled traffic.

ABOUT ACCESS POINT QVPN

Access Point QVPN supports creation of VPN tunnels within a highly-scalable, layered architecture. The architecture is extremely flexible, allowing many combinations of encryption, authentication, and secure transport of data at the IP layer for both site-to-site and client tunneling operations.

The AP supports two basic tunnel types:

- *Extranet or site-to-site tunnels, using the IPSec security protocol*
- *Tunnels for IPSec or L2TP clients*

Configuring tunnels involves adding and configuring components in the appropriate Virtual Tunnel Transport layer. The arrangement of these layers in the AP System Stack is shown in the status screen below (Figure 49).

Figure 49 Virtual Tunnel Transport Stacks

SYSTEM STACK

	ROW	Layer	Status	
	1	slot.1.1 Ethernet Driver	enabled	up
	2	slot.1.1 cbq.3	enabled	up
	3	slot.1.1 eth.1	enabled	up
	4	slot.1.1 ip.8	enabled	up
	5	slot.3.1 Ethernet Driver	enabled	up
	6	slot.3.1 cbq.1	enabled	up
	7	slot.3.1 eth.2	enabled	up
	8	slot.3.1 ip.1	enabled	up
	9	slot.4.1 Ethernet Driver	enabled	up
	10	slot.4.1 cbq.2	enabled	up
	11	slot.4.1 eth.3	enabled	up
	12	slot.4.1 ip.7	enabled	up
	13	virtual Virtual Tunnel Transport	enabled	down
Virtual Layer 1	14	virtual cbq.4	enabled	down
	15	virtual ipsec.1	enabled	down
	16	virtual ip.4	enabled	down
	17	virtual Virtual Tunnel Transport	enabled	up
Virtual Layer 2	18	virtual cbq.5	enabled	up
	19	virtual iptnl.1	enabled	up
	20	virtual Virtual Tunnel Transport	enabled	up
	21	virtual cbq.6	enabled	up
Virtual Layer 3	22	virtual l2tp.1	enabled	up
	23	virtual l2tp.1.1	enabled	up
	24	virtual l2dial.1.1	enabled	up
	25	virtual ip.5	enabled	down

ABOUT THE VIRTUAL TUNNEL TRANSPORT LAYERS

Each Virtual Tunnel Transport serves different purposes. For this reason, the layers shown in Figure 49 are labeled as Virtual Layer 1 through Virtual Layer 3. These labels represent the relative positions of each level, from the top of the stack down.

VIRTUAL LAYER 1

The first Virtual Tunnel Transport stack supports configuration of IPSec extranet (site-to-site) and client tunnels. See Page 399 for instruction on setting up IPSec site-to-site tunnels, and Page 404 for instruction on setting up IPSec client connections.

VIRTUAL LAYER 2

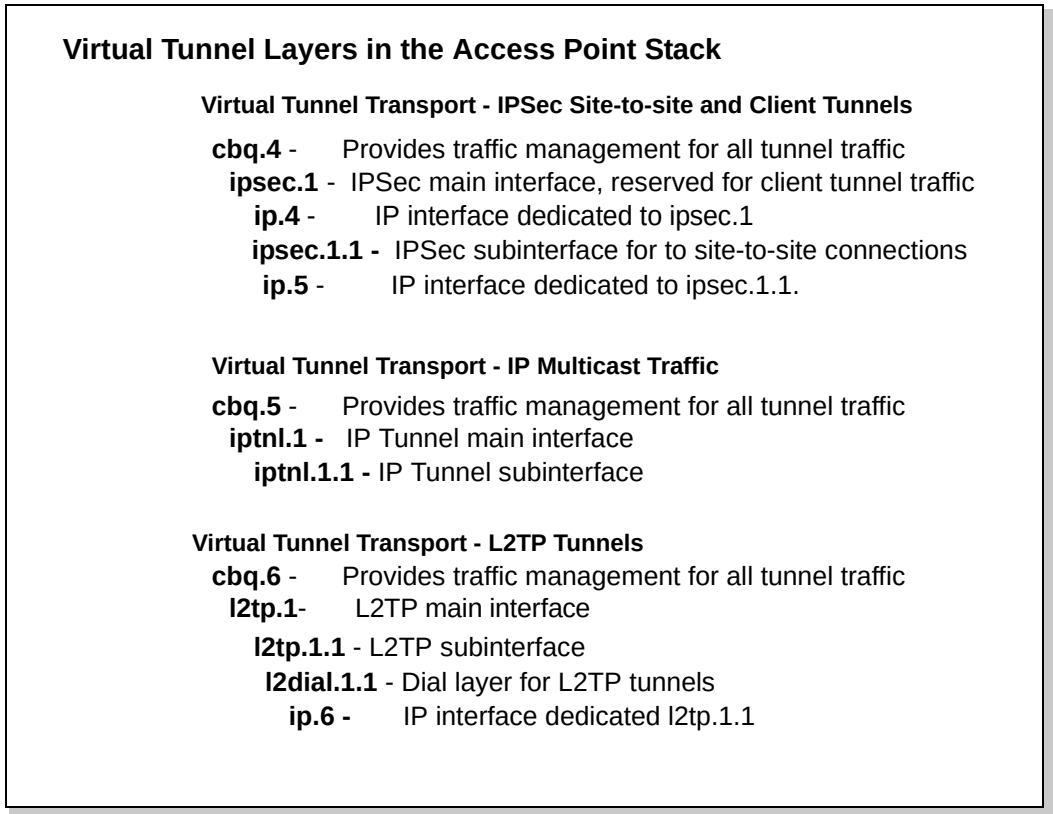
This layer is dedicated to IP-IP tunnels. IP-IP tunnels are not treated in this chapter. They can be used to carry IP Multicast and other types of traffic as explained in “Configuring IP Multicast” on Page 549.

VIRTUAL LAYER 3

This layer’s whole function is to support client connections from remote L2TP clients. The L2TP protocol does not require use of IPSec. See Page 424 for instruction on configuring L2TP tunnels.

Figure 50 provides definitions for each component in the Virtual Tunnel Transport stack. Notice that some components included in this figure do not appear in Figure 49. These layers (such as ipsec.1.1 and iptnl.1.1) are added during tunnel configuration.

Figure 50 Virtual Tunnel Transport Stack Components



ABOUT IPSEC

The AP supports IPSec tunneling for extranets (business-to-supplier, or site-to-site) and client connections. Instructions on configuring each type of IPSec tunnel are given in the pages that follow. For instructions on configuring extranet tunnels, see Page 399. See Page 404 for instruction on configuring IPSec client connections.

The IPSec (IP Security Architecture) provides a stable and flexible framework for network layer security. IPSec is designed for IPV4 and IPV6 networks. The framework provides the following:

- *Data origin authentication - To verify that datagrams originate at the host claiming to have sent them*
- *Data integrity - To ensure that data has not been corrupted or tampered with in transit*
- *Privacy - To ensure that the content of a message can only be read by an authorized party*
- *Replay protection - To ensure that an attacker cannot capture and replay packets at a later time*
- *Automated management of keys and security associations - To ensure that VPNs can be implemented throughout an extended enterprise without manual configuration, enabling the VPN infrastructure to scale easily as demands increase*

IPSec incorporates the following protocols.

- IKE -Internet Key Exchange Protocol

  IKE provides a framework for establishing security associations and managing cryptographic keys.

- ESP - Encapsulating Security Payload

  ESP provides data confidentiality and protection with optional authentication and replay-detection services. ESP completely encapsulates user data, and can be used either by itself or with Authentication Header (AH). *AH is not supported by the AP.*

- AH - Authentication Header

  A security protocol that provides authentication and optional replay-detection services. AH is embedded in the data to be protected (a full IP data-

gram, for example). AH can be used either by itself or with Encryption Service Payload (ESP). *AH is not supported by the AP.*

ABOUT IKE

The secure generation and distribution of keys is basic to the establishment of a secure data communications environment. The IKE framework (also known as ISAKMP/Oakley) provides a standard way for cooperating systems to authenticate one another, agree upon a set of security algorithms, and on a set of keys to use throughout their communication. The secure exchange of keys is the most significant factor in protecting the security of tunneled information. IKE uses preshared keys and digital certificates to authenticate tunnel connections to remote gateways.

Access Point QVPN implements the Internet Key Exchange (IKE) framework (also referred to as ISAKMP/Oakley), allowing automatic negotiation of security associations (SAs) and automated generation and refresh of cryptographic keys.

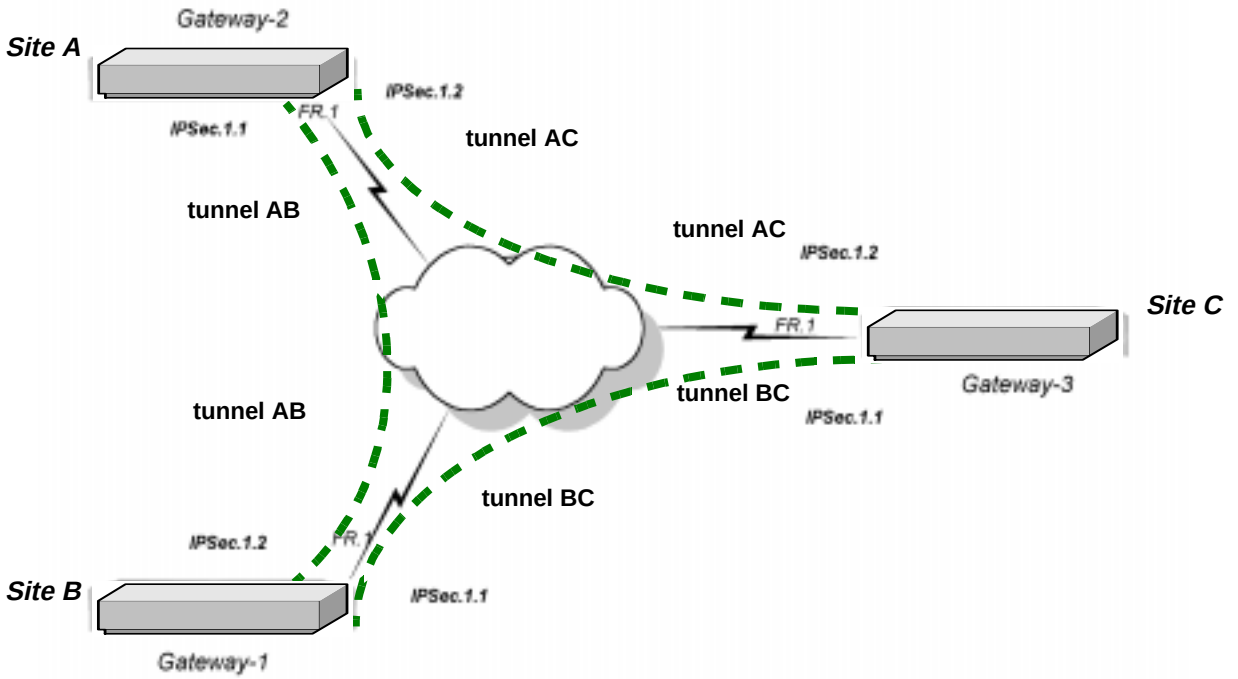
OVERVIEW: IPSEC SITE-TO-SITE TUNNELS

Before moving on, here are a few basic facts about the Access Point QVPN tunneling framework for site-to-site tunnels.

- *Site-to-site tunnels run over a virtual IPSec interface. This interface is associated with a pair of endpoints: a local gateway (represented by the router address) and a remote gateway.*
- *Each pair of endpoints can support one or more IPSec tunnels.*
- *A unique security association (SA) pair exists for each tunnel, and is shared by the two tunnel endpoints: one inbound, one outbound.*
- *Tunnels that share common endpoints can use different security profiles. Security profiles define the mechanisms used to secure communications in a tunnel. Default profiles (ranging from strong to weak) have been provided.*

Figure 51 shows a high-level representation of three site-to-site connections configured between Access Point QVPN systems.

Figure 51 Site-to-Site Tunnels Among Three Access Point QVPN Systems



Each AP shown in Figure 51 functions as a VPN gateway. Instances of ipsec.1.1 and ipsec.1.2 serve as virtual site-to-site tunnel interfaces with another remote gateway. Dotted lines represents virtual connections. Each IPSec interface can support one or many site-to-site VPN tunnels.

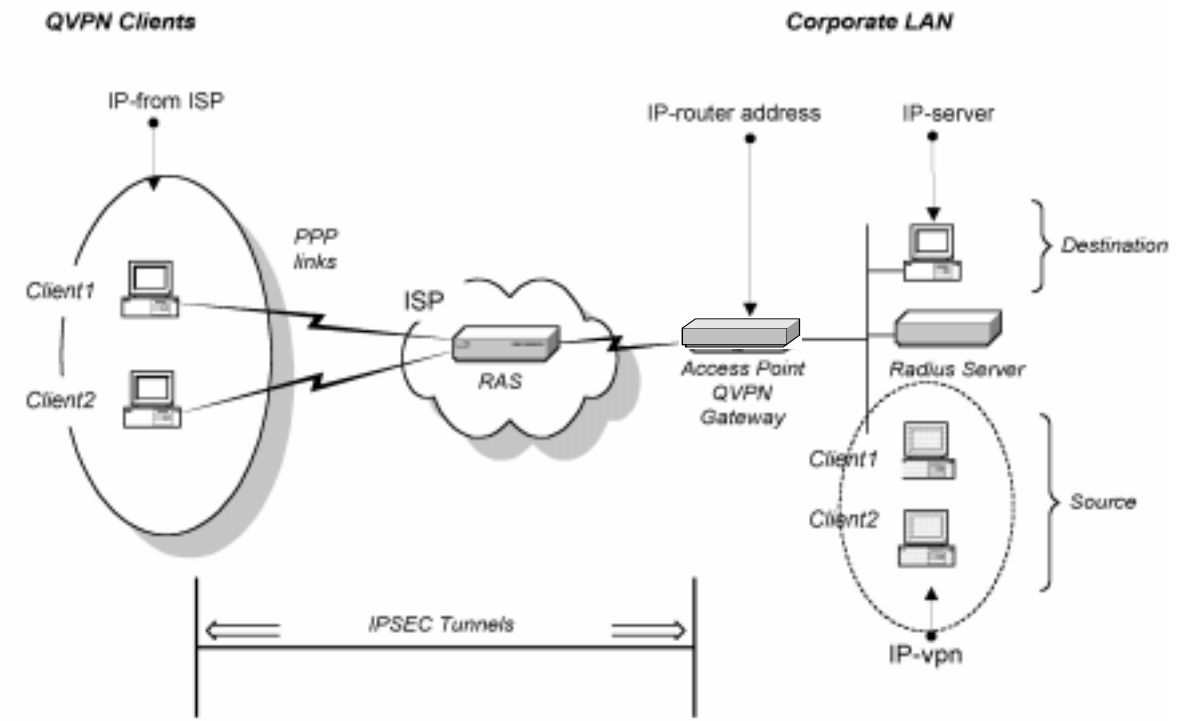
Refer to "Configuring IPSec Site-to-Site Tunnels" on Page 399 for information about configuring site-to-site tunnels.

OVERVIEW: IPSEC CLIENT TUNNELS

The mechanisms that make IPSec-based client connections possible are transparent to users of the QVPN client. These mechanisms include the IPSec and IKE protocols, described in RFCs 2401 and 2409.

In the QVPN architecture, QVPN clients and the Access Point QVPN gateway function as origination and termination points of a secure QVPN tunnel. Though packets sent through this tunnel traverse the wide area, clients appear (to destination systems) to be directly connected to the same LAN segment as the QVPN gateway. Figure 52 shows a sample IPSec-based client scenario.

Figure 52 Sample IPSec-Based Client Scenario



IPSec Client: How it Works

In an IPSec-based client scenario, a QVPN client receives two IP addresses, and uses them to construct an *inner* and an *outer* IP header for delivery of information to an ultimate destination system (IP-server). One address is provided by the ISP. The other is provided by the QVPN gateway, based on information provided in the configuration process. In general terms, here's how it works:

- *The outer IP header contains a source and destination address. The source address is usually assigned by the ISP.*

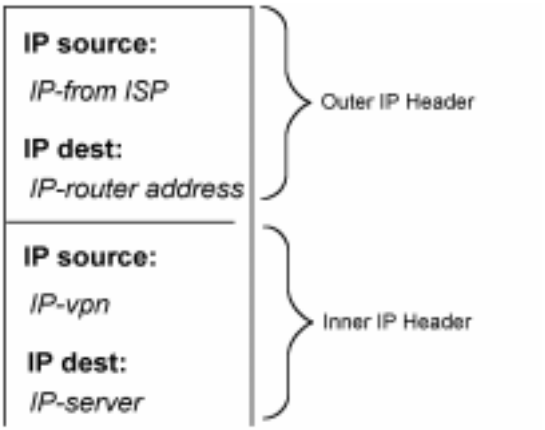
  The destination address of the outer header is the router address of the QVPN gateway. This header's purpose is simply to deliver encrypted IP packets to the gateway.

- *The inner IP header has a different source and destination address. The source address actually belongs to the gateway's internal network, and is loaned to the QVPN client for the purpose of connecting to the corporate LAN.*

The destination address is the ultimate destination of the IP packets sent by the connecting client.

The whole intent of using two sets of IP addresses is to set up a virtual connection between the client and a host on the private network. The ESP protocol is used to encapsulate one packet (and its associated data) inside another, and to send the entire construction over the wide area as an encrypted datagram.

When the encrypted datagram reaches the QVPN gateway, the gateway decrypts it, and checks its message digest to be sure it was not tampered with or corrupted in transit. The gateway then matches the packet to a specific subnet, and sends it to its destination.



The addresses shown in the header at right derive from the sample client scenario shown in Figure 52. If you turn back to this figure, you will note that the labels used in this illustration (IP-from ISP, IP-router address, and so on) correspond to the callouts used in that figure.

MORE ON CLIENT CONNECTIONS

Though the mechanics of session establishment and authentication are transparent to the user, it is helpful to know what is happening during connection setup. The two sections that follow provide details.

SESSION ESTABLISHMENT AND KEY EXCHANGE

The QVPN framework uses the Internet Key Exchange (IKE) protocol to negotiate a session between connecting clients and a QVPN gateway. Session negotiation involves two phases:

- Phase I IKE Session Establishment

  The work of Phase I consists in establishing a pair of bidirectional security associations (SAs) for the connection. Each SA applies to one side or

direction of the overall connection. That is, one SA is associated with the connection from client A to QVPN gateway B, another is associated with the connection between gateway B and client A.

- Phase II Quick Mode

  Quick mode establishes uni-directional IKE SAs between the connecting client and the QVPN gateway. The SAs allow for secure exchange of IKE key information between the two systems.

If user password authentication for IPSec has been enabled on the QVPN gateway (as explained in "Enabling User Password Authentication" on Page 440), extended authentication and configuration follow these phases. In this phase, the gateway prompts the client for a user name and password. An IP address is assigned to the client, either by the QVPN gateway or by a RADIUS server.

Apart from being prompted for a name and password, the details of this phase are transparent to the connecting user. They are explained in the following section.

Extended Authentication and Configuration

If user password authentication for IPSec is enabled, the QVPN gateway prompts the connecting client for a user name and password. If the gateway finds a match for the name/password provided, the connection is accepted and an IP address is assigned to the client. The details of this transaction are described in the steps that follow. Note that these steps will not occur if you choose to disable user password authentication for the IPSec protocol. See "Disabling User Password Authentication" on Page 441 for more information.

1. First, the QVPN gateway prompts the client for user and password information. This information is used to authenticate the client.
2. Once the user provides the name/password pair, the QVPN gateway compares this information against the records contained in its internal user table. If the information for a particular user is not present, the gateway uses the information for the default user.
3. If the gateway cannot find a match for the pair, the connection is refused. If a match is found, the gateway selects the client group associated with the pair.
4. Finally, the gateway selects an IP address from the remote address and mask specified for the client group. This is the address that the gateway allocates to the client.

When a client connection has been established, QVPN clients appear to the servers to which they are connected to exist on the same LAN segment.

OVERVIEW: IPSEC TRANSPORT MODE

IPSec Transport Mode is used to secure IP communications between two systems. Transport mode encapsulation is similar to tunnel mode, except that IPSec tunneling is used for traffic flowing through the Access Point QVPN gateway to IP systems behind the gateway.

These Transport Mode connections are set up in a fashion similar to that of site-to-site or client tunnels. Once you configure Transport Mode connections, then these connections are used to secure the traffic flowing between IP systems.

Whether you use site-to-site or client connections depends on your configuration. For example, L2TP uses IPSec Transport Mode connections as follows:

- A site-to-site Transport Mode connection is used when the connection is between the LAC and the LNS.
- A client Transport Mode connection is used when the connection is between a PC running an L2TP LAC client and the LNS.

Refer to “Configuring IPSec Transport Mode Connections” on Page 417 for information about configuring transport mode connections.

CONFIGURING GENERAL VPN PARAMETERS

When configuring VPNs, you must perform these general configuration tasks.

- Designating a router address
- Configuring a security profile

DESIGNATING A ROUTER ADDRESS

The router address is used as the local VPN endpoint. It is one of the gateway's IP addresses, and is used by the Internet Key Exchange (IKE) protocol and Public Key Infrastructure (PKI) certificates to uniquely identify the local VPN gateway.

The router address must be reachable from other Access Point interfaces and from clients over the public network. Generally, designating a LAN IP address as the router ID is sufficient. However, if the interface on which this address resides goes down or becomes unreachable, the router address also becomes unreachable. To ensure that the router address remains up at all times, you can designate the ip.loopback address as the source address of the router address. See "Using the Loopback Interface Address" on Page 180 for information on configuring the ip.loopback address.

You should also bear the following in mind.

- Only one IP address can function as the router address for the AP.
- For tunnels to come up, you must also configure a router address (a cooperating, remote VPN endpoint) on the remote VPN system. See "Configuring the Remote Gateway" on Page 400 for more information.

To designate an IP address as the router address:

```
> config ip.<instance> address.<IP address> router-address true
```

NOTE *Only one IP address can be set as the router address.*

After adding a router address, you can check to see that it is present using this command:

```
> show ip.* address-table summary
```

```
                                IP.* ADDRESS-TABLE SUMMARY

INTERFACE|       ADDRESS|          Net Mask| Broadcast Address|
         |              | Configured Protocols|         Row Status|
         |              |             Src Addr|     Router Address|
---------|--------------|---------------------|-------------------|
     ip.1|   10.1.3.201|      255.255.255.0|                 1|
         |              |                   |            active|
         |              |          0.0.0.0|             false|
     ip.3|    10.1.2.1|      255.255.255.0|                 1|
         |              |                   |            active|
         |              |          0.0.0.0|              true|
```

Note that the Router Address field for the ip.3 interface is true. Its associated IP address (10.1.2.1) is the router address.

CONFIGURING IPSEC SECURITY PROFILES

A security profile is a set of security attributes used to describe one or more IPSec tunnel or transport connections. Security profiles determine key security aspects of the tunnel or transport connections to which they apply. The AP provides a set of default security profiles. These are:

- default-auth
- default-client
- default-strong
- default-weak

All profiles are modifiable. To show existing profiles, issue this command:

```
> show ipsec security-profiles
```

IPSEC SECURITY-PROFILES

NAME	Protocol Compression Inactivity Timer Assignment Status	Encryption Expiration Timer Enable PFS Row Status	Authentication Expiration Volume Oakley Group
default-auth	esp null 0 unassigned	null 0 false active	md5 0 first
default-client	esp null 900 unassigned	des 0 false active	md5 0 first
default-strong	esp null 0 assigned	des3 86400 true active	md5 0 first
default-weak	esp null 0 assigned	des 0 false active	md5 0 first

ADDING AND MODIFYING SECURITY PROFILES

To add a new security profile, use the *add* command. An example of a new security profile is shown here:

```
> add ipsec security-profile.<profile name> protocol esp encryption des3\
  authentication sha1 compression lzs expiration-timer 1000 \
  expiration-volume 1000 inactivity-timer 0
```

To modify an existing profile, use the *config* command as shown in this example:

```
> config ipsec security-profile.<profile name> inactivity-timer 900
```

REMOVING SECURITY PROFILES

To remove a profile, use the *remove* command as shown here.

```
> remove ipsec security-profile.<profile name>
```

Note that you cannot configure or remove a security profile when its Assignment Status is active. All tunnel references to a profile must be removed before the profile can be reconfigured or removed.

CONFIGURING SECURITY PROFILE PARAMETERS

During the IKE exchange, the security profile applied to a connection is used to build a pair of bidirectional SAs (Security Associations). You can modify aspects of a security profile to customize it for specific applications. Characteristics defined by a security profile include the following:

- The type of authentication applied to a tunnel.
  MD5, SHA, and NULL (no authentication) are supported.
- Type of encryption used.
  DES, 3DES and NULL (no encryption) are supported.
- The protocols used to encapsulate traffic.
  ESP is supported.
- Automatic re-keying and generation of a new SA pair (based on connection up time, connection inactivity, and/or packet volume).
  Expiration timer and volume information.

Table 22 lists the security profile parameters for site-to-site tunnel and transport connections. Table 23 lists the security profile parameters for client connections.

To configure security profile parameters, specify the appropriate parameters from Table 22 or Table 23 with the following command:

```
> config ipsec security-profile.<profile-name>
```

Table 22 Security Profile Parameter Settings for Site-to-Site Connections

authentication	Specifies the authentication protocol used by ESP. Supported types: • md5 (the default) • sha1 • null Little performance benefit accrues from choosing null as the authentication type, since the performance impact on using MD5 or SHA1 is quite small.
compression	Available with acceleration hardware only. Specifies the compression method. Choices are: • lzs • null (the default)
enable-pfs	T/F. Default value: False. When enabled (T), PFS is enforced and requires generation of new material for each protocol SA.

encryption	Specifies the encryption protocol. Supported types: • des (the default) • des3 • null
expiration-timer	Places an upper limit on the life of an SA by specifying the number of seconds that elapse before the SA expires. Minimum allowed expiration time: 300 (5 minutes). Maximum allowed expiration time: 86400 (24 hours) Default value: 86400 (24 hours)
expiration-volume	Places an upper limit on the life of an SA by specifying the number of Kbytes that pass through a connection before re-keying. Re-keying begins after 95% of the volume has been used. Minimum allowed value: 100 KB Maximum allowed value: 10000000 KB (10GB) Default value: 0 (disabled)
inactivity-timer	Specifies the number of seconds without traffic that elapse before an SA is removed. The inactivity timer applies to an entire connection, not to individual SAs. Minimum allowed value: 600 (10 minutes) Maximum allowed value: 86400 (24 hrs) Default value: 0 (disabled)
oakley-group	Specifies the Oakley group used by ISAKMP. Values allowed: first, second Default value: first
protocol	Specifies the IPSec security profile in use. Supported values: esp, esp+ipcomp Note: Protocol must be set to esp if compression=null, esp+ipcomp otherwise. esp+ipcomp is only supported on routers with hardware that supports compression. Default: esp
row-status	Specifies the row status of the security profile. Default value: active

Table 23 Security Profile Parameter Settings for Client Connections

authentication	Supported authentication protocol types: • md5 • sha1 • null Default value: md5
compression	null
enable-pfs	Enforce PFS values: • true • false Default value: false
encryption	Supported encryption protocol types: • des • des3 • null Default value: des
expiration-timer	Minimum allowed value: 300 (5 minutes) Maximum allowed value: 86400 (24 hours) Default value: 86400 (24 hours)
expiration-volume	Minimum allowed value: 100 KB Maximum allowed value: 10000000 KB (10GB) Default value: 0 (disabled)
inactivity-timer	Minimum allowed value: 600 (10 minutes) Maximum allowed value: 86400 (24 hrs) Default value: 0 (disabled)
oakley-group	Values allowed: first, second Default value: first
protocol	esp
row-status	active

CONFIGURING IPSEC SITE-TO-SITE TUNNELS

Following are instructions for configuring IPSec QVPN tunnels. Setting up tunnels is a straightforward process that involves the following steps.

1. Designate a router address for the local Access Point QVPN gateway.
2. Choose or configure IPSec security profiles.
3. Add and configure IPSec subinterfaces.

   Subinterfaces are the virtual interfaces between the router and a remote VPN gateway.
4. Add and configure IPSec tunnels.

   Access Point QVPN supports creation of dynamic and static VPN tunnels.

The sections that follow discuss these steps in detail.

STEP 1: DESIGNATE A ROUTER ADDRESS

The router address is used as the local VPN endpoint. It is one of the gateway's IP addresses, and is used by the Internet Key Exchange (IKE) protocol and Public Key Infrastructure (PKI) certificates to uniquely identify the local VPN gateway. Refer to "Designating a Router Address" on Page 393 for more information.

STEP 2: CHOOSE OR CONFIGURE IPSEC SECURITY PROFILES

A security profile is a set of security attributes used to describe one or more IPSec tunnel or transport connections. Security profiles determine key security aspects of tunnels to which they apply. Refer to "Configuring IPSec Security Profiles" on Page 394 for more information.

STEP 3: ADD AND CONFIGURE IPSEC SUBINTERFACES

Next, add a new IPSec subinterface and a dedicated IP instance to the Virtual Tunnel Transport stack using the *stack* command, as follows:

```
> stack virtual cbq.<instance> ipsec.<instance> ipsec.<instance>.new ip.new
```

IPSec subinterface

Dedicated IP interface

Use of the "new" qualifier in this example causes an IPSec subinterface to be generated and uplinked to the IPSec layer. In addition, a new IP interface (again, generated with "new") is created.

It is not necessary to assign an address to the dedicated IP interface added here. The interface address defaults to unnumbered (0.0.0.0) when tunnels are added. Unnumbered source addresses are associated with the router address configured earlier.

CONFIGURING THE REMOTE GATEWAY

Next, configure a remote gateway address for the subinterface. From the local gateway's point of view, the remote gateway address is the remote interface to the virtual tunnel framework. The remote gateway address *must be the router address of the remote gateway.*

NOTE *This is a vital configuration step: if the remote gateway address and the remote system's router address differ, you will not be able to bring up a secure connection between the local and remote VPN gateways.*

To configure a remote gateway for an IPSec subinterface, proceed as follows:

```
> config ipsec.1.1 remote-gateway 192.2.1.1
```

CONFIGURING AN AUTHENTICATION METHOD

Before a tunnel can be established, IKE must authenticate the remote (connecting) gateway. Gateway authentication can be accomplished using preshared keys or digital certificates.

ASSIGNING PRESHARED KEYS

Using preshared keys requires configuration of a shared key on the gateway's IPSec interface. To configure a preshared key, proceed as follows:

```
> config ipsec.1.1 preshared-key <text string>
```

The specified text string must be between 1 and 32 characters long. Any com-

bination of alphabetic and symbol characters is acceptable. It is case-sensitive.

When assigning a key, obey the same security considerations used to safeguard any password. Longer text strings are generally better (because they are more difficult to break) than shorter ones; non-obvious or random strings are preferable to names (like "johnsmith") or declarative statements ("elvisisking") that can be guessed by an attacker.

ASSIGNING DIGITAL CERTIFICATES

To specify the IKE authentication method to use digital certificates:

```
> config ipsec.1.1 ike-authentication-method [ dsaSig | rsaSig | rsaEncrypt ]
```

To configure digital certificates, refer to "Configuring PKI" on Page 446.

CHECKING THE IPSEC CONFIGURATION

After completing the IPSec configuration, issue a *show* command to see that you have added all necessary data. The example shown here refers to a site-to-site tunnel to which 40 tunnels have been added.

```
> show ipsec.1.1 vpn
```

```
                                IPSEC.1.1 VPN

                    Type  = siteToSite
          Remote Gateway* = 192.168.10.1
           Preshared Key* = the-key
          Current Tunnels = 40
            Total Tunnels = 40
Last Tunnel Change        = 754 (0 hours 0 min 7 secs)
         Invalid Sequences = 0
          Invalid Authents = 0
           Invalid Formats = 0
```

STEP 4: ADD AND CONFIGURE SITE-TO-SITE TUNNELS

Dynamic tunnels use IKE to establish tunnel security associations (SAs) and securely re-establish SAs when the SAs expire due to volume or timer limits.

In contrast to this, static tunnels require the administrator to manually create SAs. This process, though time-consuming, might be necessary if IKE interoperability problems occur, or if the remote VPN gateway does not support IKE.

CONFIGURING DYNAMIC TUNNELS

Each tunnel you add requires a distinct name and a security profile. You can also include a specific flow definition. The general form for adding VPN tunnels is:

```
> add ipsec.1.1 tunnel.<tunnel name> security-profile <security profile> \
  local-address <address> local-mask <mask> remote-address <address> \
  remote-mask <mask>
```

Each IPSec subinterface can be in charge of one or many tunnels. The subinterface always directs IP datagrams into the tunnel whose definition most closely matches information in the IP packet header. Encryption and authentication measures are incorporated in the definition of the tunnel chosen.

As an example, this command adds a general purpose dynamic tunnel called all-traffic that uses the default-weak security profile:

```
> add ipsec.1.1 tunnel.all-traffic security-profile default-weak \
  local-address 194.3.4.0 local-mask 255.255.255.0 \
  remote-address 123.4.5.0 remote-mask 255.255.255.0
```

This tunnel applies to all traffic matching the flow definition. The command following adds a tunnel (called strong) that uses the default-strong security profile. This command also includes a specific flow definition.

```
> add ipsec.1.1 tunnel.strong security-profile default-strong \
  local-address  192.1.1.0 local-mask 255.255.255.0 \
  remote-address 192.2.1.0 remote-mask 255.255.255.0
```

CONFIGURING STATIC TUNNELS

Because the default tunnel type is dynamic (siteToSiteDynamic), configuring static tunnels requires you to declare the tunnel as static. Configuring static tunnels takes more work, and is consequently more error prone than relying on IKE to set up security associations dynamically. For this reason, static tunnels are usually employed when IKE has difficulties negotiating a connection between two gateways.

Following are two static tunnel definitions.

```
> add ipsec.1.1 tunnel.weak type siteToSiteStatic security-profile default-weak
```

```
> add ipsec.1.1 tunnel.strong type siteToSiteStatic local-address 192.1.1.0
  local-mask 255.255.255.0 remote-address 192.2.1.0 remote-mask
  255.255.255.0 security-profile default-strong
```

CONFIGURING STATIC SECURITY ASSOCIATIONS

When dynamic tunnels are created, IKE automatically generates inbound and outbound security associations (SAs) for each tunnel. Creating static tunnels requires administrators to create SAs manually by providing SA instance numbers for each tunnel direction. Each SA requires four pieces of information, used to create security for each direction of the tunnel. These are:

- A direction (either inbound or outbound)
- Security Parameter Index (SPI) for each tunnel direction
- An authentication key
- An encryption key

The SPI is a unique SA identifier in the header of tunneled packets. The inbound SA on one machine and outbound SA on another must have the same SPI and encryption/authentication keys. Authentication and encryption keys are keyed in as bytes of information. The number of bytes required depends upon the encryption transform used by the tunnels.

```
> add ipsec.1.1 tunnel.weak security-association.45 protocol esp spi 300
  direction ipSecinbound authentication-key
  263748596874475647463535353546576978576970584736 encryption-key
  343457687983457485746353827363454547493030595734444
```

```
> add ipsec.1.1 tunnel.weak security-association.46 protocol esp spi 303
  direction ipSecoutbound authentication-key
  263748596874475647463535353546576978576970584736 encryption-key
  343457687983457485746353827363454547493030595734444
```

After keying in static associations, you can check the information using this command:

```
> show ipsec.1.1 tunnel.weak security-associations summary
```

CONFIGURING WILDCARD TUNNELS

When configuring tunnels, you can use a wildcard for local and remote tunnel addresses. However, to ensure that traffic is directed into the correct tunnel to the destination, you must manually configure a static route to the remote destination.

The following command wildcards the destination address of the tunnel:

```
> add ipsec.1.1 tunnel.1 security-profile default-strong local-address
```

```
194.3.4.0 local-mask 255.255.255.0
```

Because IPSec tunnels are built on virtual subinterfaces, which are unnumbered, use the following command to direct traffic into tunnel.1 (assuming ip.7 is the IP instance for the ipsec.1.1 subinterface).

```
> add ip static-route.194.3.4.0 local-mask 255.255.255.0 tos.0 next-hop.ip.7
```

STEP 5: CHECK THE TUNNEL SETUP

After configuring tunnels, issue a *status* command to make a final check of the tunnel setup. For the tunnels to function, the interface to which they are attached must be *enabled*, with a status of *up*.

Specifically, you would check the following:

- The ipsec.1.1 interface is up.
- The IP instance for the ipsec.1.*n* interface is up.

You should also check that the router address is configured on the gateway. The router address is an encryption/decryption point for all tunnel traffic. Without a router address, the QVPN framework will not function.

CONFIGURING IPSEC CLIENT TUNNELS

Setting up VPN tunnels between Access Point QVPN systems and clients involves the following steps.

1. Designate a router address for the local Access Point QVPN gateway.
2. Choose or configure IPSec security profiles.
3. Configure an authentication method.
4. Add and configure IPSec client group tunnels.
5. Check tunnel setup.

The sections that follow discuss these steps in detail.

STEP 1: DESIGNATE A ROUTER ADDRESS

The router address is one of the gateway's IP addresses, and is used by the Internet Key Exchange (IKE) protocol and Public Key Infrastructure (PKI)

certificates to uniquely identify the local VPN gateway. Refer to "Designating a Router Address" on Page 393 for more information.

STEP 2: CHOOSE OR CONFIGURE IPSEC SECURITY PROFILES

A security profile is a set of security attributes used to describe one or more IPSec tunnel or transport connections. Security profiles determine key security aspects of tunnels to which they apply. Refer to "Configuring IPSec Security Profiles" on Page 394 for more information.

STEP 3: CONFIGURE AN AUTHENTICATION METHOD

When connecting to the QVPN gateway, clients propose an authentication method. This method can differ from one client to the next. For this reason, the QVPN gateway is designed to accommodate authenticating client connections by preshared key or digital certificates.

When a client initiates a connection, the gateway attempts to authenticate it using whatever method the client proposes. For this reason, it is not necessary to configure an IKE authentication method on the ipsec.1 interface.

ASSIGNING A PRESHARED KEY TO THE CLIENT INTERFACE

For the gateway to accommodate preshared keys, you must configure a key for the ipsec.1 interface. The same preshared key is associated with all tunnels owned by the client interface. To configure a key, use this command:

```
> config ipsec.1 preshared-key <text string>
```

The text string you provide must be betweeen 1 and 32 alphabetic characters long. It is case-sensitive. For security reasons, random keys (though difficult to remember) are better than phrases that are easy to guess (e.g., mynameis-johnsmith).

ASSIGNING DIGITAL CERTIFICATES

As when using preshared keys, the gateway must be able to handle the certificate information it receives from connecting clients. Otherwise, it will be unable to authenticate the connection. To configure digital certificates, refer to "Configuring PKI Certificates" on Page 446.

STEP 4: ADD AND CONFIGURE CLIENT GROUPS

Client tunnels use IKE to establish a pair of SAs (Security Associations), and also re-establish SAs that have expired due to volume or timer limits.

The QVPN framework requires each client tunnel to belong to a *client group*. Use of client groups makes it easier to administer the client tunnel framework. You can think of a client group as a profile that describes one or more clients. Once you define a group, the tunnels you add to it inherit its characteristics. So you only need to define these characteristics once.

To configure client groups, you should:

- Add the client group
- Set up extended authentication services

ADDING CLIENT GROUPS

Client groups are a mechanism used to aggregate a set of characteristics common to a group of users. Usually, client groups will have more than one user. However, a client group can contain a single user. The only restriction is that every user on the gateway must belong to exactly one client group.

To define a client group for tunnels, use this command:

```
> add ipsec.1 tunnel.<group name> type clientGroup \
  security-profile <security profile>
```

For client groups, the *type* is always clientGroup.

As an example, this command adds a general purpose tunnel called all-strong that uses the default-client security profile:

```
> add ipsec.1 tunnel.all-strong type clientGroup security-profile default-client
```

The IPSec client interface can be in charge of one or many tunnels. The interface always directs IP datagrams into the tunnel whose definition most closely matches information in the IP packet header. Encryption and authentication measures are incorporated in the definition of the tunnel group chosen.

CHOOSING A METHOD FOR ADDRESS ASSIGNMENT

When configuring client groups, you must choose a method for assigning addresses to clients belonging to the group. In this context, the address

assigned is the enterprise IP address and subnet. This address is the *inner address* of the tunneled IP packet. It describes the ultimate destination of the packet. In contrast, the outer address is typically assigned by the client's ISP.

You can specify two ways for assigning the enterprise address to clients belonging to a group. These are:

- From an internal address pool
- By a RADIUS server

By default, client address assignment is disabled so that the QVPN gateway can accommodate site-to-site tunnels (which require no address assignment) or client tunnels.

EXAMPLE: ASSIGNMENT FROM AN INTERNAL ADDRESS POOL

If you specify internalPool as the *client-address-assign* parameter, you must specify the *remote-address* and *remote-mask* parameters. This command adds a tunnel (called dialstrong) that uses the default-strong security profile. The client address assignment method is internalPool. This means that clients belonging to this group will be assigned an IP address based on the remote address and remote mask specified.

```
> add ipsec.1 tunnel.dialstrong type clientGroup security-profile default-strong
  client-address-assign internalPool remote-address 192.2.1.0 \
  remote-mask 255.255.255.0
```

Note that a remote address/mask combination are required when internalPool is the method for client address assignment. The gateway will not add the client group definition without this information.

EXAMPLE: ASSIGNMENT BY RADIUS SERVER

To specify that the client's address will be obtained from a RADIUS server as part of the user's information, use a command like the following:

```
> add ipsec.1 tunnel.strong type clientGroup security-profile default-strong \
  client-address-assign radius
```

In this case, the gateway retrieves a username/password and an IP address from the RADIUS server during the authentication phase.

CONFIGURING CLIENT GROUPS WITH RADIUS

Client configuration is based on the client group with which the user is associated. The client group comes from the user table or the RADIUS server (the Tunnel-Private-GroupID attribute).

A RADIUS client (the AP) can only retrieve attributes from the RADIUS server upon successful user authentication. To use RADIUS to store these attributes, a user must be configured in the RADIUS database. The username is that of the user. The password is configured on the AP (a password known to the AP and RADIUS server). It is this password that RADIUS authenticates, and upon success, RADIUS returns the rest of the user's attributes. If user password authentication is disabled, the client group is the (default) client group.

Once the client group is known, client configuration parameters can be sent to the client. These RADIUS attributes are supported, and can be included when configuring the RADIUS server's user entry.

Framed-IP-Address and Framed-IP-NetMask

The internal IP address and network mask. It is the inner tunneled enterprise address and network mask.

Xedia-DNS-Server

The enterprise DNS server address that the client should use to resolve domain names.

Xedia-NetBios-Server

The enterprise NetBIOS server address.

Xedia-Client-Access-Network (Only for Lucent IPSec Client)

The list of enterprise networks to which the client is allowed access through the VPN tunnel. This attribute is used only for the Lucent IPSec Client.

After successful client configuration, a client tunnel is created with the name of the user within the specified client group. For example, the connecting user Eric with Eng specified as the client group would create a tunnel called Eric within the Eng client group.

STEP 5: CONFIGURING USER AUTHENTICATION

You can configure the gateway to authenticate users. Refer to "Configuring User Authentication" on Page 89 for more information about enabling user password authentication and configuring users in the user table.

STEP 6: CHECK THE TUNNEL SETUP

After configuring client tunnels, issue a *status* command to make a final check of the tunnel setup. For the tunnels to function, the interface to which they are attached must be *enabled*, with a status of *up*.

Specifically, you would check the following:

- The ipsec.1 interface is up.
- The IP instance for the ipsec.1 interface is up.

You should also check that the router address is configured on the gateway. The router address is an encryption/decryption point for all tunnel traffic. Without a router address, the QVPN framework will not function.

MONITORING IPSEC TUNNELS

The AP QVPN framework provides useful information on tunnel operations. You can use the information to monitor tunnel usage and troubleshoot problems.

ACCESSING SUMMARY INFORMATION

To access summary information on tunnels attached to a subinterface, use this command.

```
> show ipsec.1.1 tunnels summary
```

Issuing this command shows information for all tunnels configured on the specified subinterface. The screen shown in Figure 53 provides information collected for a single tunnel (tunnel-0).

Figure 53 IPSec Tunnels Summary

```
                                IPSEC.1.1 TUNNELS SUMMARY

          INTERFACE|                        NAME|                  Type|
                   |                            |                  Mode|
                   |                            |                 State|
                   |                            |     Last State Change|
                   |                            |         Local Address|
                   |                            |            Local Mask|
                   |                            |        Remote Address|
                   |                            |           Remote Mask|
                   |                            |        Remote Gateway|
                   |                            |      Security Profile|
                   |                            |        Authentication|
                   |                            |        Address Assign|
                   |                            |            Row Status|
-----------------|----------------------------|----------------------|
          ipsec.1.1|                   tunnel-0|     siteToSiteDynamic|
                   |                            |                    up|
                   |                            |                    up|
                   |                            |583 (0 hours 0 min 5>|
                   |                            |           192.168.1.3|
                   |                            |       255.255.255.255|
                   |                            |           193.168.1.3|
                   |                            |       255.255.255.255|
                   |                            |          192.168.10.1|
                   |                            |               des-md5|
                   |                            |          presharedKey|
                   |                            |                 none|
                   |                            |               active|
```

When examining summary information, make sure the local address/mask and remote address/mask combination match. This is vital because the IKE protocol uses this information to bring up the tunnels between local and remote gateways.

When IKE negotiates a tunnel connection, the status is "initializing". The status changes to up when the tunnel is up, and changes to down if the tunnel fails for any reason.

DISPLAYING INFORMATION ON TUNNEL SAS

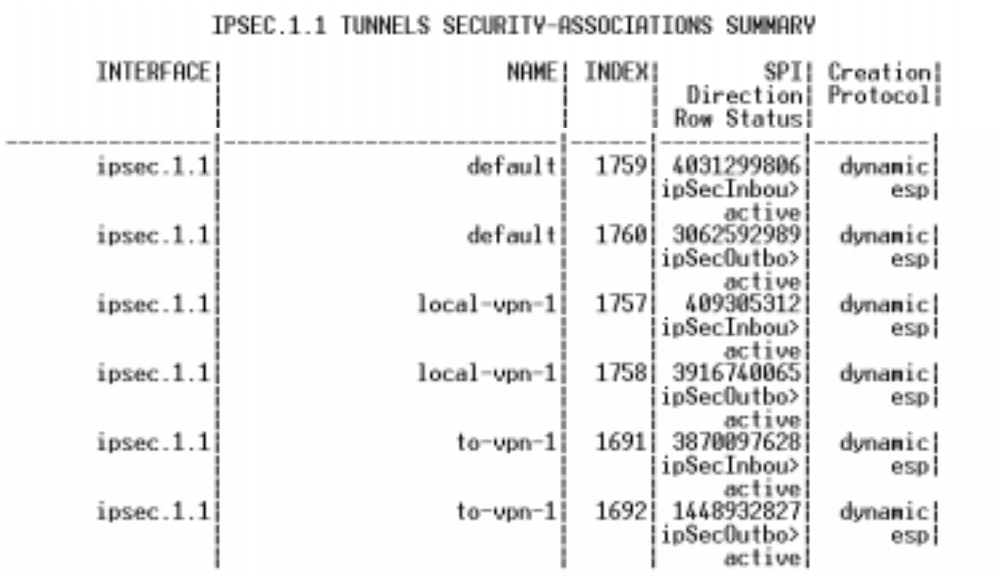
For dynamic tunnels, IKE dynamically creates and deletes the tunnel SAs during tunnel negotiation. To view tunnel SAs, use this command:

```
> show ipsec.1.1 tunnel.* security-associations summary
```

In the summary screen shown in Figure 54, the wildcard (*) is used to display all tunnels associated with ipsec.1.1. Note that each tunnel is dynamic, in an active state, and that each has a direction of inbound (ipSecInbound) or out-

bound (ipSecOutbound).

Figure 54 IPSec Security Associations Summary



IPSEC.1.1 TUNNELS SECURITY-ASSOCIATIONS SUMMARY

INTERFACE	NAME	INDEX	SPI Direction Row Status	Creation Protocol
ipsec.1.1	default	1759	4031299806 ipSecInbou> active	dynamic esp
ipsec.1.1	default	1760	3062592989 ipSecOutbo> active	dynamic esp
ipsec.1.1	local-vpn-1	1757	409305312 ipSecInbou> active	dynamic esp
ipsec.1.1	local-vpn-1	1758	3916740065 ipSecOutbo> active	dynamic esp
ipsec.1.1	to-vpn-1	1691	3870097628 ipSecInbou> active	dynamic esp
ipsec.1.1	to-vpn-1	1692	1448932827 ipSecOutbo> active	dynamic esp

DISPLAYING GENERAL SUBINTERFACE INFORMATION

To display information on the tunnel framework associated with a subinterface, use this command:

```
> show ipsec.1.1 vpn
```

The screen that appears displays the tunnel type, the remote gateway at the far end of the tunnel, and the preshared key in use by ISAKMP for authentication. Information on current and total tunnels is provided, along with error and discard counters.

Figure 55 IPSec Subinterface Summary

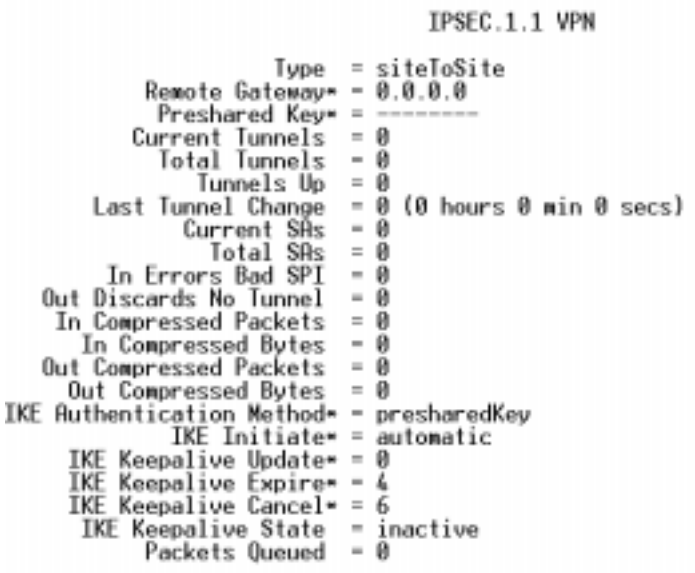


Table 24 provides detailed information on some of the objects shown in Figure 55.

Table 24 IPSec Subinterface Objects

VPN Subinterface Object	Definition
Last Tunnel Change	The value of sysUpTime; the last time a tunnel on the interface changed status (ipsecTunnelOperStatus). This object indicates to the manager when it needs to repoll for new tunnel configuration and status.
Current SAs	Gives the number of currently configured SAs. Note that during dynamic tunnel rekeying there may be 0 to 4 SAs on a tunnel since IKE removes the current SAs and adds new SAs.
Total SAs	Provides the total number of security associations (SAs) for all tunnels created on the interface since the interface was configured. For interfaces with dynamic tunnels, this object gives indicates how many times IKE rekeying events have occurred. Every time a dynamic tunnel successfully rekeys, new inbound and outbound SAs are created and this object is incremented by 2.
In Errors Bad SPI	The total number of IPSec packets received which contained an invalid SPI (Security Parameter Index) and therefore couldn't be associated with a tunnel nor a specific interface.

Out Discards No Tunnel	This counter indicates the number of packets that could not be sent out on a tunnel, owing to a failure of the tunnel lookup operation. This value should be 0. Any other value indicates that a tunnel that matches a packet cannot be found within the subinterface specified.
In Compressed Packets	The total number of compressed packets received.
In Compressed Bytes	The total number of bytes after decompressing the compressed packets received.
Out Compressed Packets	The total number of compressed packets transmitted.
Out Compressed Bytes	The total number of bytes after decompressing the compressed packets sent.
Ike Authentication Method	Specifies the authentication method used by the gateway to establish dynamic IKE tunnel connections.
Ike Initiate	Specifies when IKE initiates dynamic, site-to-site tunnel establishment with the remote gateway.
Packets Queued	The total number of packets currently queued for all dynamic, site-to-site tunnels created on the interface.

GETTING INFORMATION ON TUNNEL COUNTERS

Counters can provide insight on how much traffic is flowing into and out of tunnels, and can be useful in diagnosing problems. Use this command to get detailed information on tunnel counters:

```
> show ipsec.1.1 tunnels counters
```

The command issued here would display information on all tunnels associated with the specified subinterface.

USING THE IKE KEEPALIVE PROTOCOL

The IKE Keepalive protocol (Keepalive) is a proprietary mechanism designed to function between Access Point QVPN gateways in a site-to-site (AP-to-AP) or client (AP-to-Lucent IPSec Client) tunnel framework. The client Keepalive is used only with the Lucent IPSec Client. In essence, IKE Keepalive provides two capabilities: the ability to detect loss of connectivity between the router addresses that sit at each side of an IPSec subinterface, and a way to bring the tunnel framework back up automatically when loss of connection occurs.

Proactively detecting the loss of connectivity is extremely beneficial because the down-state of a tunnel can otherwise be detected only when data cannot be forwarded through to a cooperating gateway. Another benefit of Keepalive is the protocol's ability to reestablish tunnels that have gone down. This makes it possible to use Keepalive in conjunction with the Virtual Router Redundancy Protocol (VRRP) to create redundant site-to-site tunnel configurations.

HOW KEEPALIVE WORKS

Keepalive determines whether a remote gateway is reachable by sending update messages to it at regular (configurable) intervals. When a response is not received after a specified number of updates, Keepalive assumes the gateway is unreachable. In this event, the protocol places its IPSec subinterface in a down and then a testing state, and attempts to reestablish a tunnel connection (only in the site-to-site tunnel configuration).

When running with Keepalive enabled, polling begins as soon as an IKE session is established. When a response to an update message is not received in a specified interval (defined by the Keepalive expire interval), the gateway brings down the IPSec subinterface and attempts to bring up tunnels. The events that occur when a poll is not received on time are as follows:

1. The gateway first brings the IPSec subinterface down briefly, then places the subinterface into testing mode. All tunnels associated with the subinterface immediately go down. The static route associated with the remote gateway's router address is disabled.
2. Next, the gateway attempts to reestablish a tunnel configured on the subinterface. The gateway continues with this effort every 0-15 seconds. This interval is controlled by the retry variable.
3. When an IKE session is established for the tunnel, Keepalive puts the subinterface into an Up state. All tunnels come up, and Keepalive polling

resumes.

CONFIGURING IKE KEEPALIVE

You must configure Keepalive separately on the IPSec subinterfaces of local and remote gateways. Since the protocol is proprietary, it is only supported between Access Point QVPN gateways. Configuring Keepalive involves setting three parameters. A sample configuration is shown here:

```
> config ipsec.1.1 ike-keepalive-update 100 ike-keepalive-expire 2
  ike-keepalive-cancel 5
```

This example specifies that Keepalive updates are sent to the remote QVPN gateway associated with ipsec.1.1 every 100 seconds, and that Keepalive will bring down the ipsec.1.1 interface if two updates are not responded to. Further, the command specifies that Keepalive will send five updates to the remote gateway when an IKE session is established, and disable Keepalive if none of these updates receives a response from the remote gateway.

You can show information on Keepalive parameters with this command.

```
> show ipsec.1.1 vpn
```

```
                                IPSEC.1.1 VPN

                    Type  = siteToSite
          Remote Gateway  = 192.1.1.1
           Preshared Key  = --------
          Current Tunnels = 0
            Total Tunnels = 2
              Tunnels Up  = 0
       Last Tunnel Change = 27427272 (3 days 4 hours 11 min 12 secs)
             Current SAs  = 0
               Total SAs  = 2
        In Errors Bad SPI = 0
   Out Discards No Tunnel = 0
     In Compressed Packets = 0
       In Compressed Bytes = 0
   Out Compressed Packets = 0
     Out Compressed Bytes = 0
IKE Authentication Method = presharedKey
            IKE Initiate = automatic
     IKE Keepalive Update = 0
     IKE Keepalive Expire = 0
     IKE Keepalive Cancel = 0
      IKE Keepalive State = inactive
           Packets Queued = 0
```

In the screen shown here, the *ike-keepalive-update* timer is set to zero and means that Keepalive is disabled. The state of the IKE Keepalive is also displayed. The following section explains Keepalive parameters in detail.

ABOUT IKE KEEPALIVE PARAMETERS

The *ike-keepalive-cancel* parameter is used to disable Keepalive in the event the remote gateway is not running this protocol, or does not support it. The parameter's setting specifies how many Keepalive requests to send (during session establishment), without receiving a reply, before disabling Keepalive.

In the example shown here, three Keepalive polls will be transmitted to the remote gateway on ipsec.1.2.

```
> config ipsec.1.2 ike-keepalive-update 30 ike-keepalive-expire 6
  ike-keepalive-cancel 3
```

If no response is received, the AP automatically disables Keepalive on this subinterface. This parameter accepts values from 2 - 10. The default value is 6.

The *ike-keepalive-update* parameter specifies the frequency of Keepalive update messages, in seconds. By default, this parameter's value is 0 (that is, Keepalive is disabled). When Keepalive is enabled, the minimum and maximum values are 10 and 3600 (1 hour) respectively.

In the example shown above, Keepalive updates are sent every 30 seconds.

The *ike-keepalive-expir*e counter specifies how many unanswered Keepalive updates to send, before determining that the remote gateway is down. This parameter accepts values from 2 -10. The default value is 4.

In the example shown above, the AP declares the remote gateway to be down after six unanswered updates have been sent.

CONFIGURING IPSEC TRANSPORT MODE CONNECTIONS

IPSec Transport Mode is used to secure IP communications between two systems by using IPSec tunneling for traffic flowing through the gateway to IP systems behind the gateway.

Setting up IPSec Transport Mode for site-to-site or client connections is almost identical to setting up site-to-site or client tunnels respectively. The following sections describe how to configure your Transport Mode connections.

CONFIGURING IPSEC SITE-TO-SITE TRANSPORT MODE CONNECTIONS

For site-to-site connections, add and configure IPSec transport interfaces instead of IPSec subinterfaces. You still designate a router address, select or configure an IPSec security profile, and configure an authentication method.

Setting up an IPSec site-to-site Transport Mode connection with a remote system involves the following steps.

1. Designate a router address for the local Access Point QVPN gateway.
2. Choose or configure IPSec security profiles.
3. Configure a transport interface and its transport connections.

DESIGNATING A ROUTER ADDRESS

The router address is used as the local VPN endpoint. It is one of the gateway's IP addresses, and is used by the Internet Key Exchange (IKE) protocol and Public Key Infrastructure (PKI) certificates to uniquely identify the local VPN gateway. Refer to "Designating a Router Address" on Page 393 for more information.

CONFIGURING IPSEC SECURITY PROFILES

A security profile is a set of security attributes used to describe one or more IPSec tunnel or transport connections. Security profiles determine key security aspects of tunnels to which they apply. Refer to "Configuring IPSec Security Profiles" on Page 394 for more information.

CONFIGURING IPSEC TRANSPORT INTERFACES

The transport interface is a virtual interface with a remote device upon which one or more transport connections exist. Typically, you would configure the

authentication method (preshared key or digital certificates) for the transport interface.

For example, to configure a dedicated IPSec transport interface with a remote device (19.20.21.22) using a preshared key for authentication:

```
> add ipsec transport-interface.19.20.21.22 \
  ike-authent-method presharedKey preshared-key lemmein
```

To display the transport interface configuration for this remote device:

```
> show ipsec transport-interface.19.20.21.22 summary

                IPSEC TRANSPORT-INTERFACE.19.20.21.22 SUMMARY


                  Type   = siteToSite    IKE Keepalive Update* = 0
                  Mode*  = up            IKE Keepalive Expire* = 4
                 State   = down          IKE Keepalive Cancel* = 6
        IKE Initiate*    = automatic      IKE Keepalive State  = inactive
IKE Authent Method*      = presharedKey           Row Status*  = active
     Preshared Key*      = --------
```

CONFIGURING TRANSPORT CONNECTIONS

To configure a transport connection for L2TP traffic for the transport interface, specify UDP (17) as the protocol and L2TP (1701) for the local and remote ports as shown in this example.

```
> add ipsec transport-interface.19.20.21.22 transport.l2tp \
  type siteToSiteDynamic protocol 17 local-port 1701 remote-port 1701 \
  security-profile default-strong
```

NOTE *The AP only supports IPSec Transport Mode for L2TP traffic in this release, so protocol must be UDP (17) or any (0) and local port must be L2TP (1701) or any (0).*

There is no limit to the number of transport connections that can be established on a transport interface between two devices. However, the protocol, local port, and remote port must be unique as IKE uses these values to identify the connection.

DISPLAYING INFORMATION ABOUT TRANSPORT CONNECTIONS

To display the transport connection configuration for the remote device 19.20.21.22:

```
> show ipsec transport-interface.19.20.21.22 transport.l2tp summary
```

```
IPSEC TRANSPORT-INTERFACE.19.20.21.22 TRANSPORT.L2TP SUMMARY

                    Type* = siteToSiteDynamic
                    Mode* = up
                   State  = initializing
       Last State Change  = 33512 (0 hours 5 min 35 secs)
             Remote Addr  = 19.20.21.22
                Protocol* = 17
              Local Port* = 1701
             Remote Port* = 1701
          Security Profile* = default-strong
IKE Authentication Method  = none
              Row Status* = active
```

To display the IPSec transport counters:

```
> show ipsec transport-interface.198.202.232.23 counters
```

```
> show ipsec transport-interface.198.202.232.23 transport.l2tp counters
```

CONFIGURING STATIC TRANSPORT CONNECTIONS

By default, the transport connection type is dynamic (siteToSiteDynamic).

To use a static transport connection, you must specify siteToSiteStatic as the type and configure static security associations. To specify the type, use a command like the following:

```
> add ipsec transport-interface.198.202.232.23 transport.l2tp \
  type siteToSiteStatic security-profile default-strong
```

Dynamic connections rely on IKE to set up inbound and outbound security associations (SAs) for each connection. Static connections require administrators to create SAs manually by providing SA instance numbers for each direction. Each SA requires the following information to create security for the connection:

- A direction (either inbound or outbound)
- Security Parameter Index (SPI) for each direction
- An authentication key
- An encryption key

The SPI is a unique SA identifier in the packet header. The inbound SA on one machine and the outbound SA on another must have the same SPI, encryption key, and authentication key. The encryption and authentication keys are keyed

in as bytes of information. The number of bytes required depends upon the encryption transform used by the connections.

To configure static SAs, you might use commands resembling the following:

```
> add ipsec transport-interface.198.202.232.23 transport.l2tp \
  security-association.45 protocol esp spi 300 direction ipSecinbound \
  authentication-key
  263748596874475647462525252546576978576970584736 encryption-key
  343457687983457485746353827363454547493030595734444
```

```
> add ipsec transport-interface.198.202.232.23 transport.l2tp \
  security-association.46 protocol esp spi 303 direction ipSecoutbound \
  authentication-key
  263748596874475647462525252546576978576970584736 encryption-key
  343457687983457485746353827363454547493030595734444
```

To display the configured security associations for the remote device 198.202.232.23:

```
> show ipsec transport-interface.198.202.232.23 transport.l2tp \
  security-associations summary
```

CONFIGURING IPSEC CLIENT TRANSPORT MODE CONNECTIONS

For client connections, add and configure client groups. You still designate a router address, select or configure an IPSec security profile, and configure an authentication method.

To set up an IPSec client Transport Mode connection with a remote system, follow these steps.

1. Designate a router address for the local Access Point QVPN gateway.
2. Choose or configure IPSec security profiles.
3. Configure the transport interface and client groups.

DESIGNATING A ROUTER ADDRESS

The router address is one of the gateway's IP addresses, and is used by the Internet Key Exchange (IKE) protocol and Public Key Infrastructure (PKI) certificates to uniquely identify the local VPN gateway. Refer to "Designating a Router Address" on Page 393 for more information.

CONFIGURING IPSEC SECURITY PROFILES

A security profile is a set of security attributes used to describe one or more IPSec tunnel or transport connections. Security profiles determine key security aspects of tunnels to which they apply. Refer to "Configuring IPSec Security Profiles" on Page 394 for more information.

CONFIGURING IPSEC TRANSPORT INTERFACES

The transport interface is a virtual interface with a remote device upon which one or more transport connections exist. When a client initiates a connection, the gateway attempts to authenticate it using whatever method the client proposes.

Clients using IPSec Transport Mode typically have a dynamic IP address and therefore connect to the IPSec transport-interface.0.0.0.0 (which means any IP address). Client IKE connections can authenticate with PKI certificates or with a common preshared key.

For example, to configure the preshared key for the client group transport:

```
> config ipsec transport-interface.0.0.0.0 preshared-key "clientPresharedKey"
```

CONFIGURING CLIENT GROUPS

When a client attempts to connect, the gateway finds the best matching client group transport based on the upper layer protocol and ports of the connection. Once a client group is selected, the gateway attempts to negotiate the security policy associated with the client group. If the negotiation is successful, a client transport is created beneath the appropriate client group transport.

NOTE *The AP only supports IPSec Transport Mode for L2TP traffic in this release, so protocol must be UDP (17) or any (0) and local port must be L2TP (1701) or any (0).*

For example, to enable Windows 2000 VPN connections, a client group transport is configured with protocol 17 (UDP) and well known remote port 1701 (L2TP). The security policy is defined by the specified IPSec security policy (in this case, *default-client*).

```
> add ipsec transport-interface.0.0.0.0 transport.win2k-l2tp type clientGroup \
  protocol 17 remote-port 1701 security-profile default-client
```

For more information about configuring security profiles, see Page 394.

DISPLAYING INFORMATION ABOUT THE CLIENT GROUP TRANSPORT

To display information about the win2k-l2tp client group transport configuration:

```
> show ipsec transport-interface.0.0.0.0 transport.win2k-l2tp summary
```

```
IPSEC TRANSPORT-INTERFACE.0.0.0.0 TRANSPORT.WIN2K-L2TP SUMMARY

                    Type* = clientGroup
                    Mode* = up
                   State  = up
          Last State Change  = 1027839 (2 hours 51 min 18 secs)
                 Remote Addr  = 0.0.0.0
                    Protocol* = 17
                  Local Port* = 0
                 Remote Port* = 1701
               Security Profile* = default-client
IKE Authentication Method  = none
                 Row Status* = active
```

DISPLAYING INFORMATION ABOUT THE CLIENT TRANSPORT SESSIONS

When a client successfully connects to the gateway using these traffic selectors and the IPSec security profile is accepted, a client transport is created beneath the client group.

To display information about all the client transport sessions for the win2k-l2tp client group:

```
> show ipsec transport-interface.0.0.0.0 transport.win2k-l2tp.* summary
```

```
IPSEC TRANSPORT-INTERFACE.0.0.0.0 TRANSPORT.WIN2k-L2TP.* SUMMARY

REMOTE ADDR|                    NAME|                    Type|
           |                        |                    Mode|
           |                        |                   State|
           |                        |      Last State Change|
           |                        |             Remote Addr|
           |                        |                Protocol|
           |                        |              Local Port|
           |                        |             Remote Port|
           |                        |          Security Profile|
           |                        |  IKE Authentication Method|
           |                        |              Row Status|
-----------|------------------------|-------------------------|
   0.0.0.0|win2k-l2tp.19.20.21.22|                   client|
           |                        |                       up|
           |                        |                       up|
           |                        |  99059 (00:16:30.59)|
           |                        |          198.202.232.117|
           |                        |                       17|
           |                        |                        0|
           |                        |                     1701|
           |                        |          default-client|
           |                        |              presharedKey|
           |                        |                   active|
```

Note that the client transport session name is based on the group name and its IKE Phase 1 identity. Typically, most clients will use their IP address as their identity. Clients can use their X.509 Distinguished Name if using certificate-based authentication.

To view the counters for this particular client transport session:

```
> show ipsec transport-interface.0.0.0.0 transport.l2tp2.19.20.21.22 counters
```

```
IPSEC TRANSPORT-INTERFACE.0.0.0.0 TRANSPORT.L2TP2.19.20.21.22 COUNTERS

          Current SAs  = 2                    Out Bytes  = 121030360
            Total SAs  = 24            Out Discards State  = 0
          In Packets   = 121088        Out Discards No SA  = 0
            In Bytes   = 121030283   Out Discards Que Full  = 0
  In Discards Que Full = 0           In Compressed Packets  = 0
     In Errors Authent = 0             In Compressed Bytes  = 0
In Errors Bad Sequence = 0          Out Compressed Packets  = 0
  In Errors Bad Format = 0            Out Compressed Bytes  = 0
           Out Packets = 121019
```

CONFIGURING L2TP TUNNELS

The Layer 2 Tunneling Protocol (L2TP) provides a way to tunnel PPP packets across public networks. This enables remote users to access and use remote resources as if directly connected to them.

The AP supports L2TP as described in the IETF standard. Within the L2TP framework, the AP functions as an L2TP Network Server, or LNS. At present, the AP supports inbound calls via UDP transport. These are addressed to UDP port number 1701.

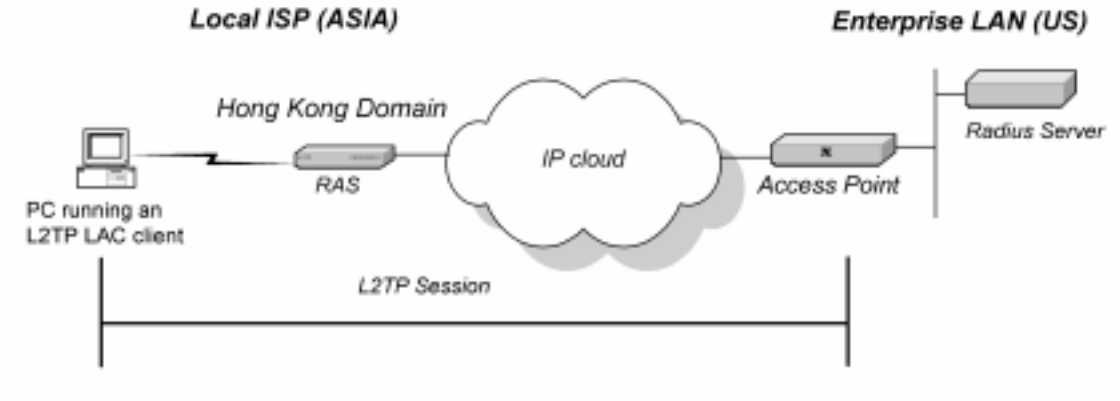
ABOUT L2TP

L2TP is an IETF standard that specifies a way to tunnel PPP connections from remote users across a public network infrastructure. This allows remote users to establish a modem connection with a local ISP, rather than incurring long-distance charges.

L2TP works by terminating layer 2 (PPP) connections at a local L2TP Access Concentrator (LAC). Following tunnel and session establishment with a remote L2TP Network Server (LNS), PPP packets are tunneled to the LNS and forwarded to their destination. This gives the connecting user a low-cost virtual connection to the remote network. Moreover, the mechanics of tunnel and session negotiation and authentication (PAP, CHAP, or RADIUS protocols are supported) are completely transparent to the end user and applications using L2TP.

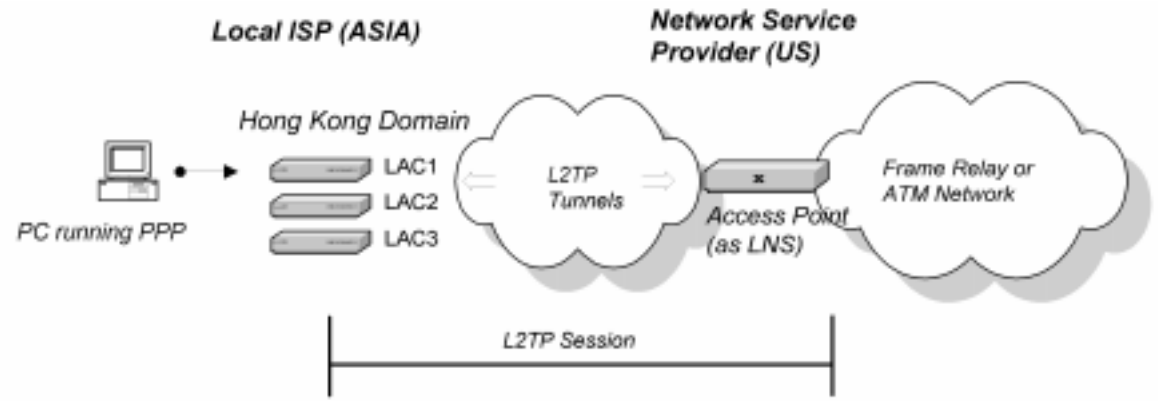
Figure 56 shows a typical use of L2TP. In this scenario, a PC running an L2TP LAC client in Hong Kong connects via modem to a local ISP. The connection passes through a RAS in the Hong Kong domain, and a tunnel connection is established with the LNS functionality in a remote AP. After being authenticated, the Windows client can access enterprise LAN resources (in the US) as if locally connected to them. The only phone charge is that of the initial call to the ISP in Hong Kong.

Figure 56 L2TP Connection: Enterprise LAN with L2TP LAC Client



In contrast to the Enterprise scenario, Figure 57 shows how L2TP can be used by a network service provider to deliver packets from remote users to systems on the far side of an ATM or FR network. Conceptually and functionally, this scenario differs little from that shown in Figure 56. The key distinction is that the PC uses standard PPP to connect to the local ISP. The L2TP tunnel and session are created at the ISP.

Figure 57 L2TP Connection: ISP Service



In this scenario, PPP packets are tunneled to the LNS, then forwarded over the wide area via Frame Relay or ATM. When configuring L2TP, address pools and routes can be chosen so that specific sessions map to specific DLCIs or ATM VCs.

How L2TP Works

You can think of L2TP as a way of extending the PPP termination point from the local connection point (the LAC) across a public network to a remote point (the LNS). End-user PPP packets are tunneled across the low-cost public infrastructure in a dedicated L2TP session. To the user making the call, the mechanics of call conversion and packet delivery are transparent. From this user's perspective, the call to a local ISP is functionally the same as a direct connection to a system on the other side of the planet. All data sent is delivered, and responses returned, as if a direct connection existed between the user and the remote system.

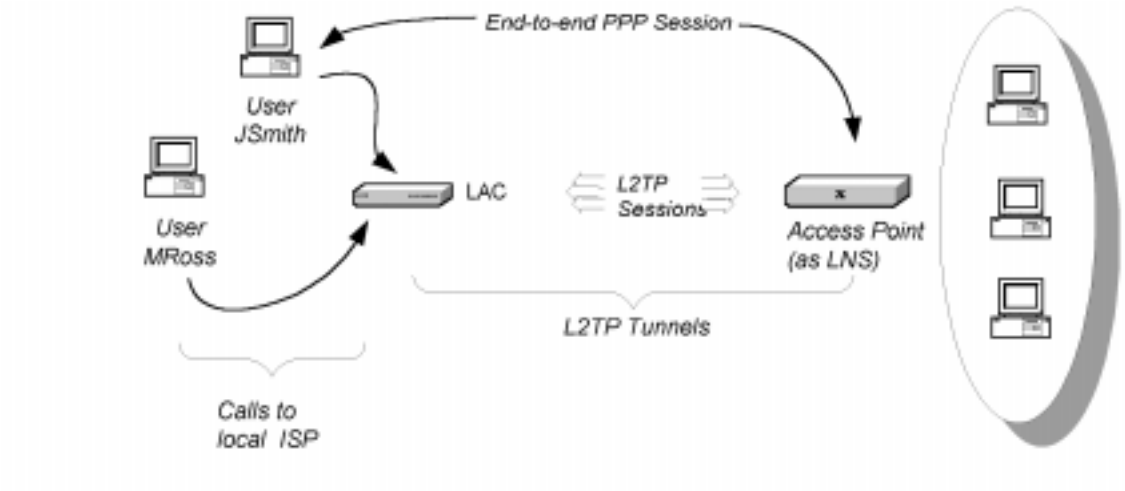
Connection Establishment

L2TP is a connection-oriented protocol used to establish and maintain sessions between L2TP peers. In this context, L2TP peers — a LAC and an LNS — act as the two endpoints of the tunnel framework. Upon receiving a call from an end system, a LAC establishes a connection to an LNS. Once this connection is established, PPP frames received at the LAC are stripped of link framing, CRC and transparency bytes and encapsulated in an L2TP header. They are then dispatched into an appropriate tunnel.

Packet Delivery

When the LNS receives L2TP packets, it strips off the L2TP header and processes the resulting PPP packets as if they had been received on a local interface. These packets can then be delivered on the enterprise LAN or forwarded over an ATM or FR network to their ultimate destination.

Figure 58 L2TP Operations: Tunnel and Session Establishment



Once this connection is established, an L2TP tunnel and session are established between the LAC and LNS. The PPP packets associated with each call are delivered through the tunnel within a specific session, which corresponds to a dedicated PPP stream between the LAC and LNS.

Once the call is complete, the tunnel created for the session can be torn down or maintained, depending on how the LAC is administered. A single tunnel can contain many sessions, each of which corresponds to a single call to the LAC. More than one tunnel can exist between the same LAC and LNS pair, and multiple LACs can maintain tunnels to the same LNS.

L2TP IN THE VIRTUAL TUNNEL TRANSPORT

The AP uses a specific Virtual Tunnel Transport stack for L2TP, as it does for IPSec and GRE tunneling capabilities. The L2TP stack contains three components for support of L2TP. These components are l2tp.1, l2tp.1.1, and l2dial.1.1. You do not have to add these components. They appear by default when you boot the AP. Figure 59 shows the L2TP stack as it appears prior to tunnel and session establishment.

Figure 59 L2TP Components in the Virtual Stack

```
40¦          virtual  Virtual Tunnel Transport   enabled      up ¦
41¦          virtual   cbq.10                    enabled      up ¦
42¦          virtual    l2tp.1                   enabled      up ¦
43¦          virtual     l2tp.1.1                enabled      up ¦
44¦          virtual      l2dial.1.1             enabled      up ¦
45¦          virtual       ip.11                 enabled      up ¦
```

L2TP Session Establishment and Stack Changes

Changes to the L2TP stack occur dynamically as tunnels are set up and sessions established by L2TP. Figure 60 shows the virtual stack for L2TP with a single L2TP tunnel and session.

Whenever a tunnel and session are established or torn down, the AP stack changes. For example, the tunnel and session shown in Figure 60 are associated with three layers. These layers (shown at right) are sandwiched between the l2tp.1.1 and l2dial.1.1 layers.

L2TP tunnel instance → l2tptnl.1
L2TP tunnel uplink → l2tptnl.1.1
Tunneled PPP session → ppp.1

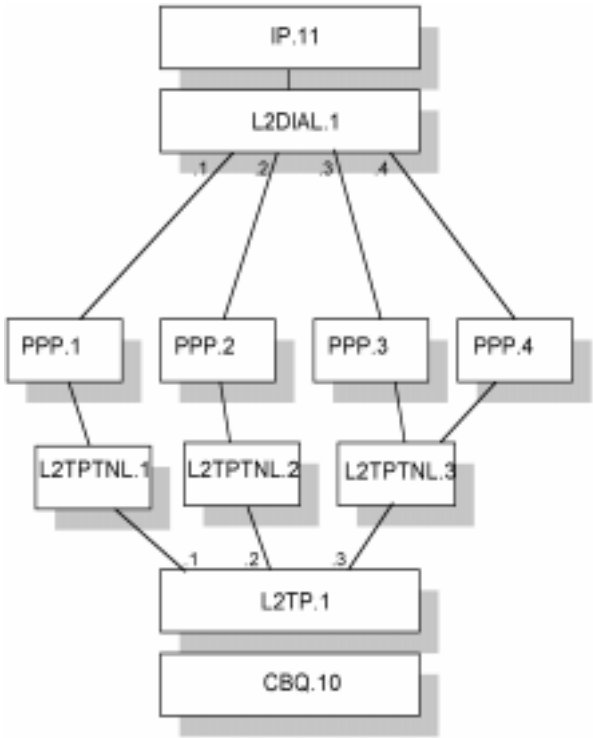
Figure 60 L2TP Stack: Single Tunnel and Session

```
40¦          virtual  Virtual Tunnel Transport   enabled      up ¦
41¦          virtual   cbq.10                    enabled      up ¦
42¦          virtual    l2tp.1                   enabled      up ¦
43¦          virtual     l2tp.1.1                enabled      up ¦
44¦          virtual      l2tptnl.1              enabled      up ¦
45¦          virtual       l2tptnl.1.1           enabled      up ¦
46¦          virtual        ppp.1                enabled      up ¦
47¦          virtual         l2dial.1.1          enabled      up ¦
48¦          virtual          ip.11              enabled      up ¦
```

The numerics associated with each of the stack components identify the uplinks and downlinks that exist between adjacent layers. This is the same in the case of L2TP as it is with any other layer type supported by the AP.

About the L2TP Stack

The figure at right provides a diagram of a typical L2TP stack, with emphasis on the uplink and downlink connections. Note that a single instance of L2DIAL.1 and L2TP.1 exist at the top and bottom of the stack. Figure 61 shows the actual stack from which this diagram derives.



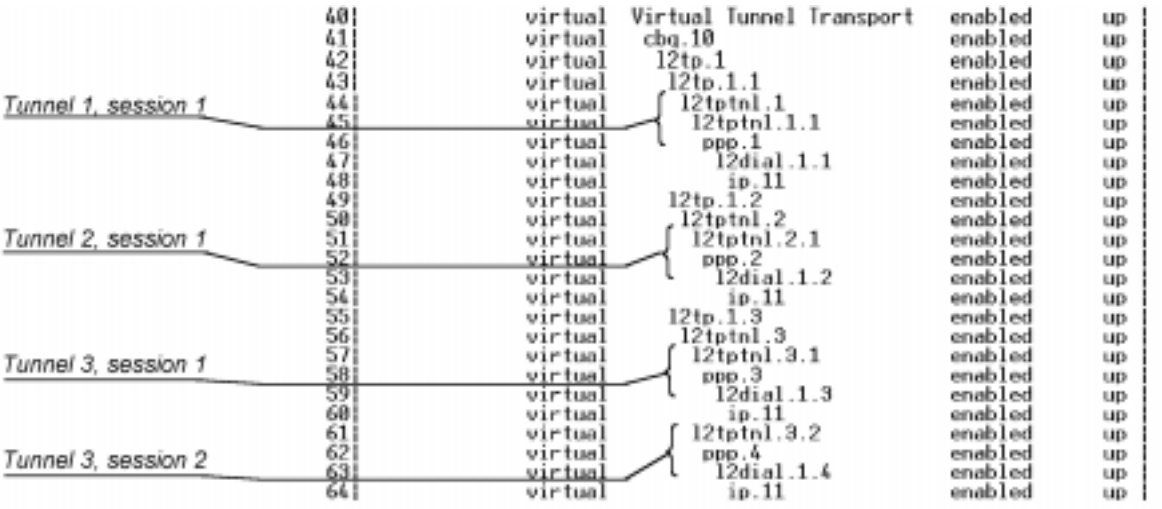
Within L2TP, the L2DIAL layer functions as an aggregation mechanism for the PPP and PPP-MP instances that connect to it via downlinks. In the diagram shown here, these downlinks are L2DIAL.1.1 through L2DIAL.1.4. The first two tunnels present (L2TPTNL.1 and L2TPTNL.2) each have one uplink, while the third tunnel (L2TPTNL.3) has two more, making a total of four active L2TP sessions.

Support for Multiple L2TP Tunnels

Each call to a LAC produces a single L2TP session, and each L2TP session is associated with a unique 3-layer group like the one shown here. More than one tunnel can exist between the same LAC-LNS pair, and many L2TP sessions — each carrying the PPP packets for a single call — can exist in a single tunnel.

Even when multiple tunnels are present, the virtual stack provides a clear map of what is happening. Figure 61 shows the stack on which the preceding figure is based. This stack has three different tunnels. One of the tunnels (labeled Tunnel 3) has two separate L2TP sessions, associated with downlinks l2dial.1.3 and l2dial.1.4.

Figure 61 L2TP Stack with Tunnels



SUPPORT FOR DIAL-IN ROUTERS

When configuring clients, the clients can be routers or hosts (end nodes). By default, an end node is the dial-in client and the sessions are set up as shown in Figure 61. However, if you have a session that supports a dial-in router instead of an end node, the stack looks different because the dial-in router is not associated with an aggregation layer (such as l2dial.1.4) and it has its own IP interface (for example, ip.12 instead of ip.11).

For example, you might see the following addition to the bottom of Figure 61 if you add a routed connection to that stack:

```
l2tp.1.4
  l2tptnl.4
    l2tptnl.4.1
      ppp.5
        ip.12
```

CONFIGURING L2TP

For the AP to function as an LNS, all L2TP layers must be up as shown in Figure 59, and an IP interface (also in the up state) must exist at the top of the L2TP stack.

Check to be sure that the mode for each layer is enabled

You can do this by issuing a *status* command. The default for each layer is enabled.

Configure the IP LAN interface behind the LNS

This is the IP address associated with the Ethernet interface.

Configure the IP interface above L2TP

Add an IP address to the IP instance above the L2TP. In this example, the IP instance is IP.11 and it is unnumbered.

```
> config ip.11 address.0.0.0.0 src-addr 128.10.1.2
```

The source address of IP.11 is associated with the AP's LAN interface. In this case, the source address (128.10.1.2) is the LAN interface of the AP. A numbered IP interface can also be used.

Configure L2TP tunnel domains

Configure tunnel domains so you can set up tunnel authentication and other specific tunnel configuration information. See Page 432.

Configure dial-in users

Configure user authentication, client groups, and address pools. See Page 437 for instructions.

CONFIGURING L2TP TUNNEL DOMAINS

The AP supports a table for storage of configuration parameters associated with specific tunnel domains. Remote host names in the form *host@domain-name* are received from the LAC during tunnel establishment, and compared to names stored in the tunnel domain table.

When a match is found, the AP uses the associated set of tunnel parameters to initialize the tunnel. A match can be to the entire remote host name, or just the domain in cases where the name is of the form:

host@domain

The AP applies domain configuration parameters during tunnel initialization to all tunnels with LACs in this domain.

For tunnels that do not have an assigned domain configuration, the AP uses the parameters associated with the (default) domain configuration for tunnel establishment. The default tunnel parameters are:

```
                L2TP.1 DOMAIN-CONFIGURATION.(DEFAULT)

          Admin State* = enabled      Control Max Retrans* = 5
       Authentication* = none     Control Retrans Timeout* = 16
               Secret* = --------      Payload Sequencing* = false
      Tunnel Security* = none       Reassembly Timeout* = 0
Tunnel Hello Interval* = 60      Proxy Authentication* = false
  Tunnel Idle Timeout* = 0                      Status* = active
          Control RWS* = 30
```

You can configure tunnel authentication for domains so that the AP challenges the LAC during tunnel establishment using a shared secret configured locally or via RADIUS. The AP does not challenge the LAC if authentication is set to none.

You can configure user authentication for domains so that the LAC is the proxy for authentication as part of proxy LCP.

The AP also provides status information on a per-domain basis, making it possible to retrieve information on tunnels in a particular domain.

CONFIGURING A TUNNEL CHALLENGE SECRET

Configuring a challenge secret for a tunnel domain enables the AP to authenticate tunnels while establishing them.

This command shows how to configure a domain called kansas to add a chal-

lenge secret (ffffff) for tunnel authentication and increase the *tunnel-idle-timeout* value to 60.

```
> config l2tp.1 domain-configuration.kansas authentication challenge secret
  ffffff tunnel-idle-timeout 60
```

These settings indicate that the AP will challenge the LAC for a shared secret during session establishment. The AP does not challenge the LAC if authentication is set to none. The AP can respond to a challenge by the LAC provided a shared secret has been configured.

CONFIGURING A RADIUS TUNNEL CHALLENGE

Configuring a RADIUS challenge lets the AP authenticate tunnels with a shared secret provided by RADIUS at L2TP tunnel startup.

This command shows how to configure a domain called kansas to add a RADIUS challenge secret (hidenseek) for tunnel authentication.

```
> config l2tp.1 domain-configuration.kansas \
  authentication radius-challenge secret hidenseek
```

Note that you must still configure the AP to use a RADIUS server for authentication. Refer to "Configuring RADIUS Authentication" on Page 507 for more information.

When a RADIUS challenge is configured for the tunnel domain, attributes stored in the RADIUS database are returned and used for authentication during L2TP tunnel startup.

CONFIGURING A RADIUS USER ENTRY FOR L2TP TUNNELS

To authenticate L2TP tunnels, configure the RADIUS user entry to include the following attributes with these values:

```
username = <LAC's name>
Password = <user-auth-passwd>
Tunnel-Password = <password used for L2TP tunnel startup>
Service-Type = Outbound-User
```

NOTE

The value for the Password attribute must be the same value as the radius-challenge secret provided for tunnel domain configuration.

You must include the Service-Type = Outbound-User attribute in your RADIUS entry or the tunnel will not come up.

AUTHENTICATING TUNNEL DOMAINS WITH RADIUS

RADIUS can be used to store shared secrets used during tunnel establishment. The secret is known by the RADIUS server and the remote tunnel endpoint. When an AP needs the secret, it queries the RADIUS server and uses the Tunnel-Password attribute value (that is returned) for tunnel establishment.

A RADIUS client (the AP) can only retrieve attributes from the RADIUS server on a successful user authentication. To use RADIUS to store these shared secrets, a user must be configured in the RADIUS database. The username is that of the remote tunnel endpoint. The password is configured on the AP (a password known to the AP and RADIUS server). It is this password that RADIUS authenticates, and upon success, RADIUS returns the rest of the user's attributes. The Tunnel-Password is the attribute containing the shared secret that the tunnel endpoints must both know. Service-Type must be set to Outbound-User for user entries that are used for tunnels.

A number of accounting attributes are also supported, and can be included when configuring the RADIUS server's user entry.

tunnel-client-endpoint

The LAC's IP address or fully-qualified domain name.

tunnel-server-endpoint

The LNS's IP address or fully-qualified domain name.

tunnel-client-auth-ID

The LAC's name used during tunnel authentication.

tunnel-server-auth-ID

The LNS's name used during tunnel authentication.

RADIUS servers that correctly support the Tunnel-Password attribute will encrypt the password before sending the access-accept message. The AP will decrypt the string, without determining if the server is compliant, and use the result as the tunnel password.

Servers can encrypt the Tunnel-Password when it is entered in the database, or when preparing to send the access-accept message. Depending on when a compliant server performs this encryption, the Tunnel-Password can be stored in the clear on the RADIUS server. This is implementation-dependent.

CONFIGURING USER PROXY AUTHENTICATION

The AP will only use the proxy authentication information received from the L2TP peer when the proxy Link Control Protocol (LCP) negotiation is successful and the L2TP domain allows proxy authentication.

PROXY LCP

When setting up user connections between L2TP peers, the connecting L2TP peer (LAC) can provide proxy LCP and proxy authentication information. The AP will always try to use the proxy LCP information received from the LAC. If the AP does not support the options proposed on its behalf, the AP restarts LCP.

CONFIGURING PROXY AUTHENTICATION

To configure the L2TP tunnel domain called kansas with proxy authentication:

```
> config l2tp.1 domain-configuration.kansas proxy-authentication true
```

Remember that the AP will only use the proxy authentication information received from the L2TP peer when the proxy LCP negotiation is successful.

SHOWING L2TP INFORMATION

The AP includes a rich set of commands that enable you to monitor individual sessions and tunnels, and derive information on LACs connecting to the AP. You can also display tunnel domain configuration information.

DISPLAYING INFORMATION ON TUNNEL DOMAINS

Use this command to show information on the l2tp10_lac domain:

```
> show l2tp.1 domain-configuration.l2tp10_lac
```

```
                    L2TP.1 DOMAIN-CONFIGURATION.L2TP10_LAC

          Admin State* = enabled       Control Max Retrans* = 5
       Authentication* = none      Control Retrans Timeout* = 30
              Secret* = --------       Payload Sequencing* = false
     Tunnel Security* = none           Reassembly Timeout* = 0
Tunnel Hello Interval* = 3600        Proxy Authentication* = false
  Tunnel Idle Timeout* = 0                          Status* = active
         Control RWS* = 30
```

You can display information on all L2TP domains with this command:

```
> show l2tp.1 domain-configurations
```

DISPLAYING INFORMATION ON TUNNELS

Use this command to display summary information on L2TP tunnels.

```
> show l2tptnl.* tunnel-configuration
```

L2TPTNL.* TUNNEL-CONFIGURATION

INTERFACE	Authentication Hello Interval Control Retransmissions Delayed Ack Timeout	Secret Idle Timeout Payload Sequencing Reassembly Timeout	Security Control RWS Payload RWS Transport
l2tptnl.1	none 2600 5 100	"" 0 false 0	none 30 30 ipUdp
l2tptnl.2	none 10 5 100	"" 0 false 0	none 30 30 ipUdp

To display information on tunnels, issue this command:

```
> show l2tptnl.* tunnel-status summary
```

L2TPTNL.* TUNNEL-STATUS SUMMARY

INTERFACE	Local TID	Remote TID	State	Remote Host Name
l2tptnl.1	1	33	estab>	NONE
l2tptnl.2	2	222	estab>	Ascend_LAC.xedia>
l2tptnl.3	3	47	estab>	iVasionVPN
l2tptnl.4	4	3	estab>	Cisco_LAC

Issue this command to display information on L2TP sessions:

```
> show l2tptnl.1.* session-statuses summary
```

L2TPTNL.1.* SESSION-STATUSES SUMMARY

INTERFACE	LOCAL ID	Datalink	User Name	State
l2tptnl.1	1	ppp.1	""	estab>
l2tptnl.1	2	ppp.2	""	estab>
l2tptnl.1	3	ppp.3	""	estab>
l2tptnl.1	4	ppp.4	""	estab>
l2tptnl.1	5	ppp.5	""	estab>
l2tptnl.1	6	ppp.6	""	estab>
l2tptnl.1	7	ppp.7	""	estab>
l2tptnl.1	8	ppp.8	""	estab>
l2tptnl.1	9	ppp.9	""	estab>
l2tptnl.1	10	ppp.10	""	estab>
l2tptnl.1	11	ppp.11	""	estab>
l2tptnl.1	12	ppp.12	""	estab>

AUTHENTICATING DIAL-IN USERS

The AP authentication framework enables clients to be authenticated locally or by a RADIUS server. This framework is based on three concepts:

- *Users*

  Individual hosts or routers that are configured with a specific authentication method and belong to an explicitly configured group or the (default) group.

- *Client Groups*

  Uniquely named groups that specify a node type (either endnode or router), an address assignment scheme, address pool, and PPP echo interval for a bunch of users.

- *Address Pools*

  Defined pools that specify the range of IP addresses from which the AP selects an IP address for a specific client.

Configuring address pools, client groups, and users is part of the overall configuration for dial-in users. For more details, see Page 439 for information about configuring user authentication, see Page 441 for information about configuring address pools, and Page 443 for information about configuring client groups.

HOW IT WORKS

For dial-in users, all these components are interrelated. Configuring user authentication lets you include users in a certain client group. Client groups specify various parameters, including address pools, for all users in that client group. The address pool specifies the IP address range from which users in a certain client group can receive an IP address.

For example, use this command to display the client groups associated with all users:

```
> show services authentication user-table
```

```
                    SERVICES AUTHENTICATION USER-TABLE

NAME                Password                Client Group            Authentication Method
                    Ssh Privileges          Row Status
--------------------------------------------------------------------------------
(default)           --------                (default)               radiusAuthentication
                    noAccess                active
```

In this case, there is only the (default) user and this user is associated with the (default) client group. As the display below shows, members of the (default) client group belong to the (default) address pool.

```
> show services client-group.(default)

                    SERVICES CLIENT-GROUP.(DEFAULT)

   Address Assign* = acceptClient+internalPool
     Address Pool* = (default)
             Type* = endNode
PPP Echo Interval* = 30
           Status* = active
```

As a result, all users in (default) client group receive an IP address from the (default) pool when connecting to the AP. The following display shows all defined address pools.

```
> show ip address-pool configurations

                    IP ADDRESS-POOL CONFIGURATIONS

            NAME|        Network|           Mask| Row Status|
----------------|---------------|---------------|-----------|
       (default)|        5.5.5.0|  255.255.255.0|     active|
           pool|      123.4.5.0|  255.255.255.0|     active|
          pool1|        1.1.1.0|  255.255.255.0|     active|
```

AUTHENTICATING CLIENTS LOCALLY

For local authentication, the AP retrieves attributes from the user table and client group table. The user table contains the client group and authentication method. For clients that belong to a group that is locally authenticated, the process is fairly straightforward. The AP determines what method to use to assign the client an address (either acceptClient or internalPool).

In a typical client scenario, when a user connects, the AP matches the user to a client group. This group's configuration determines the address assignment scheme, address pool, node type, and PPP echo interval for users belonging to it.

If the client group specifies a defined address pool (as is usually the case), the AP either chooses an address from this pool (if internalPool is specified for the group) or accepts the IP address (if acceptClient is specified, and the connecting client has an IP address). If the client group belongs to the (default) address pool, and internalPool is specified, the AP assigns an address from this pool.

AUTHENTICATING CLIENTS VIA RADIUS

When RADIUS authentication is used, attributes stored in the RADIUS database are returned and used to authenticate the connecting user, and perform other functions.

Figure 62 Sample RADIUS Entry

```
dave1

Password = “hello_1”,
Xedia-DNS-Server = “44.55.66.77”,
Xedia-NetBios-Server = “66.77.88.99”,
Tunnel-Private-Group-ID = “group1”,
Framed-IP-Address = “11.22.33.44”,
XEDIA-ADDRESS-POOL = “pool1”,
XEDIA-PPP-ECHO-INTERVAL = 62,
Framed-Route = “3.3.3.0 1.2.3.40“
```

When RADIUS is the authentication method selected for a user, information in the RADIUS database has preference over information stored elsewhere. Notice that the information for dave1 shown in Figure 62 includes a framed IP address, framed route, and the Xedia address pool. In this case, the AP applies this information to the connection attempt.

CONFIGURING USER AUTHENTICATION

You can configure user authentication in two ways:

- Using information stored locally in the user table
- Using information stored on a RADIUS server

To configure users in the authentication user table, assign a username/password pair, a client group, and an authentication method with this command:

```
> config services authentication user.<name> password <string> \
  client-group <group name> authentication-method <method>
```

Refer to “Configuring Users” on Page 90 for more information.

The client group specifies the group to which a user belongs, and identifies a subnet and mask that the AP uses to assign the user an enterprise IP address. See Page 443 for more information about configuring client groups.

CONFIGURING USERS TO BE AUTHENTICATED BY RADIUS

Sites can store authentication information for each user on a designated RADIUS server. When a user connects, the AP contacts the RADIUS server for authentication information.

To specify RADIUS authentication for a particular user, use this command:

```
> config services authentication user.<user> \
  authentication-method radiusAuthentication
```

You can find information on configuring username/passwords and tunnel names for RADIUS in your RADIUS server documentation.

With respect to the RADIUS server, the AP functions as a client. To use RADIUS authentication, you must configure the AP as a RADIUS client, and add at least one RADIUS server configuration record. See “Configuring RADIUS Authentication” on Page 507 for more information.

CONFIGURING USERS TO BE AUTHENTICATED LOCALLY

Many sites store username/password information for each user on the AP. The name of a specific client group — indicating the user’s membership in the group — is also stored. When a user connects, the AP can authenticate the user and assign the connection to a specific client group without consulting any other server.

To add a username/password pair and specify client group membership:

```
> add services authentication user.<user> password <string> \
  client-group <client-group-name>
```

The username and password chosen must be unique on the AP. The client-group-name chosen determines the subnet and mask from which the AP will chose an enterprise address for the packet. For instructions on adding client groups, see “Adding Client Groups” on Page 406.

ENABLING USER PASSWORD AUTHENTICATION

To authenticate connections, enable user password authentication for the interface. To determine the mode, use this command:

```
> show services authentication summary
```

The authentication summary screen provides a great deal of information. The first portion of the screen is shown below. The first line shows the password

authentication for IPSec are presently enabled.

```
                    SERVICES AUTHENTICATION SUMMARY

Enable User Password Authentication* = ipsec
              Accounting Log Control* = logStart+logStop
    Accounting Configured List Size* = 200
       Accounting Current List Size  = 200
           Accounting Current Index  = 0
                     Users Configured = 9
               Authentication Q Curr  = 0
                Authentication Q Max  = 1
Expected Authentication Queue Depth   = 0
               Authentication Enqueues = 3
               Authentication Dequeues = 3
```

If PPP is not listed, add it by issuing this command:

```
> config services authentication enable-user-password-authentication ppp
```

You can also remove a protocol (like IPSec) from the list as follows:

```
> config services authentication enable-user-password-authentication !ipsec
```

If you choose to disable user password authentication for IPSec or PPP, dial-in clients will not be prompted for name and password information. See the following section for more information.

DISABLING USER PASSWORD AUTHENTICATION

IPSec and PPP need not be enabled for communication to occur between connecting clients and the AP. If you choose to disable user password authentication, extended authentication and configuration will not occur. In this case, packets sent and received by clients will continue to be encrypted at each end of the tunnel. However, users will not be authenticated by the AP. Clients connecting to the AP appear as an IP address, rather than as an authenticated username/password pair.

CONFIGURING ADDRESS POOLS

Address pools define a range of addresses from which the AP can assign an IP address to connecting clients. To define an address pool, proceed as follows. This sample command defines a pool called *pool*:

```
> config ip address-pool configuration.pool network 123.4.5.0 mask
  255.255.255.0
```

Use this command to show information on all address pools:

```
> show ip address-pool configurations

                    IP ADDRESS-POOL CONFIGURATIONS

            NAME|        Network|           Mask| Row Status|
----------------|---------------|---------------|-----------|
       (default)|        5.5.5.0|  255.255.255.0|     active|
           pool|      123.4.5.0|  255.255.255.0|     active|
          pool1|        1.1.1.0|  255.255.255.0|     active|
          pool2|        2.2.2.0|  255.255.255.0|     active|
      susanaddr|        3.0.0.0|      255.0.0.0|     active|
```

USING THE DEFAULT ADDRESS POOL

If the connecting user is unknown, no authentication is performed on the connection. In this case, the connecting host receives an address from the default pool.

The default network and mask for the default pool are 0.0.0.0 and 255.255.255.0. However, you can configure this pool with a specific network and mask as you can for any other pool. If you elect *not* to do this, remote hosts will be assigned an address from the 0.0.0.0 network at connection time.

MODIFYING THE DEFAULT ADDRESS POOL

Notice that in the display shown above, the default pool name is in parentheses. To modify any aspect of the default pool, you must use the parentheses in specifying the pool name. For instance, to change the network and mask for the pool, you would enter a command like this one:

```
> config ip address-pool configuration.(default) network 3.3.3.3 mask
  255.255.255.0
```

Leaving the parentheses out when modifying the default pool will create a pool called default. This is not the default pool, which is named (default).

SHOWING ADDRESS POOL ALLOCATIONS

You can show the IP addresses that have been allocated from a specific pool, or from all pools. To show the allocations for all pools, use this command:

```
> show ip address-pool allocations

                        IP ADDRESS-POOL ALLOCATIONS

              NAME|      IP ADDRESS|           Layer|        User Name|
------------------|----------------|----------------|-----------------|
         (default)|  167.100.10.105|           ppp.5|warren_1@xedia.c>|
         (default)|    192.168.11.2|           ppp.4|         warren_1|
       lucent.com|  192.169.11.193|           ppp.2|warren_1@lucent.>|
       lucent.com|  192.169.11.194|           ppp.3|warren_2@lucent.>|
        xedia.com|  168.192.11.193|           ppp.1|warren_3@xedia.c>|
```

CONFIGURING CLIENT GROUPS

If you are supporting large numbers of users, you can configure users as members of a defined client group. Once you define a group, all users assigned to it use the same configuration. This makes it unnecessary to configure each user individually.

Each client group has the following associations.

- Address pool (pool name)
- Type (router or endnode)
- Address assignment scheme (acceptClient or internalPool)
- PPP echo interval

For example, you can define a client group named *from-ISP3* as follows:

```
> config services client-group.from-isp3 address-assign acceptClient type
  endNode
```

You can show information added for this group with this command:

```
> show services client-group.from-isp3

                        SERVICES CLIENT-GROUP.FROM-ISP3

    Address Assign* = acceptClient+internalPool
      Address Pool* = (default)
              Type* = endNode
PPP Echo Interval* = 3
            Status* = active
```

The associations provided for each client group specify how the AP handles connections from group members. Notice that no address pool has been specified when configuring from-ISP3. In this case, the AP uses the default address

pool, as it does for connections from users not assigned to any group. See Page 441 for instructions on configuring address pools. Each piece of information specified for a client group is discussed below.

ADDRESS ASSIGN

Address assign specifies the IP address assignment method for the group. Both acceptClient and internalPool are default values. If a connecting client has an IP address (and both default values are specified for the group), the AP accepts the incoming IP address rather than selecting one from the internal pool. If the client has no IP address, the AP assigns an address from the internal pool.

RADIUS provides another way for an IP address to be assigned to a client. If RADIUS is the defined authentication method for a connecting user (or for the default user), then both authentication and address assignment are performed via RADIUS. RADIUS authentication always supersedes local methods for authenticating a client.

ADDRESS POOL

The address pool specifies the name of the local pool from which the AP selects an IP address for clients belonging to a specific group. You can configure a number of distinct address pools, as well as a default pool for connecting users who do not belong to any group. See Page 441 for instructions on configuring address pools.

TYPE

Supported types are *endnode* (the default) or *router*. When configuring a client group, the type attribute determines whether group members are routers or hosts (end nodes).

The *endnode* type indicates that each session goes to an aggregation layer (such as l2dial) and thus to the same IP interface. The *router* type indicates that each session goes to a separate IP interface.

RADIUS provides another way to designate a client as a router or end node. If a framed route is provided by RADIUS, then the client is a router.

PPP ECHO INTERVAL

The echo interval functions as a keepalive mechanism for PPP sessions. Settings range from 0 - 3600 seconds. The default value is 3. In effect, setting the echo interval to 0 turns this mechanism off. In this case, PPP sessions would stay up indefinitely, whether or not connectivity with a peer exists. See “Con-

figuring PPP" on Page 168 for more information on configuring PPP.

STATUS

The status of the defined group. By default, all defined groups are active.

ADDING USERS TO A CLIENT GROUP

To add users to a client group, proceed as follows:

```
> config services authentication user.chris client-group fella
```

To show users in the user table, use this command.

```
> show services authentication user-table
```

```
                                SERVICES AUTHENTICATION USER-TABLE

                NAME|        Password|      Client Group|  Authentication Method|
                    |  Ssh Privileges|        Row Status|
--------------------|----------------|------------------|-----------------------|
           (default)|        --------|         (default)|   radiusAuthentication|
                    |        noAccess|            active|
                bill|        --------|         (default)|           alwaysAccept|
                    |        sftpRead|            active|
               chris|        --------|           fella  |internalAuthentication|
                    |        sftpRead|            active|
           davidc_new|        --------|         (default)|internalAuthentication|
                    |        sftpRead|            active|
           davidc_old|        --------|         (default)|internalAuthentication|
                    |         session|            active|
                 jim|        --------|         (default)|internalAuthentication|
                    |        noAccess|            active|
                mike|        --------|         (default)|internalAuthentication|
                    |        noAccess|            active|
               ralph|        --------|         (default)|   radiusAuthentication|
                    |        noAccess|            active|
```

CONFIGURING PKI

Public Key Infrastructure (PKI) is a service that can be used to provide authentication and confidentiality. The system generates PKI key pairs consisting of a public and a private key. The system uses its private key to encrypt or sign, and never discloses the private key. The system shares its public key with anyone who needs to communicate with it. The remote system uses the public key to decode a signature or encrypted data.

The AP supports the use of certificates and Certificate Revocation Lists (CRLs).

CONFIGURING PKI CERTIFICATES

Public keys are communicated between systems in the form of certificates. Certificates contain the device's or the user's identity and public key, along with the identity and signature of the Certificate Authority (CA) who issued the certificate.

To authenticate connections with digital certificates, perform these tasks:

1. Configure the system's subject name.
2. Generate a PKI public/private key pair.
3. Configure the certificate request and send it to the CA.
4. Generate the certificate and load it.

CONFIGURE THE SYSTEM'S SUBJECT NAME

The AP needs a subject distinguished name to identify itself to the CA and other systems. The AP includes this name with certificate requests it generates, or that are generated by CAs. The format of the subject distinguished name is an X.500 directory name.

By default, the subject distinguished name is automatically set from the system's router address. The router address is used in the Common Name field (CN) of an X.500 Distinguished Name.

To display the subject name, use this command:

```
> show services pki subject-name
```

To set the subject name manually, use this command:

```
> config services pki subject-name-format !distNameFromRtrAddr \
  subject-distinguished-name "<subject name>"
```

where <subject name> is the subject distinguished name in X.500 directory name format. For example:

```
> config services pki subject-name-format !distNameFromRtrAddr \
  subject-distinguished-name "C=US, O=Xedia, CN=AccessPoint-10.10.10.1"
```

GENERATING PKI KEY PAIRS

Next, generate a key pair with this command:

```
> add services pki key-pair.<name> algorithm [rsa|dsa] length <number>
```

where <name> is the key pair's administrative name and <number> is specified as 512 - 2048 bits when using RSA or 512 - 1024 bits when using DSA. The default length in either case is 1024 bits. For example:

```
> add services pki key-pair.rsa-sig algorithm rsa length 1024
```

This command specifies the key pair named rsa-sig will be used to generate RSA signatures that are 1024 bits long.

Key pair generation can take a few seconds or a few minutes. To check the status of key pair generation:

```
> show services pki key-pair.<name> summary
```

When key pair generation is complete, the Row Status field should indicate active.

CONFIGURING THE CERTIFICATE REQUEST

When a key pair is generated, an associated certificate request is automatically generated with the same name. To display the certificate request:

```
> show services pki certificate-request.<name> summary
```

where <name> is the name of the PKI key pair. By default, the certificate request uses the SHA-1 hash algorithm to generate the signature. The signature algorithm is sha1WithRSAEncryption for an RSA key pair or sha1WithDSAId for a DSA key pair. To change the signature algorithm, use this command:

```
> config services pki certificate-request.<name> \
  signature-algorithm <algorithm>
```

where <name> is the name of the PKI key pair and <algorithm> is the signature algorithm (md5WithRSAEncryption, sha1WithRSAEncryption, or sha1WithDSAId).

Once the certificate request is complete, the PEM-encoded PKCS#10 request is ready to be sent to the CA. To display the encoded request:

```
> show services pki certificate-request.<name> output
```

How you transfer the certificate to the CA is a local policy matter that depends on your environment. The usual method is to paste the certificate into a browser window. Refer to the CA documentation for more information about recommended certificate transfer policies.

GENERATE AND LOAD THE CERTIFICATE

Once the CA has the gateway's certificate request, the CA has all the necessary information to generate a certificate for the gateway.

NOTE *If the gateway's certificate is used for IKE authentication, make sure the CA includes the gateway's router address in the certificate as the subject alternate name.*

When the X.509 certificate is generated, it must be loaded into the gateway. Again, how you transfer the certificate is a local policy matter that depends on your environment. Refer to the CA documentation for more information about recommended certificate transfer policies.

Load certificates using this command:

```
> config services pki new-certificate "<certificate>"
```

where <certificate> is the PEM-encoded X.509 certificate generated by a CA. If the certificate contains multiple lines, you must enclose the entire value within quotation marks ("). CA certificates are also loaded with this command.

When the certificate has been loaded successfully, an entry is added to the certificate table. To view the entries in the certificate table, use this command:

```
> show services pki certificates
```

The Key Pair field indicates that the certificate corresponds to the system's own key pair (named in this field). The device certificate remains untrusted until it can be verified with the certificate of the CA that signed it.

In many cases, all peer gateways and users are signed by the same CA. In this

instance, all devices must be loaded with at least the CA certificate and the device certificate. Any device or user presenting a certificate signed by this CA certificate can be verified against the CA's public key (which is part of its certificate).

When certificates are manually loaded, they are marked static and saved to nonvolatile memory. Certificates can also be learned. For example, when IKE tries to authenticate another device, it receives the remote device certificate and adds that certificate to the certificate table. Learned certificates are marked dynamic and are not saved to nonvolatile memory.

All certificates are untrusted when first loaded. When the administrator is satisfied that the CA certificate is legitimate, the CA certificate can be set to trusted using this command.

```
> config services pki certificate.<instance> force-trusted true
```

Once the CA certificate is trusted, the AP can verify the signature of its own certificate and mark it trusted as well.

Certificate hierarchies can be more than one level, and can involve more than one verification chain. The AP will authenticate a certificate presented by another device or user as long as it can find a path to a trusted CA certificate and verify the signatures. For more information about multi-level CA hierarchies and multiple certificate chains, consult your CA documentation.

With both the CA certificate and the device certificate loaded and trusted, the system is ready to perform PKI authentication. If IKE was configured to use certificate-based authentication, the remote gateway's certificate would be sent over in an IKE payload and added to the local certificate table as a dynamic certificate. The system would perform path verification and check the remote side's signature. If the signature is verified, the certificate would be marked trusted and IKE could bring up the connection.

CONFIGURING CERTIFICATE REVOCATION LISTS

Certificate Revocation Lists (CRLs) are generated by Certificate Authorities (CAs). These contain a list of certificates issued by the CA which have been revoked. Revoking a certificate denies a user the access a certificate is designed to provide. Certificates can be revoked for any number of reasons. For example, an enterprise CA might generate certificates with a one year validity period for all employees, and revoke certificates for employees terminated before that year has concluded.

CONFIGURING SYSTEM TIME

Certificates and CRLs include GMT dates and times to specify validity periods, revocation times, and other information. For this reason, it is important to configure the AP system time in GMT. Otherwise, timing problems may occur. It is possible, for example, for the AP to receive a new certificate, but determine that it is invalid because the system time does not fall within the certificate's GMT validity period.

For information on configuring system time, see "Configuring the System Name, Date and Time" on Page 46.

LOADING CRLS

You can load CRLs through Access View, the command line interface, or using SNMP. Like certificates, the system expects the CRL in PEM format. Since CRLs are updated frequently, entering CRLs manually is not recommended. However, you may need to do this if the LDAP server is down or unavailable.

```
> config services pki new-crl <pem encoded crl>
```

To show a summary of all CRLs, use this command:

```
> show services pki crls summary
```

```
                                SERVICES PKI CRLS SUMMARY

INDEX|          Issuer|          Number|     Type|          Updated|
     |     Next Update|    Trust Status| Creation|    Revoked Certs|
     |          Remove|                |         |                 |
-----|-----------------|----------------|---------|-----------------|
     |                 |                |         |                 |
   4|C-US, O-xedia, O>|              16|  x509_v2|1999 Oct 19th, 1>|
    |1999 Oct 20th, 1>|         trusted|  dynamic|                1|
    |           ready|                |         |                 |
```

In most cases, the AP fetches CRLs from a directory server via LDAP. To configure the AP to retrieve CRLs, configure the PKI LDAP server as shown here:

```
> config services pki ldap-server 192.100.55.1
```

Once configured with a server, the gateway automatically issues LDAP requests for CA CRLs when computing trust computation for subordinate certificates. Upon receiving a new CRL, the AP adds it to its database. The AP generates an INFO event like this one to indicate the CRL addition:

```
> # PKI     (21)crl_added          x10/18/1999 15:16:53:013 INFO

  New CRL #4 added for (C=US, O=xedia, OU=eng, CN=entrustca)
```

DISABLING CRLS

If a particular CA is not capable of issuing CRLs, or your policy is not to use them, disable the CRL Issuer field in the CA's certificate as shown here:

```
> config services pki certificate.1 crl-issuer false
```

When this field is disabled, the trust computation for certificates subordinate to the CA will not search for the CA's CRL. Disabling this field for non-CRL-issuers is important, since the AP assumes that all CAs are CRL issuers. A result of this assumption is that the AP does not trust subordinate certificates until it receives a valid, up-to-date CRL for the CA. The trust status of the certificate is "noIssuerCRL" for certificates that are untrusted because the AP has not received a valid CRL.

VIEWING CERTIFICATES AND CRLS

You can show information on PKI certificates using this command:

```
> show services pki certificates summary
```

SERVICES PKI CERTIFICATES SUMMARY

INDEX	Subject Type Not Valid After CRL Issuer Key Algorithm	Issuer Serial No Creation Trust Status Remove	Key Pair Not Valid Before CA Force Trusted
1	C=US, O=xedia, OU=en> x509_v3 2019 Oct 5th, 15:49> true md5WithRSAEncryption	C=US, O=xedia, O> 939138546 static trusted ready	"" 1999 Oct 5th, 15:19> true true
2	C=US, O=xedia, OU=en> x509_v3 2002 Oct 8th, 13:49> false sha1WithRSAEncryption	C=US, O=xedia, O> 939138574 static trusted ready	rsa 1999 Oct 8th, 13:19> false false
3	C=US, O=xedia, OU=en> x509_v3 2002 Oct 6th, 12:58> false sha1WithRSAEncryption	C=US, O=xedia, O> 939138564 static revoked ready	"" 1999 Oct 6th, 12:28> false false

You can view a CRL and any revoked certificates it contains using this command:

> show services pki crl.4 summary

```
                        SERVICES PKI CRL.4 SUMMARY

      Issuer  = C-US, O-xedia, OU-eng, CN-entrustca  Trust Status  = trusted
      Number  = 16                                      Creation  = dynamic
        Type  = x509_v2                           Revoked Certs  = 1
     Updated  = 1999 Oct 19th, 17:07:42 GMT            Remove*  = ready
Next Update  = 1999 Oct 20th, 17:49:11 GMT
```

When a certificate is placed on the CRL list, its trust status is revoked. The AP generates an INFO event like this one if a certificate is on a CRL list, or is untrusted for any reason.

> # PKI (17)cert_not_trusted x10/18/1999 15:15:47:025 INFO

Certificate #2 for (C=US, O=xedia, OU=eng, CN=entrustca, CN=10.10.10.11) is NO LONGER TRUSTED

To show information on certificates on a specific CRL, use this command:

> show services pki crl.4 certificates

```
                        SERVICES PKI CRL.4 CERTIFICATES

INDEX| INDEX|     Serial Number|          Revoked|          Invalid|
     |      |           Issuer |                 |                 |
-----|------|------------------|-----------------|-----------------|
   4|     1|         939138564|1999 Oct  8th, 1>|               ""|
     |      |                  |                 |                 |
```

10

Managing With Access View

Access View is the Web-based interface to the Access Point system (AP), providing a configuration tool you can use via your browser. This chapter contains an overview of Access View, procedures explaining how to use it, and common network management and configuration tasks illustrating its use.

Overview of Access View

Access View is a Java-based management interface that enables you to configure an AP from a Web browser. You can run Access View from the following browsers:

- Netscape Communicator V4.5.1 or higher
- Microsoft Internet Explorer V4.0 or higher

Other browsers may work as well, provided they support:

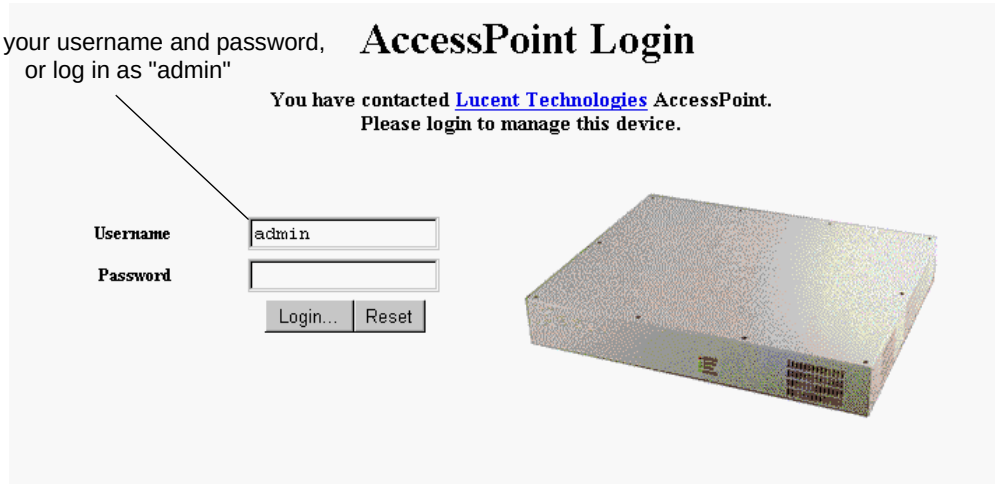
- Frames
- Java
- HTML V3.2
- Tables
- JavaScript
- Cookies

Unlike the command line interface (CLI), which provides access to specific pieces of information, Access View uses pages set up as forms. This allows you to see whole areas of the AP configuration *at a glance*, rather than by issuing specific *show* or *show summary* commands.

STARTING UP ACCESS VIEW

Access View is a built-in component of Access Point systems. To start up an Access View session, open a Web browser and enter the IP address of the AP. The login dialog appears, as shown here:

Figure 63 Access View Login Dialog Window



Next, enter a username and password that provides read/write access to the Access Point system. If you have not set up usernames for the AP, type *admin* as the username. Admin is the name of a default read/write SNMP community, which employs a NULL password.

Click *Login...* to display the Access View window shown in Figure 64.

ABOUT THE ACCESS VIEW WINDOW

As Figure 64 shows, Access View opens in a window with three frames, one for each of the following:

- *Management Navigator - provides links to manageable system and interface components*
- *Current Page - provides a work area where forms appear that you can modify to manage system and interface components*
- *Information Log - provides an area for system messages*

You can think of these windows as the application's work area. The sections that follow describe each in detail.

Figure 64 Access View Window

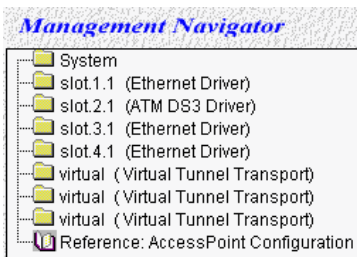


MANAGEMENT NAVIGATOR

The Management Navigator lists a tree of folders and its sub-folders and sub-entries (see Figure 65) that help you navigate within the Access Point system and its various slots. These folders include:

- The System folder that contains global system component information. This data includes interface network layer information, user interface, SNMP, and DNS configuration data, and bridging, IP, Frame Relay, and ATM information.
- The Slot folders that contain the configuration information pertaining to the cards in these slots. This data includes Class-Based Queuing (CBQ) and network layer information for that card's network.
- The Virtual folders that contain the configuration information for the Virtual Tunnel Transport stack for L2TP, as well as stacks for IPSec and GRE tunneling capabilities. This data includes configuring tunnels and security associations.

Figure 65 Management Navigator Folder Tree



GETTING HELP

On-line help is available for the System folder and in the Slot and Virtual folders for each particular instance in the book icon labeled Help. You can also get help on any Management Information Base (MIB) object that appears in the forms by clicking on the blue question mark that accompanies the object. (This is the equivalent of the CLI *describe* command.)

CURRENT PAGE

When you click on components in the Management Navigator folders, the associated page opens in the Current Page frame. Enter parameter changes and new configurations in this page.

INFORMATION LOG

This frame is a log for information that Access View is sending to the AP. It is not used for data entry or to display configuration information, but to display system messages.

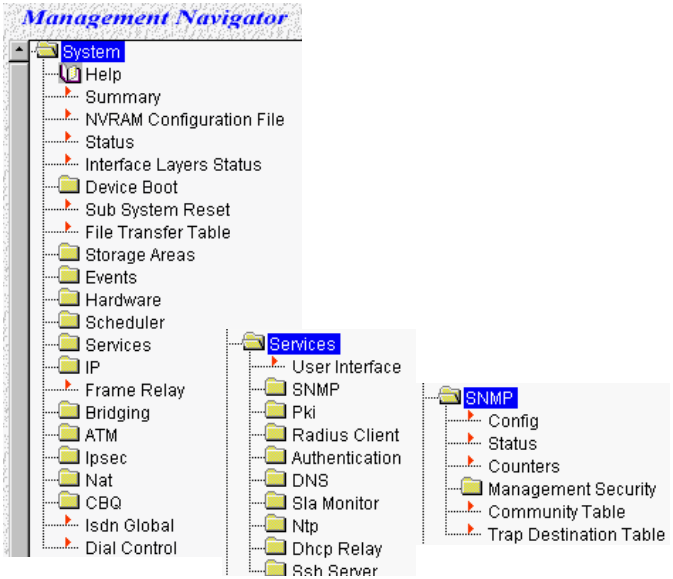
MANAGING THE ACCESS POINT SYSTEM

The remainder of this chapter provides a primer about Access View. While showing you how Access View works, it also steps you through a basic sequence of tasks that you can use to set up your AP. These tasks can be broken down into three broad areas, with sub-tasks:

- ☑ *Managing the Access Point system*
  - Assigning a system name
  - Setting the data and time
  - Setting up an SNMP community
  - Configuring the DNS client
- ☑ *Configuring WAN interfaces*
  - Configuring HSSI
  - Configuring Frame Relay
- ☑ *Configuring CBQ*
  - Configuring bandwidth
  - Configuring security

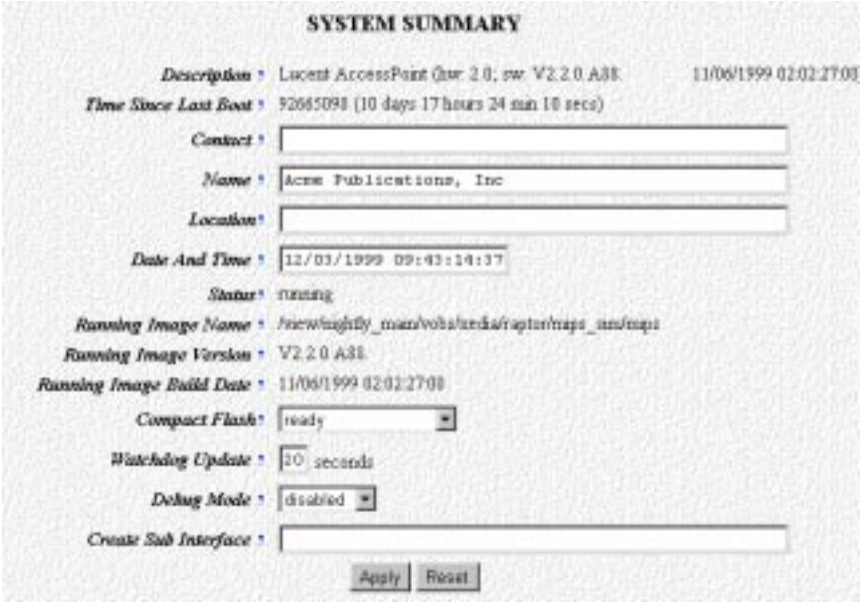
Subsequent sections assume that you have an in-depth, working knowledge of networking principles, protocols, and equipment.

The System folder provides access to global AP system component information rather than information about the interface cards. You can configure IP, ATM, bridging, Frame Relay, events, routing policy and set the system name, date, and time. In most cases you can accept the system defaults, but you should examine the settings to be sure they are what you want. The following illustration breaks out one branch of the System folder tree.



ASSIGNING A SYSTEM NAME AND SETTING THE DATE AND TIME

To assign the system name and set the date and time, select **System** → **Summary**. Enter the system name and date and time in the appropriate fields. Click *Apply*.

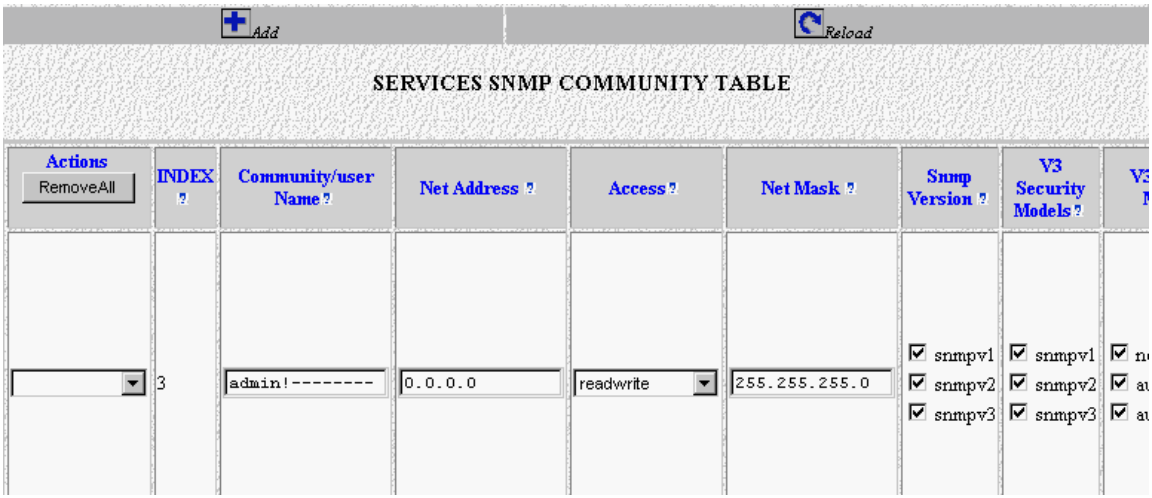


SETTING UP SNMP COMMUNITIES

SNMP allows you to secure your Access Point system by assigning groups of users and passwords, including the administrative community. Members of a designated administrative community can view and modify the SNMP Community table. Any community with read/write privileges can be the administrative community; however, there can be only one administrative community.

1 To set up an SNMP Community, select **System** → **Services** → **SNMP** → **Community Table**.

The SNMP Community Table contains multiple columns. For this reason, the following illustration presents only a portion of the table.



Click *Add*. Enter the Index. Click *Add* and *Done*.

In the Actions column for the Index you entered, select *View/Modify* in the drop down list. Enter the community name and password. Click *Apply* and *Done*.

2 To designate an administrative community, select **System** → **Services** → **SNMP** → **Config**.

Enter the community name and password for the designated administrative community. Click *Apply*.



CONFIGURING DNS

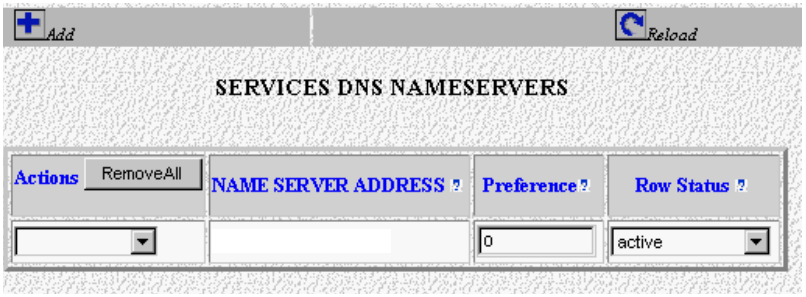
To begin configuring DNS, select **System** → **Services** → **DNS** → **Summary.** Enter your domain name in the domain field.



To configure your Access Point system to communicate with a DNS server,

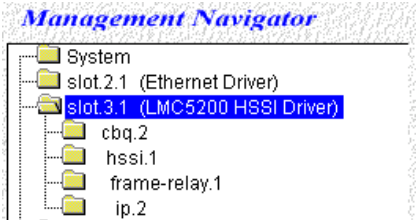
select **System → Services → DNS → Nameservers Table → Nameserver.**

Click *Add.* Enter the Name Server Address in dotted decimal format. Click *Add* and *Done*.



CONFIGURING WAN INTERFACES

In contrast to the System folder, the slot folders contain information about the interface cards installed in your AP. You can use this information to configure your interfaces for network communications. To configure the WAN interfaces for your Access Point system in this example, use the slot (labeled HSSI Driver) folders in the Management Navigator.

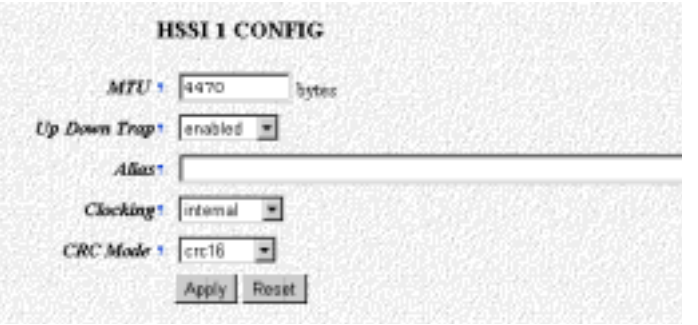


CONFIGURING HSSI

The default settings of the HSSI interface card installed in your Access Point system are likely sufficient for typical usage. Check the default HSSI configuration to see that it meets your needs.

To change the default HSSI settings, select **slot.*x.x*** → **hssi.*x*** → **Config**.

Make the changes you need and click *Apply.*

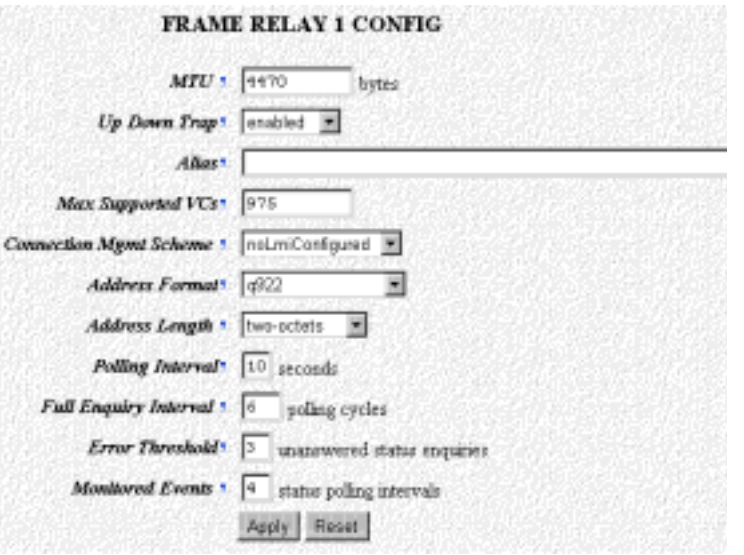


CONFIGURING FRAME RELAY

The frame-relay folder lets you manage your wide-area network connections. By default, the HSSI interface uses Frame Relay networking. A Frame Relay connection comes up automatically upon completion of the HSSI configuration. Check the default Frame Relay configuration to see that it suits your needs.

To change the default Frame Relay settings, select **slot.*x.x*** → **frame-relay.*x*** → **Config**.

If necessary, change the parameters and click *Apply.*



CONFIGURING CLASS-BASED QUEUING

CBQ lets you manage bandwidth by classifying and scheduling traffic. Moreover, CBQ can secure your network by means of firewalls, using techniques such as packet filtering and stateful classification of applications. Refer to Chapter 7, "Configuring CBQ for Traffic Management," for detailed information about CBQ and how it works.

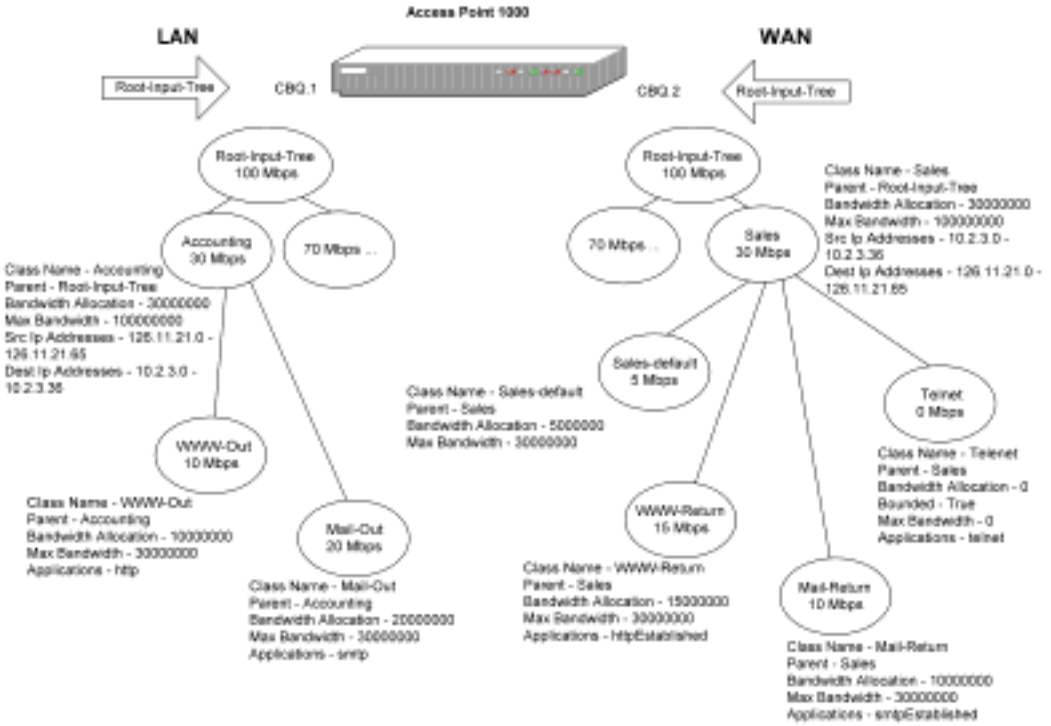
The AP in Figure 66 has two interfaces and two CBQ instances. The AP classifies inbound traffic from the LAN and WAN. To simplify this diagram and the ensuing example, only one branch from each root-input-tree is discussed — Accounting and Sales. Other branches are hidden in the boxes showing 70 Mbps and not discussed.

Accounting is a child of the root-input-tree on CBQ.1 and is allocated 30 Mbps of bandwidth, but can borrow up to 100 Mbps if available. Accounting specifies a range of source and destination addresses. Transmitted packets are matched against these addresses. If they don't match, they are dropped. Accounting has two children, WWW-Out and Mail-Out.

WWW-Out is half of a stateful pair for connections over the World Wide Web. It provides security by specifying http as the application and then pairing with an httpEstablished application, WWW-Return, on the root-input-tree of CBQ.2. WWW-Out allocates 10 Mbps, but can borrow more bandwidth if it's available.

Mail-Out also makes use of stateful classification by specifying the application smtp and pairing it with smtpEstablished (Mail-Return) on the root-input-tree of CBQ.2. It allocates 20 Mbps and can borrow more bandwidth if it's available.

Figure 66 CBQ Policies



Sales is a child of the root-input-tree on CBQ.2 and is allocated 30 Mbps of bandwidth, but can borrow up to 100 Mbps if available. Sales specifies a range of source and destination addresses. Transmitted packets are matched against these addresses. If they don't match, they are dropped. Sales has four children, Sales-default, Mail-Return, WWW-Return, and Telnet.

Sales-default is useful at initial setup to capture any traffic that does not meet the classification criteria you've established. Adding a default class guarantees that the AP passes all traffic through to its destination whether or not it matches classes you've defined. Later on, for greater security, you can remove the default classification. Sales-default allocates 5 Mbps and can borrow more bandwidth if it's available.

WWW-Return is the other half of the statefully classified pair previously discussed. By specifying an application type of httpEstablished, a relationship with the http input from the LAN is established and unwanted packets from the WAN are dropped. WWW-Return allocates 15 Mbps and can borrow more bandwidth if it's available.

Mail-Return participates in a stateful classification by specifying the application smtpEstablished and pairing with smtp input from the LAN. Mail-Return allocates 10 Mbps and can borrow more bandwidth if it's available.

The Telnet class illustrates another way of filtering packets coming from the WAN. Bandwidth is set to 0 and bounded, to true. Because Telnet has no bandwidth allocated to it and is not permitted to borrow idle bandwidth, all Telnet packets are dropped.

ADDING CBQ TRAFFIC CLASSES

The following example shows you how to set up the traffic classes using the CBQ folder hierarchy.

Adding traffic classes is the principal step in configuring CBQ. In the Management Navigator, select **slot.*x*.*x*** → **cbq.1** → **Traffic Classes** → **Add**.

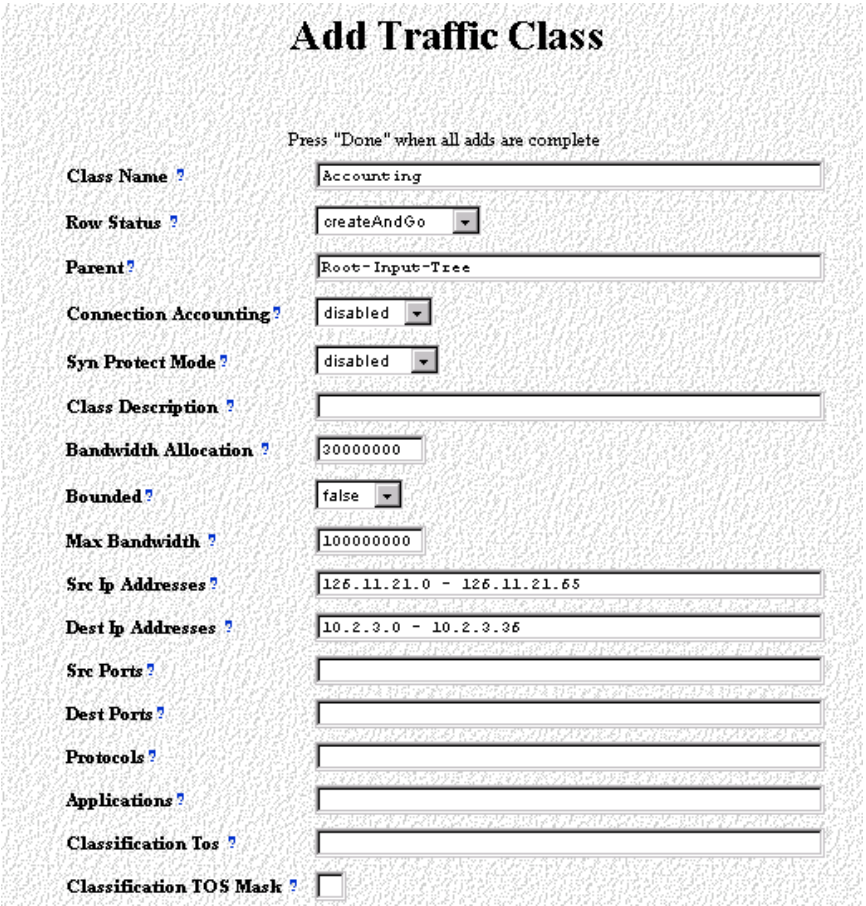
LAN-side root-input-tree

In the **Add Traffic Class** window, enter:

- Class Name - Accounting
- Parent - Root-Input-Tree
- Bandwidth Allocation - 30000000
- Max Bandwidth - 100000000
- Scr Ip Addresses 126.11.21.0 - 126.11.21.65
- Dest Ip Addresses 10.2.3.0 - 10.2.3.36

Click *Add...* and *Done.*

Figure 67 Add Traffic Class



In the Management Navigator, select **slot.*x.x*** → **cbq.1** → **Traffic Classes** → **Add.** In the **Add Traffic Class** window, enter:

- Class Name - WWW-Out
- Parent - Accounting
- Bandwidth Allocation - 10000000
- Max Bandwidth - 30000000
- Applications - http

Click *Add...* and *Done*.

In the Management Navigator, select **slot.*x.x*** → **cbq.1** → **Traffic Classes** → **Add.** In the **Add Traffic Class** window, enter:

- Class Name - Mail-Out
- Parent - Accounting
- Bandwidth Allocation - 20000000
- Max Bandwidth - 30000000
- Applications - smtp

.

WAN-side root-input-tree

In the Management Navigator, select **slot.*x.x*** → **cbq.2** → **Traffic Classes** → **Add.** In the **Add Traffic Class** window, enter:

- Class Name - Sales
- Parent - Root-Input-Tree
- Bandwidth Allocation - 30000000
- Max Bandwidth - 100000000
- Scr Ip Addresses 10.2.3.0 - 10.2.3.36
- Dest Ip Addresses 126.11.21.0 - 126.11.21.65

Click *Add...* and *Done.*

In the Management Navigator, select **slot.*x.x*** → **cbq.2** → **Traffic Classes** → **Add.** In the **Add Traffic Class** window, enter:

- Class Name - Sales-default
- Parent - Sales
- Bandwidth Allocation - 5000000
- Max Bandwidth - 30000000

Click *Add...* and *Done.*

In the Management Navigator, select **slot.*x.x*** → **cbq.2** → **Traffic Classes** → **Add.** In the **Add Traffic Class** window, enter:

- Class Name - WWW-Return
- Parent - Sales
- Bandwidth Allocation - 15000000
- Max Bandwidth - 30000000
- Applications - httpEstablished

Click *Add...* and *Done.*

In the Management Navigator, select **slot.*x.x*** → **cbq.2** → **Traffic Classes** → **Add.** In the **Add Traffic Class** window, enter:

- Class Name - Mail-Return
- Parent - Sales
- Bandwidth Allocation - 10000000
- Max Bandwidth - 30000000
- Applications - smtpEstablished

Click *Add...* and *Done.*

In the Management Navigator, select **slot.*x.x*** → **cbq.2** → **Traffic Classes** → **Add.** In the **Add Traffic Class** window, enter:

- Class Name - Telnet
- Parent - Sales
- Bandwidth Allocation - 0
- Max Bandwidth - 0
- Bounded - True
- Applications - telnet

Click *Add...* and *Done.*

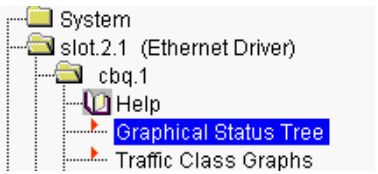
OVERVIEW OF THE TRAFFIC STATUS APPLICATION

The Traffic Status application is a single application comprising a Bandwidth Allocation and a Traffic Equalizer applet. These complementary applets provide administrators with three visualization tools: a graphical representation of the CBQ tree running on an Access Point system, a pie chart that shows the bandwidth allocated to each class as a percentage of its parent, and bar charted information on the actual bandwidth in use by selected classes. The bar charts used by the Traffic Equalizer make a great deal of information available at a glance, including:

- *Actual bandwidth in use at any given time, in relation to allocated bandwidth levels*
- *Allocated and maximum bandwidth levels configured for each class*
- *Average actual bandwidth utilization*
- *The extent of bandwidth borrowing by each class that is able to do so*

Following is a functional overview and instructions on starting up and using the Bandwidth Allocation and Traffic Equalizer applets within Access View.

ACCESSING THE APPLETS FROM ACCESS VIEW

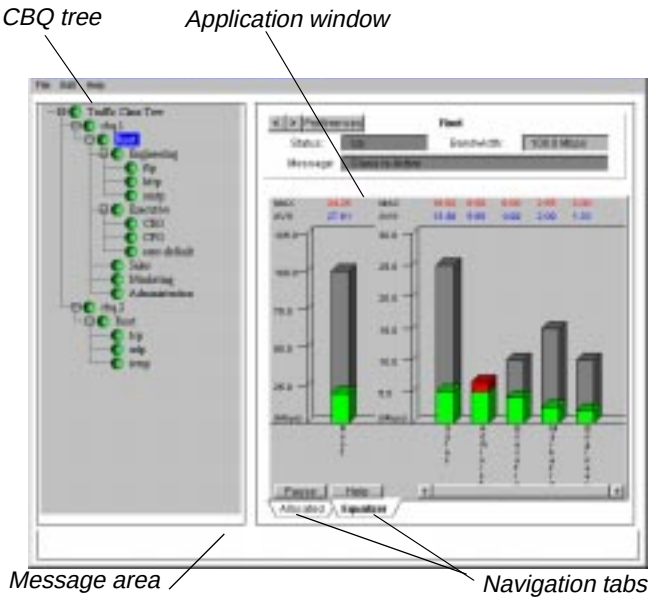


To access the Traffic Status application, select **slot.*x.x*** → **cbq.*x*** → **Graphical Status Tree** from the Management Navigator window of Access View. When you start the Traffic Status application, there will be a slight delay as it gathers information from traffic classes. Notification that classes are being gathered appears in the message area. Once data has been retrieved, the application spawns a new browser window like that shown on the following page.

HOW THE INFORMATION IS ORGANIZED

The Traffic Status application organizes information in two frames:

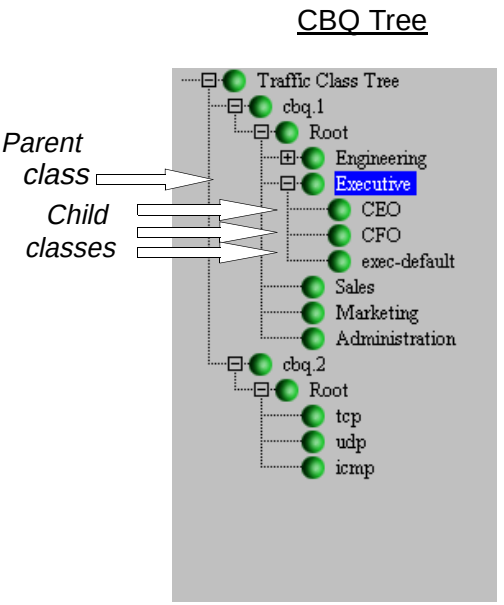
- CBQ tree
- Application window



The left frame shows the CBQ structure, with child classes linked to their parents in a hierarchical manner. The right frame contains graphs of a parent class and its children (in the case of the Traffic Equalizer) or a labeled pie chart showing bandwidth allocation relative to the total owned by a parent class (in the case of the Bandwidth Allocation applet).

MAPPING INFORMATION FROM CBQ

All information mapped by these tools derives from the CBQ tree associated with the interface. The application displays the tree side-by-side with whatever visualization tool is in use. You can select any class in the tree with a mouse click. When you do this, the class's name appears at the top of the application window.



CBQ TREE STRUCTURE

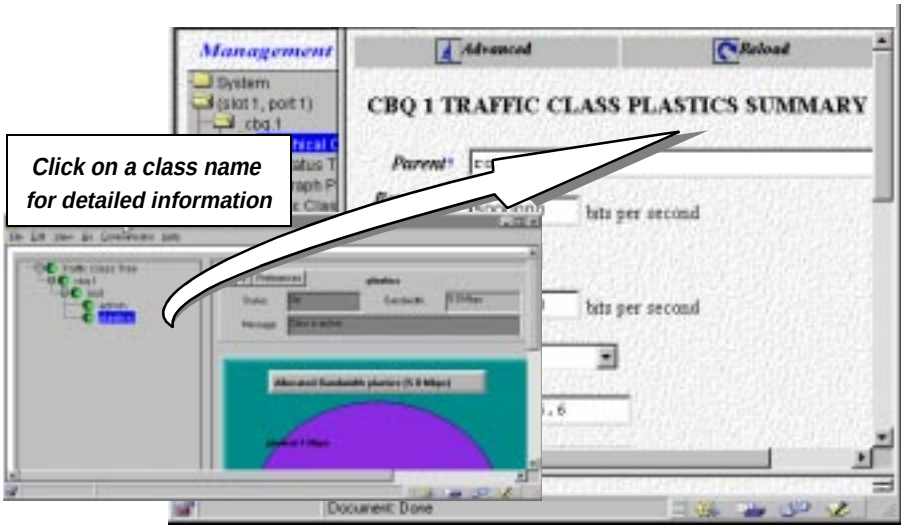
A sample CBQ tree appears in the picture above. Note that the class named Executive has been selected. This class has three child classes:

- CEO
- CFO
- exec-default

Functional relationships between classes are expressed by indenting sibling classes at the same level, and connecting them to their common parent by means of dotted lines. The + character that precedes some traffic classes (Engineering, for example) indicates that child classes are nested beneath them. You can click on any class with a preceding + to display its children. In contrast to this, the minus character (-) is used to indicate that a class has no hidden children. Selecting Executive brings the data associated with this class and its associated children into the application window.

CBQ TREE STATUS INFORMATION

Active classes represented in the CBQ display are green; inactive or conflicting classes are red. Orphan classes are also red, and connected to a parent class labeled Unattached. You can also get detailed status on any class by double-clicking on the class name. Doing this displays detailed information on the class within Access View as shown below.



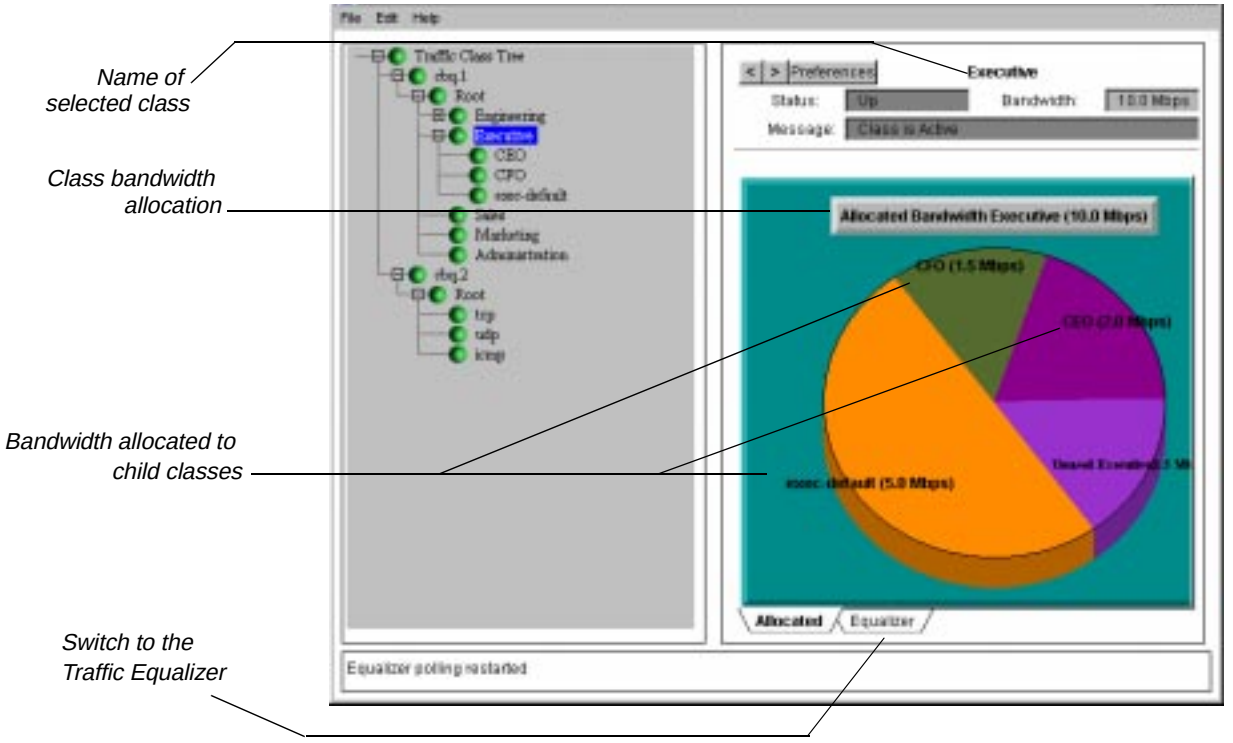
In addition, the CBQ tree is updated and redrawn periodically according to a user-defined polling interval. This feature is most useful and interesting with respect to AutoClass entries, as these come into being and are aged out of existence much more frequently than ordinary classes.

MAPPING THE CBQ TREE TO THE BANDWIDTH ALLOCATION APPLET

Figure 68 shows how these classes might be mapped by the Bandwidth Allocation applet. This mapping occurs automatically when you start up the Traffic Status application as explained on page Page 469.

Bandwidth allocated to each of Executive's child classes is labeled on the pie chart. Each slice of the pie is color-coded, and the size of each slice represents a portion of the total bandwidth owned by Executive. Any unallocated bandwidth is labeled as unused.

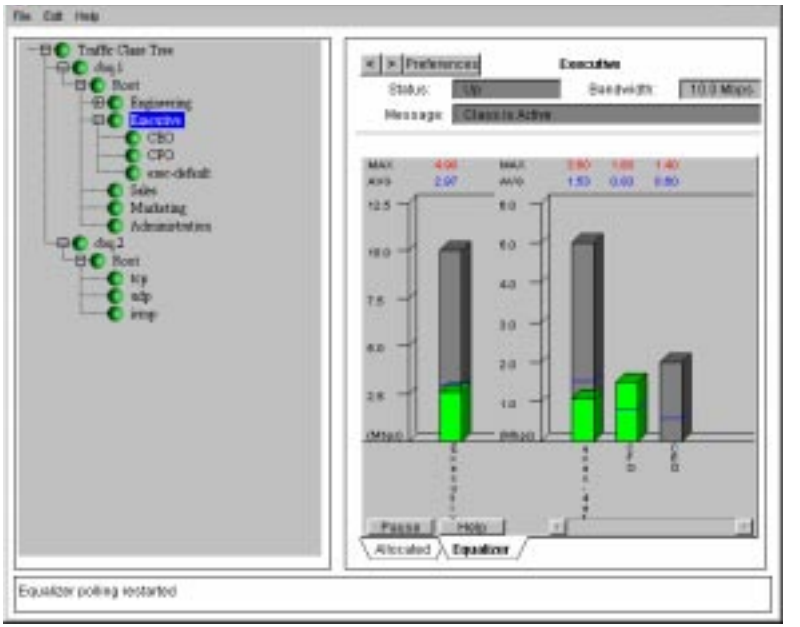
Figure 68 Sample Data Mapped in the Bandwidth Allocation Applet



USING THE TRAFFIC EQUALIZER

You can use the Traffic Equalizer to derive real-time information on the actual bandwidth utilization of each configured class. The Traffic Equalizer uses bar graphs to chart the bandwidth used by parent and child classes, and to correlate actual utilization with allocated bandwidth for each class. Color coding and other features make it easy to grasp or derive detailed information on each class. To fire up the Equalizer, click the Equalizer tab at the bottom of the screen. Figure 69 shows how the information shown in Figure 68 might be represented within the Traffic Equalizer.

Figure 69 Sample Mapping With the Traffic Equalizer



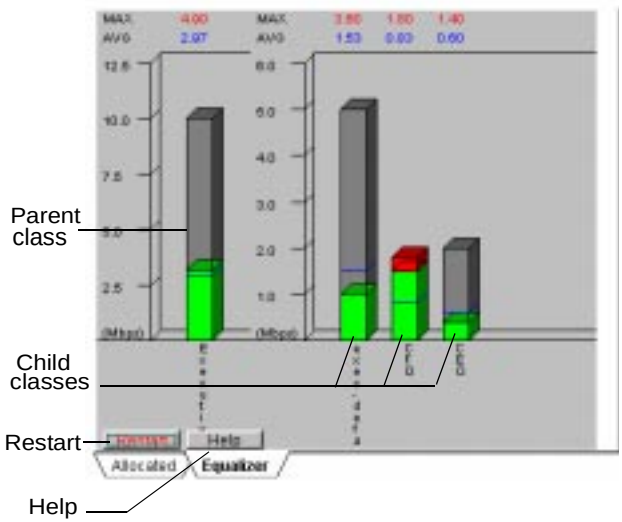
INTERPRETING THE INFORMATION

Viewed from left to right, the graphs show the behavior of the selected parent class (shown at left) and its child classes (shown at right).

Each graph contains one parent class. The number of children represented depends on the number of child classes that live below the parent. In the case of AutoClass children, this number can fluctuate over time as classes are created by packet matches, and are aged out of existence. Bandwidth utilization (in Mbps) is shown on the vertical, and traffic class instances on the horizontal axis. Preferences can be adjusted to allow child classes to be displayed alphabetically, or from highest to lowest average bandwidth utilization.

WHAT THE EQUALIZER SHOWS

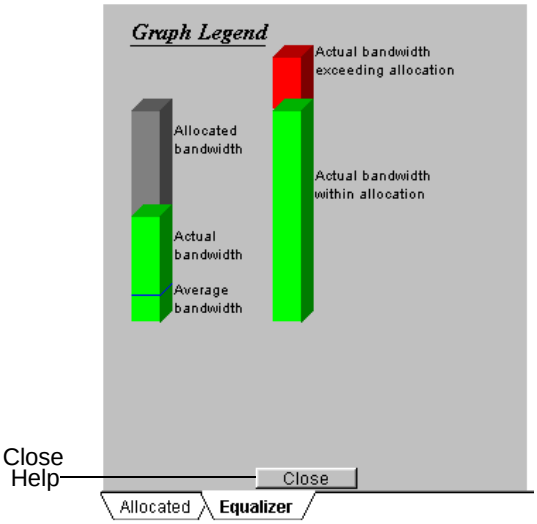
The Traffic Equalizer provides actual bandwidth utilization statistics in real-time. Changes in bandwidth utilization are reflected by color and size changes in the bar graphs. These changes are shown for parent and child classes alike.



You can click Pause at any time to pause polling operations; clicking Restart commences polling again, allowing the graphs for each class to fluctuate in accordance with their bandwidth consumption.

COLOR CODING

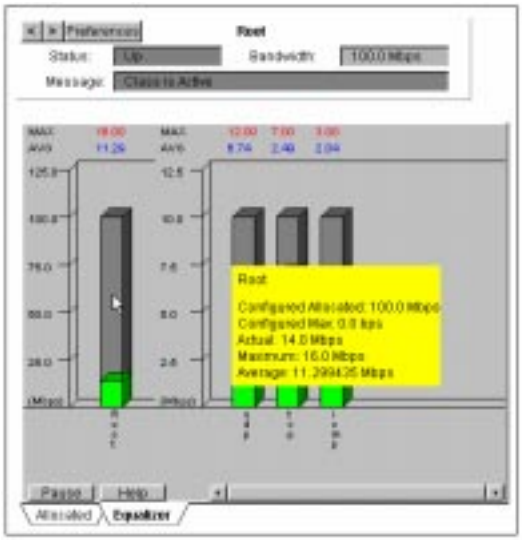
The color coding and labeling used by the Equalizer provides at-a-glance information on the behavior of traffic classes. All of this information is available by clicking the Help button at the bottom of the screen. The Graph Legend is shown at right.



- The amount of bandwidth associated with each class is represented by the gray background bar.
- The green area within each bar represents actual utilization at any given time.
- If a class is borrowing bandwidth, the amount it has borrowed is shown in red.
- The blue band that floats within the green and red areas gives the average utilization for each class.
- Max and average utilization levels are also provided in red and blue text at the top of each bar.

GETTING DETAILED HELP INFORMATION

The Traffic Status application provides floating box Help for some key areas. Simply move the cursor over designated areas to get information.



CLASS INFORMATION WITH THE TRAFFIC EQUALIZER

To get detailed information on any class, move the mouse cursor over one of the bars as shown above. The Equalizer Help is a dynamic display that provides the following information for any class:

- Allocated and maximum allowed bandwidth utilization
- Actual bandwidth in use
- Average and maximum usage values

HELP ON BANDWIDTH ALLOCATION

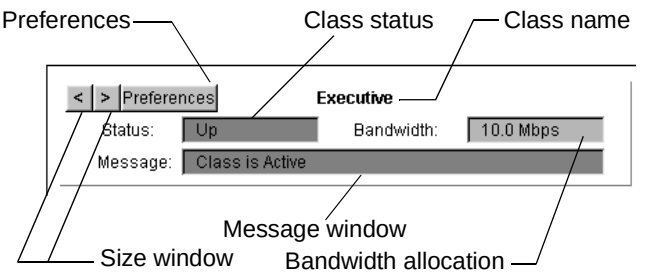
Similarly, you can get detailed information on the bandwidth used by any child class. The floating Help banner gives the bandwidth allocated for a selected class, and the percentage of the class's parent.

APPLICATION CONTROLS AND MESSAGES

A number of controls exist to allow you to adjust the way the application works, and several message fields that provide operational status.

SIZING THE SCREEN

You can use the sizing arrows to adjust the relative size of the application space and the space allocated to the CBQ class tree information.



Clicking on the right arrow moves the central pane to the right; clicking on the left moves it to the left.

BANDWIDTH

The Bandwidth field shows the bandwidth allocated to the selected class. This field is writable. To change the bandwidth allocated to a class, click in the Bandwidth field and enter a new value. You can enter a numerical value in bps (with no suffix), Kbps (using K as a suffix), or Mbps (using M as a suffix). If accepted, the new bandwidth allocation will be displayed in the field, and a message indicating the change will appear on the bottom left of the window.

Two important restrictions to bear in mind are the following. First, the bandwidth assigned cannot cause the aggregate of bandwidth apportioned to all child classes to exceed what their parent has to offer. Second, you can only edit the Bandwidth field for ordinary classes, not AutoClasses.

STATUS

The Status field gives the operational status of the class selected. The states that can appear are as follows:

- Up
- Down
- Down Conflict

A status message of Up can apply to ordinary classes or to AutoClasses. This message indicates that the class is up and functioning properly. The messages Down and Down Conflict apply only to ordinary classes. A message of Down indicates that a class is not functioning; in this case, examine the message asso-

ciated with the class to determine why not. A status message of Down Conflict indicates that either the classes' classification or scheduling parameters conflict with its sibling classes or with the amount of bandwidth owned by the parent class. This conflict prevents the class from coming up. For more information, see Chapter 7, "Configuring CBQ for Traffic Management."

MESSAGES

Messages that appear in the message area are closely related to those that appear in the Status field. These messages can be any of the following text strings:

- Class is Active
- Class is not yet Active
- Class configuration conflicts with siblings or parent
- Class operational status has been set to notInService

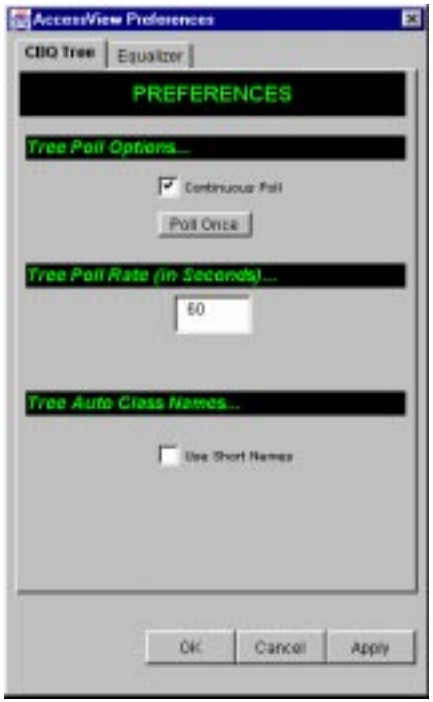
SETTING PREFERENCES

Click the Preferences button to adjust preferences. The window shown on the following page appears. The Preferences window's tabs enable you to switch between preferences relating to the CBQ tree and the Traffic Equalizer. After making changes in either window, click OK or Apply and OK to return to the application.

CBQ PREFERENCES

CBQ preferences enable you to specify the frequency with which the application polls CBQ to update information on the tree. You can also choose short names for AutoClasses.

CHANGING THE POLLING INTERVAL



By default, polling of CBQ is continuous and occurs every 60 seconds. To turn off polling, uncheck the Continuous Poll check box. To change the poll rate, enter a value in the Tree Poll Rate field.

You can click the Poll Once button to poll CBQ and repaint the tree information at any time. If you only use the Traffic Status application occasionally, or your CBQ setup changes very little over time, you may want to turn off polling and use the Poll Once button to gather fresh information on an ad hoc basis.

USING SHORT NAMES FOR AUTO-CLASSES

AutoClass names are generated automatically by the AP, based on assigned classification parameters. These names can be quite large. To shorten them for easier viewing, check the Use Short Names check box. Using short names removes the parent portion of the AutoClass name. See Chapter 7, “Configuring CBQ for Traffic Management,” for more information on AutoClass naming.

EQUALIZER PREFERENCES

To set preferences for the Traffic Equalizer, click the Equalizer tab at the top of the window. Settings made in this window influence the way the Traffic Equalizer displays information.

SORT BY...

Sorting refers to the left-to-right placement of the child bar graphs, each of which represents a distinct traffic class. You can sort these classes alphabetically, or in descending order from greatest to lowest average rate.



SHOW BARS IN...

The bars can be rendered in 3D or 2D format. The default is 3D.

INCLUDE IN GRAPH...

You can choose to include maximum and average bandwidth utilization levels, and to show the average utilization for each class by means of a floating blue band. By default, these features are on.

POLL RATE...

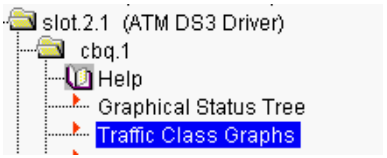
The poll rate determines how often the Equalizer polls the CBQ tree for information.

Depending on the tree's stability and levels of activity, you can set this rate to poll infrequently (by entering a value of, say, 600), or frequently (5). All values equate to time in seconds. The default value for this setting is 5 seconds; the application can be set to poll as frequently as every 2 seconds.

PLACEMENT OF BAR LABELS...

You can place labels either beside or below the bars to which they pertain. Default placement is below the bars.

TRAFFIC CLASS GRAPHS



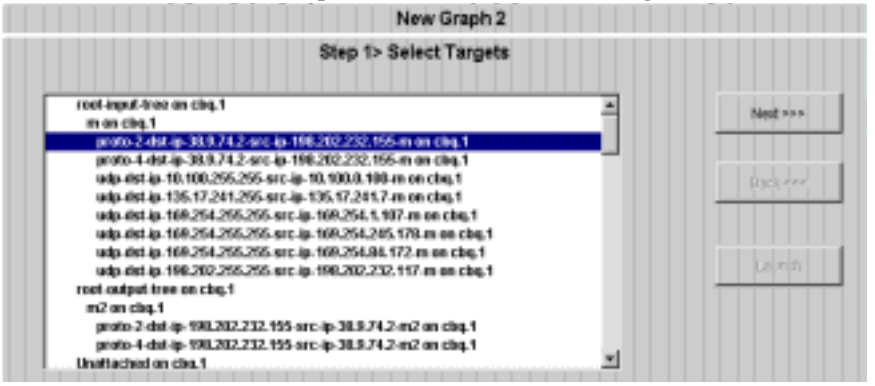
Access View also provides a Traffic Class Graphing wizard. The wizard allows you to select various items and obtain graphical representations of their usage. To access the Traffic Class Graphs wizard from the Management Navigator window of Access View, select **slot.*x*.*x*** → **cbq.*x*** → **Traffic Class Graphs.**

A browser window appears that contains three graph wizards that allows you to track the performance of your defined traffic classes. The four step wizard lets you:

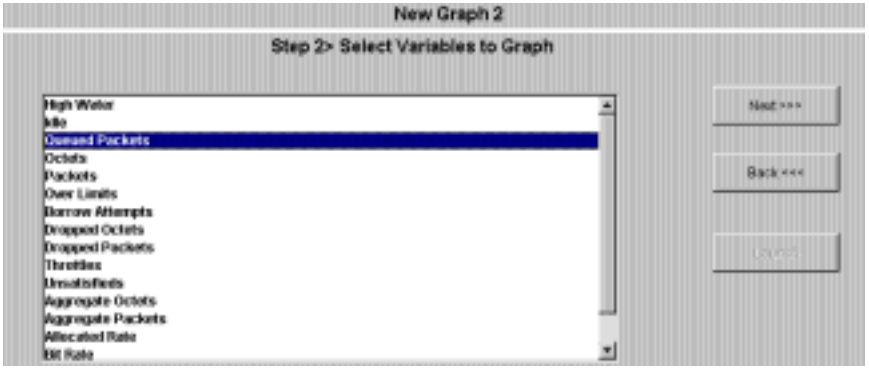
1. Select targets
2. Select variables
3. Modify graph parameters
4. Adjust graph variables

When you exit the window, graphing stops. You can launch multiple windows to obtain more than three graphs.

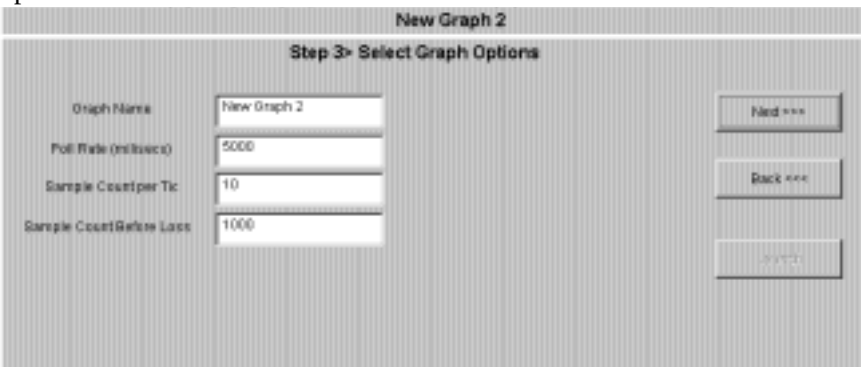
Select one or more CBQ components from the Select Targets window.



From the Select Variables to Graph window, select one or more variables.



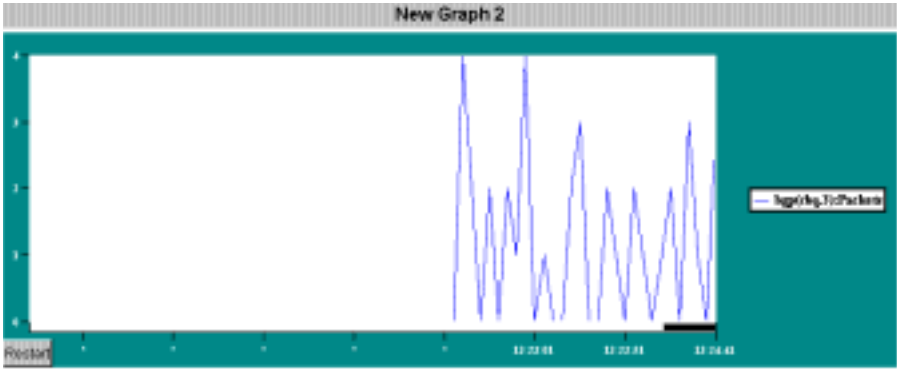
In the Select Graph Options window, change the graph name, polling rate, and sample count as desired.



In the Modify Graph Items window, you can change graph variables that affect the appearance of graph, and click the Modify Var button. Finally, click the Launch button to start the graph.



The graph dynamically updates in your browser window. You can restart the graph at any time by clicking on the Restart button at the bottom left of the window.



A

ADVANCED CONFIGURATION

This chapter provides you with information about advanced configuration options. It includes the following topics:

Configuring the Network Address Translator (NAT) 486
Configuring DHCP Relay 499
Configuring the Network Time Protocol (NTP) 503
Configuring RADIUS Authentication 507
Configuring and Using SSH 517
Configuring IP Multicast 549
Configuring Virtual LANs 557
Configuring the AP Stack With SNMP 563

CONFIGURING THE NETWORK ADDRESS TRANSLATOR

The Access Point Network Address Translator allows a domain using private IP addresses to connect to the Internet using globally-unique, registered IP addresses.

The Network Address Translator performs three types of translation:

- Basic Network Address Translation (Basic NAT) — Basic NAT allows a private IP address to be replaced by a registered IP address.
- Network Adddress Port Translation (NAPT) — NAPT allows private IP addresses and port numbers to be replaced by the registered IP address and a port number allocated from the pool.
- Load Sharing Network Address Translation (LSNAT) — LSNAT allows one registered IP address to serve as a virtual IP address that maps to a pool of multiple private IP addresses.

THE TRANSLATOR IN A NUTSHELL

The Network Address Translator runs on a router at the border of a stub domain. Private IP addresses are removed from the outgoing IP packets and replaced with the registered IP addresses. The reverse translation is done on incoming IP packets.

Basic NAT and NAPT can function either dynamically (by configuring pools) or statically (by adding entries to the binding table). Together, Basic NAT and NAPT are commonly referred to as Traditional NAT.

Traditional NAT, referred to as NAT in the rest of this document, is not viewed as a pure architectural component in IP networks. The IP protocol has traditionally assumed end-to-end address significance, such that many applications (including FTP) embed IP addresses in the data beyond the IP header. This requires NAT to then look into many applications to modify data, checksums, and sometimes sequence numbers on an application-by-application or protocol-by-protocol basis. Additionally, packets that have authenticated or encrypted addresses or data will simply not work through a NAT function.

NAT is used to deal with these issues:

- The use of private addresses inside a domain for ease of changing ISPs or conserving IP address space

- The use of illegal IP addresses inside a domain with a desire to connect to the global Internet
- Load balancing using a virtual server address approach
- The hiding of internal IP addressing structure from the outside world

CONFIGURING NAT

The configuration of address translation centers around these tasks:

- Adding static bindings
- Configuring the pools associated with a specific type of address translation
- Configuring the private networks to be translated
- Associating the private networks with the pools
- Adding the NAT layer on the interface to public Internet

CONFIGURING BASIC NAT POOLS

Basic Network Address Translation (Basic NAT) is a translation method where a private IP addresses is replaced by a registered IP address. The port number is not modified. To configure pools for Basic NAT, you must:

1 Create a new pool using this command.

```
> add nat address-translation-pool.<pool name>
```

This command creates a row in the NAT Pool Table indexed by the pool name. Pool names cannot exceed 15 characters.

2 Add ranges of registered IP addresses to the pool using this command.

```
> add nat address-translation-pool.<pool name> range-start.<ip address>
  range-end.<ip address> net-mask <mask>
```

This command creates a row in the NAT Pool Range Table indexed by the pool name and beginning IP address. A pool must exist before ranges can be added to it. A pool can have up to five ranges. Ranges are limited to 254 addresses, cannot contain network or broadcast addresses, and cannot overlap.

CONFIGURING NAPT POOLS

Network Address Port Translation (NAPT) is a translation method where the translation pool consists of a single registered address and a pool of port numbers. Private IP addresses and port numbers are replaced by the registered IP address and a port number allocated from the pool. NAPT is also referred to as port address translation (PAT) and masquerading. The pool address is usually the address of the public network interface. A pool consists of a single range of ports.

To configure NAPT pools, use this command to create a row in the NAPT Pool Table indexed by pool name.

```
> add nat port-translation-pool.<poolname> pool-address <ip address>
  net-mask <mask> [range-start <port number> range-end <port number>]
```

The range parameters are optional and specify the range for TCP and UDP port numbers and ICMP sequence numbers. If not specified, the range defaults to 20000 through 24095. The port range must begin above port 5000 and cannot exceed 8192 ports.

CONFIGURING LSNAT POOLS

Load Sharing Network Address Translation (LSNAT) allows one registered IP address to serve as a virtual IP address that maps to a pool of multiple private IP addresses. LSNAT must be configured statically, and can operate in mode similar to either Basic NAT or NAPT.

To configure LSNAT pools:

1. Create a new load sharing pool using this command.

   ```
   > add nat load-sharing-pool.<pool name> virtual-address <ip address>
     net-mask <mask> [virtual-port <port number> protocol [6|17]]
   ```

   This command creates a row in the LSNAT Pool Table indexed by the pool name. The port number and protocol are optional and default to 0. (A port number and protocol of 0 will match any port number and protocol.) If a port number and protocol are specified for a pool, a port number and protocol must also be specified for any range assigned to the pool.

2. Add ranges of private IP addresses to the pool using this command.

   ```
   > add nat load-sharing-pool.<pool name> range-start.<ip address>
     range-end <ip address> net-mask <mask> [port <port number>]
     [protocol [6|17]]
   ```

This command creates a row in the LSNAT Pool Range Table indexed by the pool name and beginning IP address. A pool must exist before ranges can be added to it. A pool can have up to five ranges. Ranges are limited to 254 addresses, cannot contain network or broadcast addresses, and cannot overlap. The port number is optional and defaults to 0. The protocol specified by 6 is TCP and the protocol specified by 17 is UDP.

CONFIGURING PRIVATE NETWORKS

To indicate which private networks are to be translated, an entry is made in the Private Network Table. To configure private networks:

1. Add a private network and mask to the Private Network Table using this command.

   > add nat private-network.<network address> net-mask <mask>

   This command creates a row in the table indexed by the network address.

2. Associate dynamic translation pools with private networks using this command.

   > add nat private-network.<network address> pool.<pool name>

   This command creates a row in the Private Network Associated Pool Table indexed by network address and pool name. Private networks can have a maximum of three pools.

ADDING NAT LAYERS

A network address translator operates as a layer between the IP forwarding layer and the data link layer on the interface leading to the public Internet. To add NAT layers, use the *stack* command as shown in this example.

> stack slot.2.1 cbq.2 hssi.2 ppp.2 nat.1 ip.2

DISPLAYING NAT CONFIGURATION INFORMATION

This section describes how to display the NAT configuration information.

DISPLAYING NAT POOLS

To view the configured NAT pools, use this *show* command.

```
> show nat address-translation-pools summary
```

To view a particular NAT pool, use this command.

```
> show nat address-translation-pool.<pool name> summary
```

To view the configured NAT pools and ranges, use this command.

```
> show nat address-translation-pools ranges summary
```

DISPLAYING NAPT POOLS

To view the configured NAPT pools, use this *show* command.

```
> show nat port-translation-pools summary
```

To view a particular configured NAPT pool, use this command.

```
> show nat port-translation-pool.<pool name> summary
```

DISPLAYING LSNAT POOLS

To view all configured LSNAT pools, use this *show* command.

```
> show nat load-sharing-pools summary
```

To view a particular configured LSNAT pool, use this command.

```
> show nat load-sharing-pool.<pool name> summary
```

To view the configured ranges associated with LSNAT pools, use this *show* command.

```
> show nat load-sharing-pools ranges summary
```

DISPLAYING CONFIGURED PRIVATE NETWORKS

To view the configured private networks, use this *show* command.

```
> show nat private-networks summary
```

To view a particular configured private network, use this command.

```
> show nat private-network.<network address> summary
```

To view the pools associated with private networks, use this command.

```
> show nat private-networks pools summary
```

A pool can be associated with one or more private networks, and a private network can be associated with one or more dynamic pools.

Using Dynamic or Static Bindings

Bindings are one-to-one mappings between a private IP address and a registered IP address or between a private IP address/port pair and a registered IP address/port pair.

Bindings can be static or dynamic. Both static and dynamic bindings are listed in the Binding Table. The Binding Table is checked before allocations are made from a pool. The Binding Table entries take precedence. Dynamic bindings are added to the table when an address or port number is allocated from a dynamic pool. Static bindings are added through user configuration. Configuring an LSNAT pool also adds static bindings to the table.

Static address translation is accomplished by configuring a static entry in the Binding Table. The table is indexed by private IP address and port, registered IP address and port, and protocol. Actual IP addresses must be specified. Port numbers and protocol may be specified as zero. A zero functions as a wildcard and will match any value. If you specify a registered port, you must also specify a private port. Similarly, if you specify a private port, you must also specify a registered port. If you specify ports, you must also specify a protocol.

Duplicate or ambiguous bindings are not allowed. Duplicate bindings occur when two bindings have identical registered or private IP address/port pairs. Ambiguous bindings occur when the registered or private IP address/port pairs have matching IP addresses and the port for one binding is 0. Because of the one-to-many nature of LSNAT, load sharing bindings are exceptions to the duplicate binding rule and are maintained differently.

```
> add nat binding.<ip address> pri-port.<port number> reg-addr.<ip address>
  reg-port.<port number> protocol.[6|17|0]
```

CONFIGURING MAXIMUM NUMBER OF SESSIONS

To prevent your router from being flooded by session requests, you might want to limit the maximum number of sessions that can be created from a binding. The maximum number of sessions is initially set to a system default, but it can be modified if needed.

If you want to set the maximum number of inbound sessions for your bindings, use this command.

```
> config nat max-inbound-sessions <number>
```

If you want to set the maximum number of outbound sessions for your bindings, use this command.

```
> config nat max-outbound-sessions <number>
```

0 is the minimum value and 5000 is the maximum value that you can specify for the number of sesssions. Setting the value below its current value does not cause sessions to be removed. Sessions above the new value are allowed to expire normally.

DISPLAYING BINDINGS

To view all the bindings, use this *show* command.

```
> show nat bindings summary
```

To view a particular binding, use this command.

```
> show nat binding.<instance> summary
```

DISPLAYING TRANSLATION SESSIONS

A translation session represents a connection between a host on the private network and a host on an outside network. A session is uniquely defined by the private IP address and port, and the outside IP address and port. Translation sessions can be initiated from inside or outside the private network. Multiple sessions may be instantiated from a single binding.

All active translation sessions are listed in the Session Table. To view all the sessions, use this *show* command.

```
> show nat sessions summary
```

To view a particular session, use this command.

```
> show nat session.<instance> summary
```

USING SESSION TIMERS

A timer is associated with each type of NAT session. If no traffic is seen for an entire timeout period, the session is removed from the Session Table. Timers are initially set to a system default, but they can be modified if needed. Time-out values are expressed in seconds. NAT provides the following timers:

- UDP timeout (udp-timeout)
- DNS timeout (dns-timeout)
- TCP timeout (tcp-timeout)
- FINREST timeout (finrest-timeout)
- SYN timeout (syn-timeout)
- ICMP timeout (icmp-timeout)

To view the timers, use the *show* command.

```
> show nat timers
```

To modify a timer value, use the *config* command.

```
> config nat timers udp-timeout <value>
```

DISPLAYING NAT COUNTERS

To view the NAT counters, use this command.

```
> show nat counters
```

ENABLING AND DISABLING NAT

When the NAT layer is enabled, the network address translator will translate those addresses that match the configured private networks. Addresses that do not match will pass through NAT unchanged. When the NAT layer is disabled, no address translation occurs.

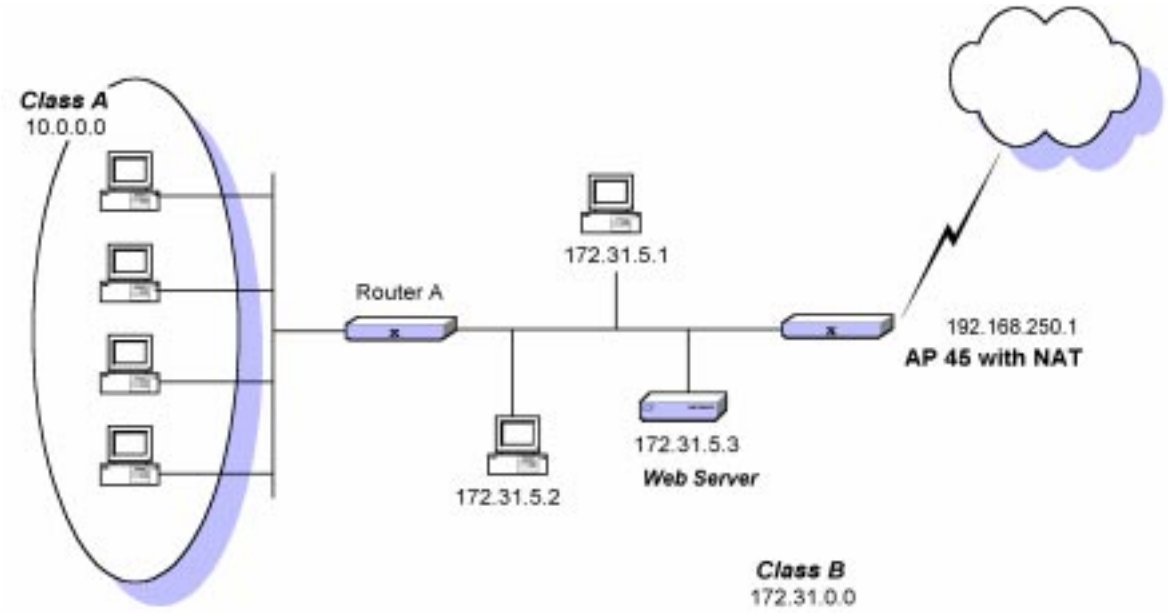
To enable NAT, use the *config nat state enabled* command.

To disable NAT, use the *config nat state disabled* command.

SAMPLE CONFIGURATIONS

This section contains some samples for configuring your pools. Figure 70 illustrates the sample NAT and NAPT configurations.

Figure 70 Sample NAT and NAPT Configuration



NAT POOL CONFIGURATION

In this example, a private network has been allocated the Class C network address of 192.168.5.0. The private network has a publicly available Web server (172.31.5.3) that requires a static binding. The network administrator chooses 192.168.5.80 as the address used for the static binding.

Refer to Figure 70 when following these steps to configure the NAT pool for this sample configuration.

1 Add the static binding.

```
> add nat binding.172.31.5.3 pri-port.0 reg-addr.192.168.5.80\
  reg-port.0 protocol.0
```

2 Create an address translation pool called pool1and add two ranges.

```
> add nat address-translation-pool.pool1
```

```
> add nat address-translation-pool.pool1 range-start.192.168.5.1\
  range-end.192.168.5.79 net-mask 255.255.255.0

> add nat address-translation-pool.pool1 range-start.192.168.5.81\
  range-end.192.168.5.254 net-mask 255.255.255.0
```

Note that the address used for the static binding (192.168.5.80) must be omitted from the specified ranges.

3. Add networks and attach to the pool from step 2.

```
> add nat private-network.10.0.0.0 net-mask 255.0.0.0

> add nat private-network.10.0.0.0 pool.pool1

> add nat private-network.172.31.5.0 net-mask 255.255.0.0

> add nat private-network.172.31.5.0 pool.pool1
```

4. Add a NAT instance to stack.

```
> stack slot.2.1 cbq.2 hssi.2 ppp.2 nat.1 ip.2
```

5. Enable NAT.

```
> config nat state enabled
```

NAPT Pool Configuration

In this example, the private network uses the registered address of the router's interface to the public Internet. The private network has a publicly available Web server (172.31.5.3) that requires a static binding. Assume the network administrator chooses 192.168.5.80 as the address used for the static binding.

Refer to Figure 70 when following these steps to configure the NAPT pool for this sample configuration.

1. Add the static binding.

```
> add nat binding.172.31.5.3 pri-port.0 reg-addr.192.168.5.80\
  reg-port.0 protocol.0
```

Note that the static binding does not use the outside interface address.

2. Create a port translation pool called pool2.

```
> add nat port-translation-pool.pool2 pool-address 192.168.250.1\
  range-start 50000 range-end 54095
```

3 Add networks and attach to the pool from step 2.

```
> add nat private-network.10.0.0.0 net-mask 255.0.0.0
> add nat private-network.10.0.0.0 pool.pool2
> add nat private-network.172.31.5.0 net-mask 255.255.0.0
> add nat private-network.172.31.5.0 pool.pool2
```

4 Add a NAT layer to the stack for the appropriate slot between the data link layer and IP layer.

```
> stack slot.2.1 cbq.2 hssi.2 ppp.2 nat.1 ip.2
```

5 Enable NAT.

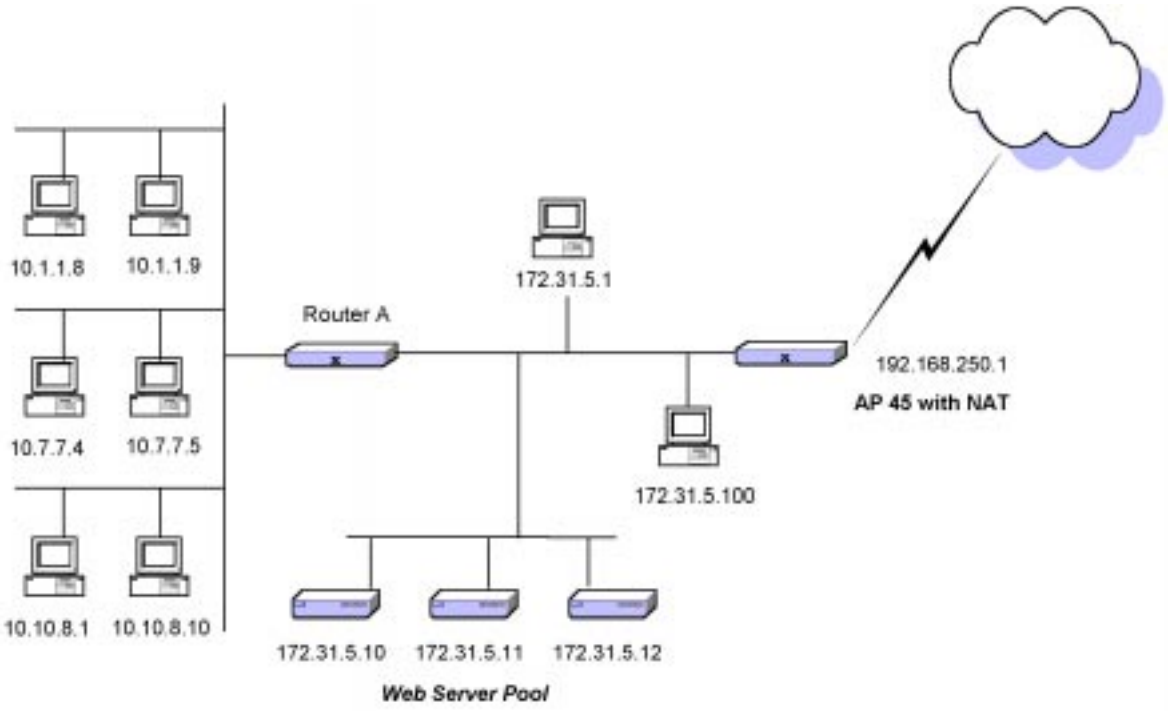
```
> config nat state enabled
```

LSNAT Pool Configuration

In this example, a private network has been allocated the Class C network address of 192.168.5.0. The private network has a publicly available pool of Web servers (172.31.5.10 - 172.31.5.12) that requires a load sharing pool. The network administrator chooses 192.168.5.80 as the virtual address.

Figure 71 illustrates the sample LSNAT configuration.

Figure 71 Sample LSNAT Configuration



Refer to Figure 71 when following these steps to configure the LSNAT pool for this sample configuration.

1 Create a load sharing pool called loadpool.

```
> add nat load-sharing-pool.loadpool virtual-address 192.168.5.80\
  virtual-port 0 protocol 0
```

2 Add a range to the load sharing pool.

```
> add nat load-sharing-pool.loadpool range-start.172.31.5.10\
  range-end.172.31.5.12 net-mask 255.255.255.0 port 0 protocol 0
```

3 Create a dynamic NAT pool called pool2 and add ranges.

```
> add nat address-translation-pool.pool2
```

```
> add nat address-translation-pool.pool2 range-start.192.168.5.1\
  range-end.192.168.5.79 net-mask 255.255.255.0
```

```
> add nat address-translation-pool.pool2 range-start.192.168.5.81\
  range-end.192.168.5.254 net-mask 255.255.255.0
```

Note that the virtual address (192.168.5.80) must be omitted from the specified ranges.

4 Add networks and attach to the NAT pool created in step 3.

```
> add nat private-network.10.0.0.0 net-mask 255.0.0.0
> add nat private-network.10.0.0.0 pool.pool2
> add nat private-network.172.31.0.0 net-mask 255.255.0.0
> add nat private-network.172.31.0.0 pool.pool2
```

5 Add a NAT instance to stack.

```
> stack slot.2.1 cbq.2 hssi.2 ppp.2 nat.1 ip.2
```

6 Enable NAT.

```
> config nat state enabled
```

If a server pool is intended to provide only a single service, an LSNAT pool can be configured for a specific port. For instance, if access was to be restricted to Web traffic only in the previous example, the LSNAT pool could be configured with the following command:

```
> add nat load-sharing-pool.loadpool virtual-address 192.168.5.80\
  virtual-port 80 protocol 6
```

The ranges would also require a port number.

```
> add nat load-sharing-pool.loadpool range-start.172.31.5.10\
  range-end.172.31.5.12 port 80 protocol 6
```

Note that if in this scenario, outbound sessions are expected to be instantiated from any of the servers. An NAPT pool must be assigned to the private networks on which the servers reside. Otherwise, ambiguous bindings might be created.

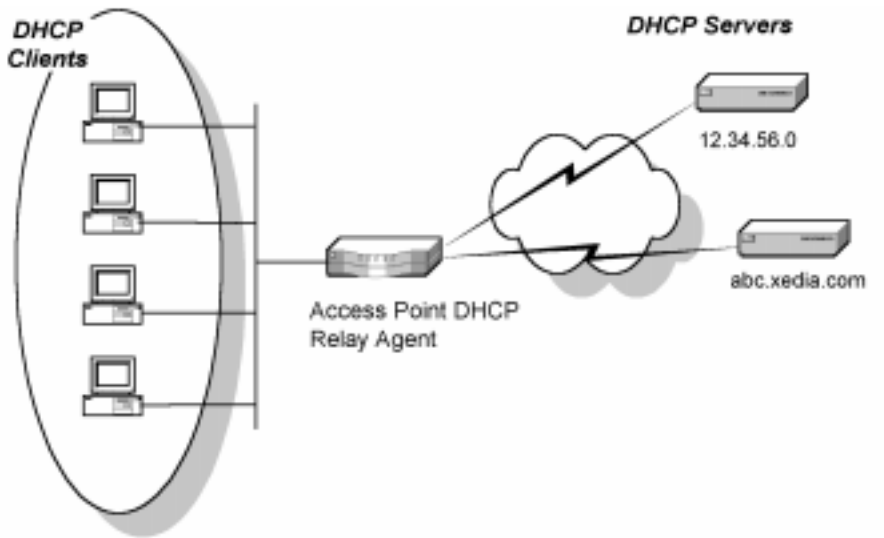
CONFIGURING DHCP RELAY

Dynamic Host Configuration Protocol (DHCP) is an industry standard method of configuring host computer systems on a TCP/IP network from a central database server. You can configure the AP to act as a DHCP relay agent, sending requests to remote DHCP servers and passing replies back to requesting clients.

ABOUT DHCP RELAY

Ordinarily, DHCP works within a single network segment. Client systems contact DHCP servers by sending out a discover message to the universal announce IP address (255.255.255.255) with a return address of 0.0.0.0. Servers reply to these messages with an offer to lease clients an IP address and other information. Clients then select an IP address.

Figure 72 Access Point as a DHCP Relay Agent



Once a server acknowledges an address selection, clients use information retrieved from the server to configure their TCP/IP stack. The client sends a message back to the server selecting the IP address lease, and the server replies with an acknowledgement. The client then configures the local TCP/IP stack using the information received from the server.

When the lease period is about half over, the DHCP client will normally con-

tact the DHCP server and negotiate a lease extension. This allows the client to retain the same IP address over an extended period of time even if the lease periods are short. If the client tries to renegotiate a lease and cannot reach the DHCP server, it will continue to use the current lease, and will re-attempt to contact the server on a regular basis, until the lease expires.

GLOBAL DHCP CONFIGURATION PARAMETERS

You can set these global DHCP relay parameters: DHCP relay mode, max-hops count, and add circuit ID.

To enable DHCP relay, use this command:

```
> config services dhcp-relay mode enabled
```

You can also set the max-hops count using this command:

```
> config services dhcp-relay max-hops <number>
```

Configuring max-hops limits the number of DHCP relay agents a request can traverse. By default, max-hops is set to 4. This should be sufficient for most situations, as a single relay agent is normally all that is needed.

You would enable add circuit ID so that you can minimize the traffic on ATM or Frame Relay networks. Enabling this parameter causes a response to a DHCP request to be sent back over the same circuit on which the DHCP request was received by the relay agent. Otherwise, all hosts on all circuits could receive all responses.

To allow the relay agent to add the circuit ID:

```
> config services dhcp-relay add-circuit-id true
```

To display the global DHCP relay parameter settings:

```
> show services dhcp-relay summary
```

```
                    SERVICES DHCP-RELAY SUMMARY

          Mode* = disabled
      Max Hops* = 4
Add Circuit Id* = false
```

CONFIGURING SERVER DESTINATIONS

You configure DHCP servers by adding them to a server destination table. The AP uses the table to relay requests it receives to appropriate destinations. Like the AP file transfer table, each entry is indexed by a single number. The AP relays requests in ascending order, based on the index. This is to say that requests are relayed to destinations associated with an index of 1 first, followed by requests associated with an index of 2, and so on.

You can identify DHCP servers by server address, server name, or interface address. Following are examples of each approach:

```
> add services dhcp-relay destination.<index> \
  admin-address 192.202.232.247
```

```
> add services dhcp-relay destination.<index> admin-address sky.lucent.com
```

```
> add services dhcp-relay destination.<index> admin-address 192.202.232.0
```

If you specify a server address, the AP unicasts requests to this address. If you use a server name (such as sky.lucent.com), the AP's DNS client resolves the name and unicasts to this address. Note that you must configure the DNS client to resolve any names you specify.

If you specify an interface address that matches a local subnet, the AP broadcasts requests from the interface corresponding to the subnet. The AP will not broadcast requests on the same interface on which a request was received.

RELAYING REQUESTS BY INTERFACE

You can specify an IP instance so requests coming in through this interface are relayed to a particular DHCP server. By default, requests are relayed to all specified servers. To use a server attached to a particular IP instance:

```
> add services dhcp-relay destination.<index> admin-address <IP address>
  incoming-interface ip.<instance>
```

The *admin-address* is the server address and the *incoming-interface* is the specific IP instance.

RELAYING REQUESTS ON A MULTINETTED INTERFACE

If more than one IP address exists on an interface, the AP relays requests on the last address added to the interface.

SHOWING DESTINATIONS

To show DHCP destinations, use this command:

```
> show services dhcp-relay destinations
```

```
                        SERVICES DHCP-RELAY DESTINATIONS

INDEX|                  Admin Address| Resolved Address|   Requests|
     |                        Replies|        Protocol| Row Status|
     |              Incoming Interface|                |           |
-----|--------------------------------|----------------|-----------|
   1|                       1.2.3.4|         1.2.3.4|          0|
     |                             0|     dhcpAndBootp|     active|
     |                          ip.1|                |           |
```

Use this command to show the protocols that will be relayed to a specific destination.

```
> show services dhcp-relay destination.<index>
```

A summary screen is shown here. Note that this destination (index 1) relays both DHCP and bootp. These protocols are both relayed by default.

```
                        SERVICES DHCP-RELAY DESTINATION.1

    Admin Address* = 1.2.3.4
  Resolved Address = 1.2.3.4
          Requests = 0
           Replies = 0
         Protocol* = dhcpAndBootp
        Row Status* = active
Incoming Interface* = ip.1
```

You can relay one or both of these protocols. For instance, to relay only bootp (and not DHCP) to destination 1, issue this command:

```
> config services dhcp-relay destination.1 protocol bootp
```

Issuing this command instructs the AP to relay only bootp requests to the destination associated with index 1. Unless otherwise instructed, the AP continues to relay both protocols to other destinations.

CONFIGURING THE NETWORK TIME PROTOCOL

The AP uses the Network Time Protocol (NTP) to synchronize its system clock with Coordinated Universal Time, or UTC.

The section that follows explains how NTP is implemented on Access Point, and how to configure NTP on each AP system.

NTP IN A NUTSHELL

Basically, NTP works by exchanging time-stamped UDP packets within a redundant hierarchy of network hosts. Known as the *synchronization subnet*, this hierarchy consists of a set of primary and secondary time servers and clients, together with the network paths connecting them. Clients running NTP request time-keeping information from the NTP servers to which they are connected. As a client system, the AP can receive time-keeping information from one or many systems. The NTP client collects all of this information on an ongoing basis, and updates its system clock accordingly.

The synchronization subnet consists of a hierarchy of up to 15 strata of hosts, which exchange time information with one another on an ongoing basis. The lower the stratum number, the more accurate the time information. Primary and secondary hosts are known as stratum 1 and stratum 2 servers, respectively. NTP assumes that these hosts have more reliable time information than hosts at stratum 3. Accordingly, the protocol ensures that NTP clients only synchronize their time with hosts at a lower (that is, more accurate) or equal (peer) stratum.

CALCULATING THE CORRECT TIME

It is important to note that NTP does not assume that any host has the *correct* time. Instead, the protocol uses a selection algorithm to determine the most accurate time from all received information. In theory, the more time samples NTP has to generate its time, the more accurate that time will be.

For this reason, NTP works best when a client receives time-keeping information from several servers of a lower stratum number. For instance, an AP that is a stratum 2 host should synchronize its clock with at least one stratum 1 clock source. If it is possible, administrators should synchronize each AP with two lower-stratum servers, and one server at a peer stratum. The systems at a lower stratum are assumed to be more accurate; peer systems are assumed to have an equal level of accuracy.

CHECK THE ACTUAL UTC BEFORE CONFIGURATION...

Before enabling NTP, make sure the AP's local clock does not differ from the actual UTC by more than 1000 seconds. If the difference between these two times exceeds 1000 seconds, NTP is unable to synchronize.

For this reason, you should begin by setting the AP clock according to the UTC. Do this using this command:

```
> config system date-and-time <date and time>
```

where

date and time	As expressed in this format: format: month/day/year hour:minute:seconds:centiseconds range: 01-12/01-31/xxxx 0-23:00-60:00-60:00-99

For example, the following command sets system date and time to March 30, 1999 and 3:30pm.

```
> config system date-and-time 3/30/99 15:30:00
```

FINDING NTP COMMANDS

All NTP configuration commands are located under services. To show all NTP commands, enter the following string and press the spacebar:

```
> show services ntp
```

By default, NTP is disabled on the AP. Configuring NTP involves two steps:

- Enabling NTP
- Configuring NTP

ENABLING NTP

By default, NTP is disabled on the AP. Enabling NTP automatically creates and activates any configured associations. Disabling NTP destroys all associations, including the association with the current synchronization source.

Use this command to enable NTP.

```
> config services ntp system-state enabled
```

To view the state of NTP, use the *show* command. This command shows rele-

vant top level information about NTP including the current synchronization source, delay to the root of the synchronization subnet, and the last time the local clock was corrected by NTP.

> show services ntp summary

```
                        SERVICES NTP SUMMARY

                System State* - disabled
System Source Address State* = disabled
                System Stratum  = 16
                  System Mode   - unspecified
             System Precision   = -6
          System Clock Source   = 0.0.0.0
          System Poll Interval  - 4
        System Leap Indicator   = unsynchronized
            System Root Delay   -  00000.000000
       System Root Dispersion   -  00000.000000
System Reference Timestamp      = 1900/01/01 00:00:00:000000
            System Local Time   - 1999/07/21 11:35:23:513766
```

ADD NTP SERVERS

To create a client association, you must add a server to the list of servers maintained by NTP. Entries in the server list are created with the recommended default values for the NTP version and the minimum and maximum polling intervals. Polling intervals are expressed as a power of 2. If NTP is enabled, the AP attempts to create an association with the server immediately.

Use this command to add a server to the NTP server list:

> config services ntp server.<ip address>

To show the list of servers, use the show command:

> show services ntp servers summary

To create a symmetric active association with another peer, add a peer to the peer list maintained by NTP. Do this using this command:

> config services ntp peer.<ip address>

Entries in the peer list are created with the recommended default values for the NTP version and the minimum and maximum polling intervals. Polling intervals are expressed as a power of 2. If NTP is enabled, an association with the peer is attempted immediately.

If you are configuring an NTP peer in a situation where NTP packets can be sent to the same peer over different interfaces (such as when using a multi-

homed router), you must enable the system source address state and set a router address as shown with these sample commands:

```
> config services ntp system-source-address-state enabled
```

```
> config ip.6 address.2.1.1.2 router-address true
```

DISPLAYING NTP CONFIGURATION INFORMATION

To show the list of peers, use the *show* command:

```
> show services ntp peers summary
```

The AP maintains information about every association. To view general information about associations, use this command:

```
> show services ntp associations summary
```

General information includes the peer's stratum and the precision of the peer's system clock.

More specific information about association statistics and time-stamps is also available. To view association statistics, issue the following command:

```
> show services ntp associations statistics
```

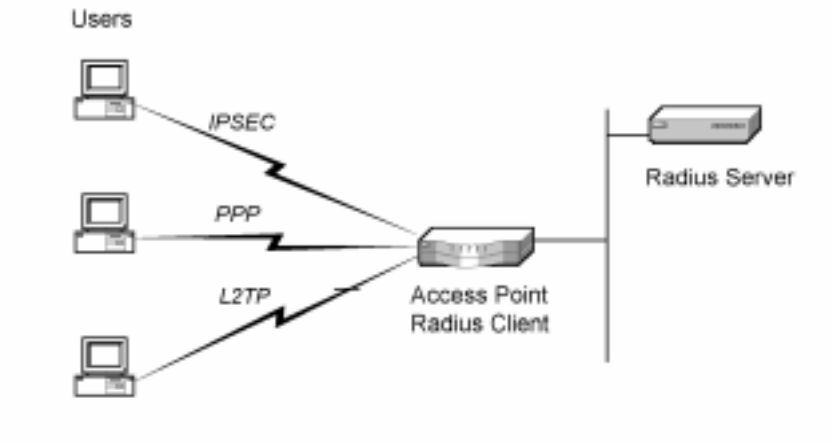
To show time-stamp information, use this command. The time-stamps shown are time-stamps included in the last message between the local host and the peer.

```
> show services ntp associations timestamps
```

CONFIGURING RADIUS AUTHENTICATION

Remote Authentication Dial In User Service (RADIUS) is a client/server protocol used to transfer authentication, authorization, configuration, and accounting information between the Access Point client applications (for example, the QVPN Client) and the RADIUS server (see Figure 73). As a result, you must configure both the client and the server to transfer information between them.

Figure 73 RADIUS Configuration



RADIUS uses a *secret* (shared secret key), that is never sent over the network, to authenticate transactions between the client and server. On most RADIUS servers, the IP address of the AP (RADIUS client) used to talk to the server identifies the client key that should be used for authentication. Both the client and the server must be configured to use the same secret key for authentication to succeed.

When you specify RADIUS as the user authentication method, you must configure the RADIUS client with at least one RADIUS authentication server configuration record.

CONFIGURING THE RADIUS SERVER

A RADIUS server must be running and reachable to exchange authentication and accounting requests. The RADIUS server must be configured with the following information:

- Shared secret keys for all client APs
- Appropriate user passwords
- Dictionary file with Xedia attributes and required standard attributes

When RADIUS authentication is used, attributes stored in the RADIUS database are returned and used to authenticate the connecting user, and perform other functions.

ABOUT THE DICTIONARY FILE

Most RADIUS servers have a dictionary file that maintains supported attributes. The dictionary file contains the attribute value (name to number mapping) and the attribute type (number, string, or IP address). To support some of the Xedia attributes, you will probably have to modify the dictionary file to include lines for the vendor-specific attributes in the appropriate number space. Many servers use lines containing tab-separated text. Refer to your RADIUS server documentation for detailed information.

This sample shows part of a dictionary file containing vendor-specific attributes (with the vendor ID for Xedia already defined earlier in the file):

```
#-----------------------------------------------
# Xedia Vendor Specific
#-----------------------------------------------
ATTRIBUTE  Xedia-DNS-Server            1   ipaddr   Xedia
ATTRIBUTE  Xedia-NetBios-Server        2   ipaddr   Xedia
ATTRIBUTE  XEDIA-ADDRESS-POOL          3   string   Xedia
ATTRIBUTE  XEDIA-PPP-ECHO-INTERVAL     4   integer  Xedia
ATTRIBUTE  XEDIA-SSH-PRIVILEGES        5   integer  Xedia
ATTRIBUTE  XEDIA-CLIENT-ACCESS-NETWORK 6   string   Xedia
```

Note that you should not change the vendor offset for the RADIUS server unless this server does not support RFC recommended vendor encoding using attribute #26. Note that you must specify the vendor (the Xedia vendor ID is 838) before you specify the attributes using that vendor information. If you define attributes in the vendor-specific number space, you just specify the attribute number (1 and 2).

If your server does not support vendor attribute extensions, then you should add lines similar to the following example:

```
ATTRIBUTE Xedia-Std-DNS       201         ipaddr
ATTRIBUTE Xedia-Std-NetBios   202         ipaddr
```

Note that these attributes reside in the normal RADIUS attribute range. The attributes use the base that you specify with the *vendor-offset* parameter (see "Configuring Server Entries" on page 513). For example, when you define attributes in the standard number space, you would add the *vendor-offset* parameter value to the attribute number to get this attribute's mapping number (201 and 202 as the *vendor-offset* parameter is 200).

About RADIUS Attributes

The AP supports the following list of RADIUS attributes. Some are standard attributes and some are vendor-specific. Table 25 indicates the subsystem and whether it supports authentication processing or accounting for the attribute.

Table 25 Supported RADIUS Attributes

	L2TPTNL	PPP	IPSec	SSH
Standard Attributes				
User-Name	B	B	B	B
Password	B	B	B	B
Chap-PassWord		B		
NAS-IP-Address	B	B	B	Au
NAS-Port-ID	Ac	Ac	Ac	Au
Service-Type	B	B		
NAS-ID	B	B	B	Au
Actg-Status-Type	Ac	Ac	Ac	
Actg-Session-ID	Ac	Ac	Ac	
Actg-Session-Time	Ac	Ac	Ac	
Actg-Input-Octets	Ac	Ac	Ac	
Actg-Output-Octets	Ac	Ac	Ac	
Actg-Input-Packets	Ac	Ac	Ac	

	L2TPTNL	PPP	IPSec	SSH
Actg-Output-Packets	Ac	Ac	Ac	
Chap-Challenge		B		
NAS-Port-Type	Ac	Ac	Ac	
Framed-IP-Address		B	B	
Framed-IP-NetMask			B	
Framed-Route		B		
Standard Tunnel MIB Attributes				
Tunnel-Private-GroupID		B	B	
Tunnel-Password	B			
Tunnel-Client-Endpoint	Ac			
Tunnel-Server-Endpoint	Ac			
Tunnel-Client-Auth-ID	Ac			
Tunnel-Server-Auth-ID	Ac			
Vendor-Specific Attributes (Xedia)				
XEDIA-DNS-Server		B, 2	B	
XEDIA-NETBIOS-Server		B, 2	B	
XEDIA-Address-Pool		B		
XEDIA-PPP-Echo-Interval		B		
XEDIA-SSH-Privileges				B
XEDIA-Client-Access-Network			B	
Vendor-Specific Attributes (Ascend)				
ASCEND_Client_Primary_DNS		B		
ASCEND_Client_Secondary_DNS		B		
ASCEND_Client_Primary_WINS		B		
ASCEND_Client_Secondary_WINS		B		

Au indicates authentication processing only
Ac indicates accounting only
B indicates both

For standard attributes, refer to the relevant RFC for more details. For vendor-specific attributes, a single instance is retrieved unless otherwise indicated by a number following the B.

In addition to the vendor-specific attributes, make sure the following standard attributes are already defined:

```
User-Name                       1
Password                        2
Tunnel-Private-GroupID          81
```

About Client File and User File Entries

In addition to the dictionary file, the RADIUS servers usually have a client file that defines the secrets of the RADIUS clients (listed by IP address) and a user file that identifies individual user accounts. The client file might have these types of entries:

```
vanuatu                         Vanuatu
19.20.21.22                     xxxVanuatu
127.0.0.1                       meToo
```

The user file might have these entries:

```
mary  Password = "moretest"
      Service-Type = Login-User,
      Tunnel_Private_GroupID = "dialstrong"
```

Refer to your RADIUS server documentation for information about how to configure your server with this information.

Configuring the RADIUS Client

To use RADIUS authentication or accounting, you must configure the AP as a RADIUS client by performing these tasks:

- Configure the user authentication method.
- Make sure RADIUS is enabled on the AP.
- Configure authentication and accounting server entries on the AP.

CONFIGURING THE USER AUTHENTICATION METHOD

Users can be authenticated using internal authentication or RADIUS authentication methods. Internal authentication performs password validation using the user table on the AP. RADIUS authentication uses a configured RADIUS server for authentication. Refer to "Configuring User Authentication" on page 89 for more information about configuring user authentication methods.

To specify the RADIUS authentication method for particular users, use this command:

```
> config services authentication user.<user> \
  authentication-method radiusAuthentication
```

where <user> specifies the name of the user.

When using RADIUS authentication, you must enable the RADIUS client and configure it with at least one RADIUS authentication server entry.

ENABLING RADIUS

By default, RADIUS is enabled on the AP. To display the RADIUS client configuration, use this command:

```
> show services radius-client config
```

Make sure the Authentication Mode and Accounting Mode fields indicate enabled. Accounting Mode does not have to be enabled if you are not using the RADIUS accounting server.

To enable RADIUS on the AP, issue these commands:

```
> config services radius-client authentication-mode enabled
```

```
> config services radius-client accounting-mode enabled
```

Note that if you have changed modes from disabled to enabled and the operational state is down, you must reboot the AP to change the operational state to up. Check the Authentication Status and Accounting Status fields for the operational state by issuing this command:

```
> show services radius-client summary
```

In most cases, you just need to check the authentication-mode and accounting-mode parameters and make sure you have specified at least one authentication server configuration using the servers parameter (see "Configuring Server

Entries" on page 513). However, you can also configure these parameters on the AP for the RADIUS client configuration:

background-timer	Specifies the number of seconds between cleanups (including resetting servers from the not responding state to the unknown state). The default value is 100 seconds.
debug-command	Specifies the control strings for displaying RADIUS packets and error or trace messages on the console.
max-outstanding-requests	Specifies the maximum number of outstanding requests. The default value is 2.
nas-id	Specifies the name of the AP. If you do not specify a name, the system name of the AP is used.
nas-ip	Specifies the IP address of the AP. If you do not specify an address, 0.0.0.0 is used.

CONFIGURING SERVER ENTRIES

Server entries provide information about the RADIUS server configurations used by RADIUS clients. This information includes the IP address, priority setting, and shared secret key.

To configure authentication and accounting server configuration records, use these commands:

```
> config services radius-client servers authentication server.<instance>
```

```
> config services radius-client servers accounting server.<instance>
```

where <instance> is a number from 1 to 10 and you must specify the IP address or name and the shared secret key. For example:

```
> config services radius-client servers authentication server.1 \
  ip-address 10.20.30.40 shared-key itsasecret
```

If you do not specify an IP address, the name is used by DNS to look up the corresponding address. You can configure these parameters:

ip-address	Specifies the IP address of the server.
name	Specifies a name for identification purposes only, if you use an IP address. Otherwise, it specifies the DNS name.

priority	Specifies the priority of the server entry with a value from 1 to 10, where the lowest number is used first. The default value is 1. The server list is searched in a modified round-robin format starting with the last server used. NOTE: When searching the server list, the server with the lowest priority number is used first and the server at the next priority level is not used unless all the servers having the first priority level are not available. When matching the start and stop requests, this modified round-robin format is used to find a server for the stop request *only* if the server used for the start request is unavailable.
retries	Specifies the number of times a timed-out request is resent to the server. The default value is 2.
row-status	Specifies the status of the entry. If set to destroy, the row might be visible until there are no outstanding requests for this particular server.
shared-key	Specifies the shared secret key for the RADIUS server. It is not displayed to maintain security.
timeout	Specifies the number of seconds to wait for a response before a request is considered timed out. The default value is 2 seconds.
udp-port	Specifies the UDP port of the server to which requests are sent. The UDP port numbers for RADIUS Authentication servers are: • 1812 (official) • 1645 (unofficial) The UDP port numbers for RADIUS Accounting servers are: • 1813 (official) • 1646 (unofficial) The unofficial port numbers are used widely, but are incorrect. The default value is the official port number.
vendor-offset	Specifies the offset used to assign vendor-specific attribute IDs to the standard attribute number space. Setting this value to 0 (zero), the default value, forces the encoding of vendor-specific attributes within the vendor's number space. Setting this value to a non-zero value uses standard attribute encoding. If you enter a non-zero value, this number is used as the base value for the Xedia attributes in the dictionary file used by the RADIUS server. For example, if vendor-offset is set to 200, then Xedia attribute 1 would be 201 and Xedia attribute 2 would be 202 in the standard attribute number space.

DISPLAYING INFORMATION

You can display the following information about your RADIUS configuration:

- User information
- Accounting log information for the client
- Statistical information for the servers
- Summary information

DISPLAYING USER INFORMATION

If you want to monitor user authentication requests, you must set up an entry for the user being monitored and then review the information for that user.

To set up an entry for the user being monitored, use this command:

```
> config services authentication debug-a-user <user>
```

where <user> specifies the name of the user. To review the information for that user, you then issue this command:

```
> show services authentication debug-a-user debug-info <info type>
```

where <user> specifies the user name and <info type> has one of these values:

base-request	Returns the initial request sent to the authentication task from the subsystem.
intermediate-request	Returns the formatted version of the requests as needed by the authentication methods being used (internal or RADIUS).
response	Returns the result of the request, usually REJECT or ACCEPT. However, some additional attributes can be returned as well.

DISPLAYING ACCOUNTING LOG INFORMATION

If you want to monitor accounting information, use this command to specify the types of accounting messages that you want recorded for a session:

```
> config services authentication accounting-log-control <message type>
```

where <message type> is logStart (startup), logStop (stop), and logUpdate (update).

To display this accounting information, use this command:

```
> show services authentication user-accounting-log
```

Displaying Statistical Information

To display statistical information for the servers, use these commands:

```
> show services radius-client servers authentication server-table stats
```

```
> show services radius-client servers accounting server-table stats
```

The resulting table provides information about requests, retransmissions, responses, bad authenticators, timeouts, unknown types, bad packet IDs, and errors.

Displaying Summary Information

To display summary information for the client, use this command:

```
> show services radius-client summary
```

The resulting display provides statistical information for the RADIUS client that includes information about status, NAS ID, NAS IP, general errors, queuing, and requests.

To display summary information for the servers, use these commands:

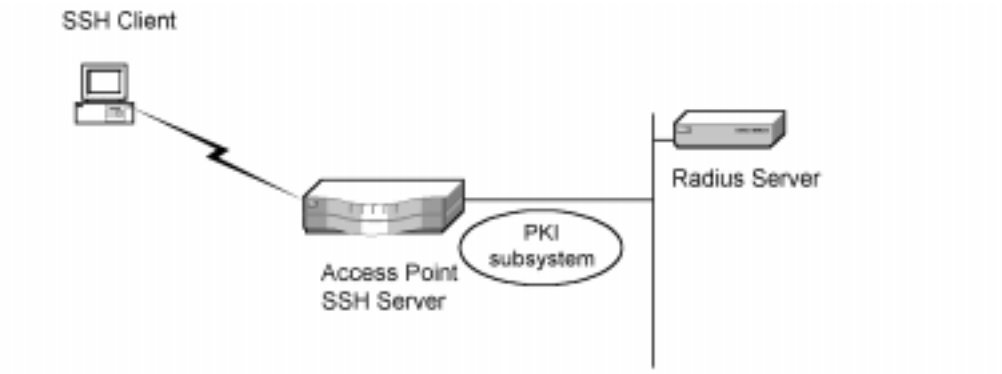
```
> show services radius-client servers authentication server-table summary
```

```
> show services radius-client servers accounting server-table summary
```

The resulting table provides information about the IP address, UDP port, roundtrip time, operational information, and last state change. The operational states are unknown, responding, not responding, and not available.

CONFIGURING AND USING SSH

The AP supports an SSH server and an SSH client for SSH v2.0. This support lets remote SSH clients (running on Windows, UNIX, or some other system) establish a secure SSH session with the AP's SSH server.



The SSH server provides encrypted, secure access to the Access Point. The SSH server can authenticate SSH users locally, through entries in its user table, or via entries resident on a RADIUS server. In addition, you can configure the SSH server to specifically accept access from or deny access to a list of designated hosts.

The AP's SSH server supports both host and user authentication. Even without authentication, all SSH communication with the AP is encrypted and checked for data integrity. After establishing a basic SSH configuration, you can add host and user authentication.

The SSH server also supports SSH ftp (SFTP). The SFTP application is used to support the SSH client's Secure Copy Protocol (SCP). See "Using the SSH File Transfer Protocol (SFTP)" on page 543 for instruction on configuring the SFTP options, and "Using the Secure Copy Protocol (UNIX System)" on page 548 for instruction on using SCP.

CONFIGURING THE SSH SERVER

Configuring the SSH server involves these steps:

1. Make sure that your IP interface is properly configured and SNMP user access is set.
2. Create a PKI key pair, and designate which pair will be used by the SSH server.

Strictly speaking, steps 3, 4 and 5 are optional. All the SSH server needs to accept client connections is a PKI key pair. However, you may want to configure users, user authentication, and host authentication as well.

3. Configure user authentication.
4. Configure SSH users.
5. Configure host authentication.

After reviewing these steps, refer to "What Happens During the SSH Connection Process" on page 526 for a discussion of how the SSH server authenticates users and grants them user privileges.

STEP 1: CONFIGURE THE AP'S INTERFACE

This step is required to allow any kind of connectivity to the AP. First, add an IP address to the appropriate AP interface, and configure SNMP communities to allow access for users from SSH clients. For information on adding an IP address, see "Configuring IP" on page 178. For information about configuring SNMP communities, see "Setting Up SNMP Communities" on page 38.

STEP 2: CREATE ONE OR MORE PKI KEY PAIRS

The SSH server presents a key to connecting SSH clients. If a DSA key already exists on the AP, creating another key is unnecessary. However, if a key does not exist, you need to add a DSA key for use by the SSH server as explained here.

A WORD ON PKI KEY PAIRS

Each PKI key pair comprises a *private* and *public* portion. While the private portion of the key is never sent over the network, the public portion is exchanged by clients and servers during session establishment. In the AP PKI subsystem, each PKI key pair you configure is indexed by a single name (such as a_ssh_key). The SSH server supports both DSA and RSA keys. However,

since all SSH clients support DSA, using the DSA algorithm is the best choice when creating keys for SSH.

Creating a PKI Key Pair

To create a DSA key pair named a_ssh_key, proceed as follows. Note that you have to provide a row-status of createAndGo for the key to become active.

```
> config services pki key-pair.a_ssh_key algorithm dsa length 1024 \
  row-status createAndGo
```

Use this command to display key information:

```
> show services pki key-pairs
```

```
                              SERVICES PKI KEY-PAIRS

                NAME| Algorithm| Length| Row Status|
--------------------|----------|-------|-----------|
           a_ssh_key|       dsa|    512|     active|
```

Configuring a Specific Key Pair for the SSH Server

It is best to configure a specific PKI key pair for use by the SSH server. If you do not do this, the server uses the first key it finds within the PKI subsystem.

To specify the PKI key for the SSH server, use this command:

```
> config services ssh-server keypair-name a_ssh_key
```

To check the operational status of the SSH server, issue this command:

```
> show services ssh-server summary
```

```
                            SERVICES SSH-SERVER SUMMARY

                 Oper Status  = upEnable
     Configured Encryption    = 3des-cbc,des-cbc
      Available Encryption    = 3des-cbc,des-cbc
Configured Authentication     = hmac-sha1,hmac-md5
 Available Authentication     = hmac-sha1,hmac-md5
   Configured Compression     = none
    Available Compression     = none
           Allowed Hosts      = ""
            Denied Hosts      = ""
     Key Types Presented      = ssh-dss
                Key Name      = a_ssh_key
```

STEP 3: CONFIGURE USER AUTHENTICATION

By default, user authentication is disabled (*none*) on the SSH server. With authentication disabled, the SSH server does not authenticate connecting users at all. However, it is far more common (and recommended) to configure some form of authentication. The sections that follow explain how to authenticate users with the user table or public key.

BEGIN BY ENABLING USER AUTHENTICATION

To authenticate users by user table or public key, enable user authentication on the SSH server with this command:

```
> config services ssh-server user-authentication \
  [none|passwordUserTable|publicKey|publicKeyORPassword]
```

Assuming a setting other than *none* (the default) has been chosen, the server attempts to authenticate connecting users by user table only (*passwordUserTable*), public key only (*publicKey*), or by either method (*publicKeyORPassword*).

For example, use the following command to enable user authentication by either public key or user table:

```
> config services ssh-server user-authentication publicKeyORPassword
```

User authentication is enabled so that the SSH server attempts whichever method the connecting client proposes. In most cases, public key is proposed first.

Creating at least one SSH user who can be authenticated by an internal password (in the user table) or by public key is strongly recommended to ensure that secure access to the AP is available.

ENABLING AUTHENTICATION WITH THE USER TABLE

The AP has a user table where you can specify authentication methods for particular users. To enable user authentication by user table only:

```
> config services ssh-server user-authentication passwordUserTable
```

See Page 521 for information about how to configure different authentication methods for SSH users in the user table.

Enabling Authentication With Public Key

The AP supports Public Key authentication for SSH users. To enable user authentication by public key only:

```
> config services ssh-server user-authentication publicKey
```

To configure the AP to perform Public Key authentication, you need to configure hosts and users, and their key data. See Page 523 for information about how to configure SSH users for Public Key authentication.

Step 4: Configure SSH Users

Configuring SSH users involves configuring their form of user authentication (that is, user table or public key) and their privileges.

Configuring Users for User Table Authentication

If you plan to authenticate users via the AP user table, you must configure user names and passwords for SSH users according to the chosen authentication method. Refer to "Configuring User Authentication" on page 89 for more information about configuring users in the authentication user table.

To configure users in the user table:

```
> config services authentication user.<name> password <password>
  ssh-privileges [noAccess|session|sftpRead|sftpReadWrite]
  authentication-method
  [internalAuthentication|alwaysAccept|alwaysReject|radiusAuthentication]
```

A sample set of entries is shown in Figure 74.

Figure 74 User Table with Sample Entries

SERVICES AUTHENTICATION USER-TABLE

NAME	Password	Client Group	Authentication Method
	Ssh Privileges	Row Status	
(default)	--------	(default)	radiusAuthentication
	noAccess	active	
bill	--------	(default)	alwaysAccept
	noAccess	active	
chris	--------	(default)	internalAuthentication
	noAccess	active	
davidc_new	--------	(default)	internalAuthentication
	noAccess	active	
davidc_old	--------	(default)	internalAuthentication
	session	active	
jim	--------	(default)	internalAuthentication
	noAccess	active	
mike	--------	(default)	alwaysReject
	noAccess	active	
ralph	--------	(default)	radiusAuthentication
	noAccess	active	

About Authentication Methods

If an authentication method has not been explicitly configured, the default method is the user table (*internalAuthentication*). The other settings are always allow access (*alwaysAccept*), always reject access (*alwaysReject*), and use the RADIUS database (*radiusAuthentication*).

To authenticate SSH users with RADIUS, you must configure appropriate user entries and access to a RADIUS server. If users are authenticated by RADIUS, the RADIUS server passes SSH privileges (and other information) back to the SSH server, provided an entry for the user exists in the RADIUS database. See "Configuring RADIUS Authentication" on page 507 for detailed information on RADIUS attributes, and RADIUS authentication in general.

About the (default) User

Note the presence of a (default) user in Figure 74. Like any user configured on the SSH server, the (default) user has a set of associated logon and authentication characteristics. The server applies these characteristics to users who are not found in the user table.

Here's how it works:

- Let's say a user named Jill at an unknown host attempts to start an SSH

session.

- If user authentication is configured to authenticate connecting users with the user table (see "Step 3: Configure User Authentication" on page 520), the server looks for Jill's information in the user table. In this case, Jill is not present. So the server looks for information on Jill in the RADIUS database.
- If the RADIUS database returns no information, the server applies the privileges associated with the (default) user in the user table to Jill. If *noAccess* is returned as the SSH privileges, the server drops the connection attempt.

While the (default) user's authentication method is *radiusAuthentication*, you can reconfigure this user like any other. However, for purposes of security, you should assign an authentication method of either *radiusAuthentication* or *alwaysReject* to the (default) user.

CONFIGURING USERS FOR PUBLIC KEY AUTHENTICATION

The Public Key authentication method for SSH users is based on a public key, which is shared, and a private key which is never transferred over the network. A connecting SSH client knows its own private key. When the client presents its key to the SSH server, the server validates the key information.

Many SSH clients have an application that generates public and private keys. Ordinarily the extensions for the key information are *.pub* and *.priv*, though sometimes the private key has no extension.

To configure the AP to perform Public Key authentication, you need to configure hosts and users, and their key data. Following is a sample command for user davidc on host 198.202.232.108. The public key portion of a client's key information is all that must be provided to the AP to complete the *key-data* attribute:

```
> config services ssh-server client-key.198.202.232.108 user.davidc key-data
  "AAAAB3NzaC1kc3MAAAABIQAAAIB5Ozaugublh5suolpltwC58+gky6ZjGo
  22PXmsnmKKFqcNf6HzpexsPyESzbcCQn0xUenn6M63Wu9zIENr36wg6P
  O5gm3R4ZdYgS+YFJOa3qlEXFO7yPXryPp8v5e6GZp0LxFylttNi1AJ5w/z
  mlwlho/Xv88LDp2LjL1tjH/Ew=="
```

You can show all client keys added to the SSH server with this command:

```
> show services ssh-server client-key-table summary

                    SERVICES SSH-SERVER CLIENT-KEY-TABLE SUMMARY

  CLIENT ADDRESS|             USER|         Key Data| Row Status|
----------------|-----------------|-----------------|-----------|
 198.202.232.108|           davidc|AAAAB3NzaC1kc3MA>|     active|
```

To show key data stored for a specific client address and user, use this command:

```
> show services ssh-server client-key.198.202.232.108 user.davidc

          SERVICES SSH-SERVER CLIENT-KEY.198.202.232.108 USER.DAVIDC

Key Data* = AAAAB3NzaC1kc3MAAACBAI+FRt4/hff8rI3VAS1pgHKSEkANnBNi67AOqUfSfDg/n\
            r+9jGPajFXODI2IBotqEeHnccQRGBFSF2gS5v7alxUQhIFDnk71p8+jwti2O+Xvon\
            QeOr8yVYhCpeabLhGKs8GroZ/+zjx9zyp3soIVZ+mEfG93Th89dNEzK1Dj9zupAAA\
            AFQCbuNKqjOxH6yMinkfeWwTxmeBNXQAAAIAzJ8sSeZNea6k1U2v9IzG95TAARPRI\
            HllrsKvNLKBKMRCQZydcvyZIA9y9hAW7KwlxVRzElpTUvSZkR8XEMVwCs1vBrv1IW\
```

CONFIGURING USER PRIVILEGES

The user table stores a wealth of information that determines access rights for all users. While the table is used for all client connections (such as SSH, L2TP, PPP, and IPSec clients), the Ssh Privileges field is specific to the SSH server.

The *ssh-privileges* parameter is particularly relevant with respect to the use of SFTP as its value indicates the connecting user's SFTP privileges. For detailed information on using this parameter, see "Configuring SFTP Access for Individual Users" on page 545.

Notice that in the sample screen shown in Figure 74, all users except davidc_old have an Ssh Privilege of *noAccess*. While it would appear that these users would be refused SSH access, this is not the case because the SSH server only checks the value of the *ssh-privileges* parameter if the *user-privilege-validation* parameter is enabled.

By default, the *user-privilege-validation* parameter is disabled. In this state, SSH users for whom *noAccess* (the default value) is set are allowed to establish an SSH session, provided they pass user authentication (by password or public key). See "Configuring SFTP Access on the AP" on page 544 for detailed information about this process.

STEP 5: CONFIGURE HOST LIST AUTHENTICATION

Host authentication enables you to accept or deny certain hosts, based on their inclusion in a host list. By default, host authentication is disabled. To turn on this feature, use this command:

```
> config services ssh-server host-authentication hostlist
```

To use host authentication, you need to configure accept and deny lists containing specific hosts as explained here.

ACCEPTING AND DENYING SPECIFIC HOSTS

You may want to configure the AP to accept or deny access from and to specific hosts, or host ranges. To accomplish this, you must turn on host authentication, and create appropriate accept or deny lists.

Configure the host-accept and host-deny lists, as follows:

```
> config services ssh-server accept-hosts \
  198.2.3.4,189.3.4.5,202.223.198.0-202.223.198.255

> config services ssh-server deny-hosts
  190.2.3.4,172.202.215.5,232.223.198.0-232.223.198.255
```

Each list is a string allowing for entry of 255 characters. You can designate hosts as a comma-separated list, a dash-separated range, or both. You can also enclose a list of hosts and host ranges with double-quotes.

USING ACCEPT AND DENY LISTS TOGETHER

Using the accept and deny hosts lists together makes it easy to identify a single subnet as accepted, and then deny a set — or several — specific hosts. For instance, you may want to allow an entire subnet, as follows:

```
> config services ssh-server accept-hosts 198.234.0.0-198.234.255.255
```

Now, prune a few users from the list of accepted hosts using the deny list, like this:

```
> config services ssh-server deny-hosts 198.234.126.0-198.234.126.255
```

This disallows users from the 126 subnet, while allowing other users from the 198.234 subnet. If necessary, you can refine the list of denied hosts further by specifying individual IP addresses. This command adds two additional IP addresses to the denied range specified earlier:

```
> config services ssh-server deny-hosts \
  198.234.126.0-198.234.126.255,198.234.198.196,198.234.198.199
```

USING THE MINUS SIGN WHEN SPECIFYING ACCEPTED HOSTS

Another way of denying specific users or ranges is by using the minus sign and an IP address (or ranges) when specifying allowed hosts. Here is a sample, based on the allow hosts command used above. A single host has been pruned from the list of those allowed, using a minus sign:

```
> config services ssh-server accept-hosts 198.2.3.0-198.2.3.255,-198.2.3.211
```

After configuring a set of allowed and denied hosts, you can get an operational view of the hosts the SSH server will allow by checking the summary screen. The summary screen provides an exact specification of the hosts that are allowed and denied. To access this screen, use this command:

```
> show services ssh-server summary
```

```
                        SERVICES SSH-SERVER SUMMARY

              Oper Status  = upEnable
    Configured Encryption  = 3des-cbc,des-cbc,none
     Available Encryption  = 3des-cbc,des-cbc,none
Configured Authentication  = hmac-sha1,hmac-md5
 Available Authentication  = hmac-sha1,hmac-md5
   Configured Compression  = none
    Available Compression  = none
            Allowed Hosts  = 198.202.232.131, 198.202.232.108
             Denied Hosts  = ""
      Key Types Presented  = ssh-dss
```

WHAT HAPPENS DURING THE SSH CONNECTION PROCESS

For any user, the SSH server can be configured to perform the following steps in sequence:

1. First, the SSH server authenticates the connecting user.
2. Next, the server checks the setting of the *global-access-mode* parameter. To establish an SSH session, this parameter's setting must be something other than *noAccess*. See "Setting the Global Access Level for SFTP" on page 544 for information on setting this parameter.
3. Next, providing *user-privilege-validation* has been set to *enabled*, the server checks the user's privileges.

It is important to understand what each of these steps entails. Depending on

how the SSH server is configured, values provided in the user table, a RADIUS database, and the values associated with the (default) user may all come into play in determining:

- Whether a user is allowed to establish an SSH session, and
- The level of access that user is granted.

It is easiest to explain how these factors interact by focusing on a single user, as in the sections that follow.

Authenticating Jill Through the User Table: How It Works

Unless you have configured SSH to use Public Key authentication (as explained on Page 523), the authentication process begins at the user table. Stepping through a sample scenario is helpful in understanding how this process works.

Let's say a user named Jill attempts to connect via SSH. If Jill's information is not found in the user table, the server uses the configured characteristics of the (default) user to authenticate the connection. If no (default) user exists or the (default) user is inactive, the server drops the connection attempt. This is one possible scenario.

However, another is that the (default) user's authentication method is *radius-Authentication*. In this case, the server looks for Jill's information on a RADIUS server. If this server returns attributes for Jill, Jill is authenticated by RADIUS password and assigned a level of SFTP access. If RADIUS returns no information, the SSH server checks the SSH privileges of the (default) user to determine whether the session can be established.

Checking the Global Access Mode

As mentioned earlier, the *global-access-mode* parameter must be set to allow access for an SSH session to be established. By default, this parameter is set to *session*. A setting of session allows SSH access to the AP login screen.

You can set the *global-access-mode* parameter to *session* with this command:

```
> config services ssh-server global-access-mode session
```

Other settings include *sftpRead* and *sftpReadWrite*. These settings relate to SFTP access. They are discussed on Page 544.

CHECKING JILL'S SFTP PRIVILEGES

Assuming Jill has been authenticated by RADIUS, the server can return a RADIUS attribute granting SFTP privileges of either *noAccess*, *session*, *sftpRead*, or *sftpReadWrite*. At least session access is needed for Jill to have any SSH access at all.

Let's say, however, that Jill is successfully authenticated via RADIUS, but that no SFTP privileges are returned. In this case, the SSH server checks the user table for Jill's SFTP privileges. If Jill does not have an entry in the table, the server uses the SFTP privileges assigned to the (default) user.

ENSURING SECURITY AND FLEXIBILITY

It is a good idea to configure one "superuser" to be authenticated internally or to be authenticated by public key. This ensures that this user will have access to the AP in the event the RADIUS server is unreachable.

If you want to ensure that no unauthorized or unknown user has access to the AP, make sure the SSH server checks user privileges. To do this, set *user-privilege-validation* to *enabled* as follows:

```
> config services ssh-server user-privilege-validation enabled
```

Enabling this parameter causes the server to check each user's SSH privileges (as captured in the password user table) before establishing an SSH session. Then, configure the (default) user with *noAccess*. The combination of these measures provides both security and flexibility: only RADIUS authenticated users are allowed access, except for one superuser who is locally authenticated. In addition, the SSH server applies the characteristics of the (default) user to all unknown users, and grants them no SSH access at all.

A SAMPLE SSH SERVER SESSION (UNIX)

A sample session from a UNIX client to the SSH server appears below. In this example, user authentication has been enabled on the SSH server (as explained on Page 520). The *-l* option is used to specify the user name, followed by the address of the target system.

Upon receiving the user name, the SSH server prompts for a password, as shown:

```
> ssh -l chris 198.202.232.159
```

```
chris's password:

AccessPoint SSH (SSH-2.0)


        Lucent AccessPoint (hw: 1.7; sw: V2.3.0.A15.QVPN 03/07/2000 17:33:06;

Username: admin
Password:

You are logged in with read-write access.
Enter ? or 'help' for commands.

[ initialization files:  flash/.admin.tcl  ]
```

If chris were configured to use Public Key authentication (as explained on Page 523), the public/private key exchange would be used to authenticate him. In this case, chris would not be prompted for a password.

SSH Server Configuration Settings

The SSH server is designed to provide a great deal of flexibility in the administration of secure access to the AP. To this end, the server uses a number of default values to make initial configuration easy, and to establish a reasonable baseline for interactions between SSH clients and the AP. All SSH communication with the AP is encrypted and checked for data integrity, whether or not user authentication has been configured.

admin-status (Default: enabled)

Specifies the administrative status of the SSH daemon.

- enabled

  Indicates that the SSH daemon will process requests from clients. If there are enough node resources and the host/user pass authentication, then the user is allowed to log into the CLI or perform SFTP operations as their SSH privileges permit.

- disabled

  Indicates that the SSH daemon will not process requests from clients. Existing sessions are not torn down.

- disabledDown

  Indicates that the SSH daemon will not process requests from clients, and, at system initialization, the SSH daemon will *not* be started. This can be used to free up node resources if SSH is not going to be used. Existing ses-

sions are not torn down.

The SSH daemon can only be started at initialization time. So, if it was set to *disabledDown* and the daemon never started, a full node boot is required to enable the SSH daemon.

TCP-PORT (DEFAULT: 22)

Specifies the TCP port number that will be used to listen for requests. Changing this parameter requires a reboot to take effect.

SESSION-EVENT-MODE (DEFAULT: ENABLED)

Controls overall session event generation used with the normal system event filters. If the session flag is disabled, no session events are sent. If the flag is enabled, the normal event filter mechanisms are still used to customize input or output of the events.

USER-AUTHENTICATION (DEFAULT: NONE)

The user authentication mode. If *passwordUserTable* is used, then the user will be looked up and authenticated using the general authentication user table and the specified authentication method. If *publicKey* is used, then the client/user public key will be looked up in the SSH Client Key table.

HOST-AUTHENTICATION (DEFAULT: NONE)

The authentication method for client machines. If *hostList* is used, then the client address is validated against the accept/deny list of clients.

GLOBAL-ACCESS-MODE (DEFAULT: SESSION)

Controls overall access to the SSH daemon for any given user. SSH access includes normal CLI login, as well as SSH FTP read/write. At a minimum, the *session* level is required for any SSH session establishment, such as a normal CLI login.

- *noAccess* - SSH daemon is running, but no users are allowed to establish a session.
- *session* - normal CLI login is allowed, but no SSH file transfer protocol (SSH FTP) is allowed.
- *sftpRead* - normal CLI login is allowed as well as SSH FTP read-only access to files.

- *sftpReadWrite* - normal CLI login as well as SSH FTP read/write access to files.

USER-PRIVILEGE-VALIDATION (DEFAULT: DISABLED)

Controls whether the SSH server checks individual user privileges for SSH sessions and SFTP read/write operations. By default, this parameter is disabled. User privileges are configured under the authentication subsystem, through the password user table or in a RADIUS database.

If the user authentication mode is *none* or *publicKey*, then the user/password table privileges for the specified user are used if the entry exists. If the entry does not exist, but a (default) user exists, then the default entry privileges are used. If there is no specific user or default entry, no privileges are allowed.

The user authentication mode of *passwordUserTable* works in the same manner, except the user table entry can be overridden by a RADIUS attribute if RADIUS authentication is used for the individual or default user. The expected attribute (XEDIA-SSH-Privileges) is vendor-specific (Xedia #5). The server can return RADIUS attribute values granting SFTP privileges of either *noAccess* (1), *session* (2), *sftpRead* (3), or *sftpReadWrite* (4).

ENCRYPTION (DEFAULT: DES + DES3)

Controls the ciphers that can be used during negotiation with a client.

AUTHENTICATION (DEFAULT: SHA + MD5)

Controls the HMAC algorithms that can be used during negotiation with a client.

COMPRESSION (DEFAULT: NONE)

Specifies the types of compression that can be negotiated.

MAX-AUTHENTICATION-TIME (DEFAULT: 20 SECONDS)

If you configure the server to authenticate SSH clients, they must do this within 20 seconds. If not, the SSH server terminates the connection.

To change the time allotted to users authenticating themselves, use this command:

```
> config services ssh-server max-authentication-time <number>
```

You may want to bump up this time to a higher value if users are timing out

while authenticating themselves to a RADIUS server. Such timeouts can be caused by network congestion, particularly if the SSH server is authenticating users via a RADIUS server or if the user is allowed multiple retries.

KEYPAIR-NAME (DEFAULT: NONE)

The name of the PKI key pair that is configured in the box for use by SSH. The key should be DSA formatted. If this index is left blank, the first available key will be used.

IDLE-TIMEOUT (DEFAULT: 2 MINUTES)

By default, all idle connections are terminated after two minutes to help protect the AP against unauthorized access (as from someone walking away from an open session).

To change the idle timeout value, use this command:

```
> config services ssh-server idle-timeout <number>
```

A value of 0 (zero) disables the timeout feature, allowing an idle session to remain open indefinitely.

PASSWORD-RETRY (DEFAULT: 2)

In addition, the server allows 2 default retries for users who fail to authenticate themselves properly on the first attempt. Again, the intent is to allow for a few mistakes, without compromising security. To change the number of retries allowed:

```
> config services ssh-server password-retry <number>
```

If you change number to 0 (zero), no retries are permitted.

MAX-SESSIONS (DEFAULT: 2)

By default, the maximum concurrent SSH sessions the server will allow is 2. This value can be reset to any number from 1-9. Whether the server will actually support a large number of sessions depends on the availability of system resources.

To change the maximum number of concurrent SSH sessions:

```
> config services ssh-server max-allowed-sessions <number>
```

TUNING THE SSH SERVER CONFIGURATION

SSH clients propose connection parameters, such as encryption and authentication type, when connecting to the server. However, the server determines what parameters to use to authorize a connection. If a client proposes parameters that the server is not configured to accept, the server denies the connection.

The SSH server summary screen provides information on the configured options that are available within the software image, and on those that have been configured.

```
> show services ssh-server summary
```

```
                        SERVICES SSH-SERVER SUMMARY

                Oper Status  = upEnable
     Configured Encryption   = 3des-cbc,des-cbc
      Available Encryption   = 3des-cbc,des-cbc
Configured Authentication    = hmac-sha1,hmac-md5
 Available Authentication    = hmac-sha1,hmac-md5
   Configured Compression    = none
    Available Compression    = none
           Allowed Hosts     = 1.1.1.0-1.1.1.25, 1.1.1.31-1.1.1.255
            Denied Hosts     = ""
```

This screen indicates which configured options are available, based on what has been configured. Available options are those supported by the software image. Configured options include only those that the server can present to a connecting client. Often, available and configured options are the same.

This screen indicates that 3DES and DES encryption are available, along with HMAC-SHA1 and HMAC-MD5 authentication.

You can alter the SSH server configuration as needed. For example, to configure the server to accept only 3DES encryption, proceed as follows:

```
> config services ssh-server encryption !none+!des
```

You can configure other options in the same manner.

SHOWING INFORMATION ON THE SSH SERVER

The AP console provides access to detailed information on the SSH server configuration and SSH sessions. To show information on SSH sessions, use this command:

```
> show services ssh-server session.* summary
```

```
                    SERVICES SSH-SERVER SESSION.* SUMMARY

 INDEX|           Remote Ip|          Remote Node|      Remote User|
      | Negotiation Results|         Session Time|      Ui Bytes In|
      |        Ui Bytes Out|        Sftp Bytes In|   Sftp Bytes Out|
      |          Row Status|
------|--------------------|---------------------|-----------------|
     1|     198.202.232.131|      bioko.xedia.com|            chris|
      |C->S: 3des-cbc hmac>|  86959 (00:14:29.59)|               56|
      |                1743|                    0|                0|
      |              active|
```

To show counters, use this command:

```
> show services ssh-server counters
```

```
                          SERVICES SSH-SERVER COUNTERS

        Current Sessions  = 1
  Total Session Requests  = 1
     Completed Requests  = 1
Authentication Failures  = 0
      Resource Failures  = 0
      Total Client Keys  = 0
     Active Client Keys  = 0
```

CONFIGURING THE SSH CLIENT

The AP supports an SSH client, allowing for secure connections to other SSH servers. All SSH client configuration options are available under the SSH client subsystem. To display all parameters, enter this command and press the spacebar:

```
> config services ssh-client
```

If the SSH server uses Public Key authentication, the client needs a PKI key. However, SSH server can also authenticate a user by means of a password. In this case, no PKI key is needed. If you fail to specify a user name when connecting from the client, the username from the AP's current login session is used. Thus, if you have logged in to the AP's console as *Joe*, the SSH server

prompts you for a password.

The SSH client requires no configuration. However, you may want to configure a specific host authentication mode and user authentication as described in the sections that follow.

Choosing a Host Authentication Mode

You can configure the SSH client to use any of three modes to check public key information when establishing a connection. These modes are *none*, *easyHostKeyCheck*, and *strictHostKeyCheck*. To set key check mode for the client to *easyHostKeyCheck*, use this command:

```
> config services ssh-client host-authentication-mode easyHostKeyCheck
```

The default host authentication mode value is *none*. When *none* is the setting, the client always accepts the host key presented by the server. When the mode is set to *easyHostKeyCheck*, the client compares the key data received from the server with the key data it has stored during a previous connection. If this data does not exist, data obtained from the server is stored and the connection is allowed. If the data exists, but does not match, the client disallows the connection.

When the mode is set to *strictHostKeyCheck*, the client always compares data obtained at connection time with key data it has stored. If no data is stored or this data does not match with data received from the server, the connection is disallowed.

User Authentication

User authentication is performed on the SSH server, rather than the client. You can configure the SSH client to authenticate itself by password rather than public key. If you elect to do this, the server prompts the SSH client for a password at session setup. A sample session is shown here:

```
AP-System> ssh 10.1.3.97

...Connecting.


----------------------------------------
Password Required for remote SSH access
        User: admin
 Remote Host: 10.1.3.97
----------------------------------------
Enter Password: ........

AccessPoint SSH (SSH-2.0)


        Lucent AccessPoint (hw: 1.7; sw: V2.3.0.A58.QVPN 05/07/2000 00:02:01:00)

Username: admin
Password:

You are logged in with read-write access.
Enter ? or 'help' for commands.
```

Notice that the user is listed as admin in the password prompt at the top of the sample screen. This is because the SSH client automatically uses the login name used within the native CLI session when connecting to an SSH server. In this case, user admin enters the correct password (stored in the SSH server's password user table or a RADIUS database) and the SSH session opens.

Note that the client may not be prompted for a password at all. This is the case if the SSH server is configured to perform no authentication, or is configured to authenticate clients using public key.

See Page 521 for information on configuring usernames, passwords, and authentication methods on the AP's SSH server.

CONTROLLING TCP SESSION RETRIES DURING CONNECTION

As the previous example shows, a ...Connecting message appears during session establishment. The three dots indicate the maximum number of TCP connection retries to be attempted while bringing up the session. Three retry attempts (indicated by the three dots) is the default. If you are running on a slow network, you can increase the maximum number of retry attempts (for example, to 9) using the *-r* argument as shown here:

```
> ssh -r 9 10.1.3.97
```

The Connecting message reflects the maximum number of retry attempts specified (as shown by the dots preceding the message) and the actual number of attempts made (as shown by the dots trailing the message):

```
.........Connecting.


----------------------------------------
Password Required for remote SSH access
       User: admin
 Remote Host: 10.1.3.97
----------------------------------------
```

USING PUBLIC KEY AUTHENTICATION

Unlike the SSH server, the SSH client only requires a PKI key pair if Public Key authentication is used to authenticate the session. Public Key authentication is highly secure, and makes exchange of passwords between the SSH client and server unnecessary.

To allow the SSH client to use Public Key authentication, you need to configure a host/user combination and associated key data as explained below. The host/key combinations and public key data you configure define the host addresses of SSH servers to which the client can connect, the users authorized to make the connection, and the public key matching the server's Private Key.

CONFIGURING HOST KEYS

To configure host keys and key data for the client, use this command. The example shows the addition of user davidc on host 198.202.232.108:

```
> config services ssh-client host-key.198.202.232.108 user.davidc key-data
  <key data>
```

To add key data, cut and paste the public key information as explained in "Configuring Users for Public Key Authentication" on page 523.

You can show all host keys and users with this command:

```
> show services ssh-client host-key-table
```

```
                        SERVICES SSH-CLIENT HOST-KEY-TABLE

    HOST ADDRESS|              USER|         Key Data| Row Status|
----------------|------------------|-----------------|-----------|
 198.202.232.108|            davidc|AAAAB3NzaC1kc3MA>|     active|
```

Note that the key data, users, and host addresses shown in this example are fic-

titious.

ADDING KEY DATA

One easy way to add key data for users is to set the host authentication method to *easyHostKeyCheck*. Doing so causes the SSH client to capture and cache the public key information at session setup. However, if you elect to use this method for expediency, it is best to make a subsequent out-of-band check on the host key data before proceeding further. The goal of this check is to ensure that the SSH client has captured the correct key data from the SSH server to which it is attempting to connect. This ensures that the client is not actually connecting to a system that is masquerading as (spoofing) an SSH server.

A SAMPLE SSH CLIENT SESSION

Once you configure the client, you can SSH from the AP to another SSH server (including, of course, the SSH server residing on another AP system). The specific nature of the client's interaction with the SSH server depends on the server's configuration. The sample session that follows uses the *-l* argument to pass a user name to the server. Since user authentication has been enabled on the AP SSH server, the server prompts the client for a password:

```
> ssh -l chris 10.1.3.97

...Connecting.


----------------------------------------
Password Required for remote SSH access
       User: chris
 Remote Host: 10.1.3.97
----------------------------------------
Enter Password: ....

AccessPoint SSH (SSH-2.0)


       Lucent AccessPoint (hw: 1.7; sw: V2.3.0.A58.QVPN 05/07/2000 00:02:01:00)

Username: admin
Password:

You are logged in with read-write access.
Enter ? or 'help' for commands.
```

Another scenario is to configure the SSH server with user authentication disabled. This is the case in the following example. As a result, the server does

not prompt the client for a password.

```
> ssh 10.1.3.97

...Connecting.

AccessPoint SSH (SSH-2.0)

        Lucent AccessPoint (hw: 1.7; sw: V2.3.0.A58.QVPN 05/07/2000 00:02:01:00)

Username: admin
Password:
```

Once you establish an SSH session, it functions exactly as a Telnet or console session, but is encrypted by SSH. Like the SSH server, the SSH client depends upon the exchange of the public portion of a PKI key pair to establish a secure session. By default, the client performs no checking on the key presented to it by the SSH server to which it is connecting. However, you can configure the client to perform two different levels of key checking. (Both *easy* and *strict* key check modes are supported.) In addition, you can cause the client to authenticate by password or use Public Key authentication.

Host authentication and user authentication are available:

- Host authentication

  Consists in the comparison of key information presented by the server with key information stored on the client. *By default, host authentication is not performed.*

- User authentication

  The SSH client authenticates itself by password.

SSH Client Configuration Settings

Like the SSH server, the SSH client uses a number of default values to make initial configuration easy, and to establish a reasonable baseline for interactions between SSH clients and the AP.

host-authentication-mode (Default: none)

Specifies the type of host key checking the SSH client performs on key data received from the SSH server. A value of *none* means the client always accepts the data as presented. *EasyHostKeyCheck* causes the client to compare the key with an existing key (if one is known) or to accept the key as is (if one is not

known). *StrictHostKeyCheck* always performs a comparison with known data. If no key is found, the key is automatically rejected.

HOST-KEY-TABLE

Specifies the key data for the associated host and user.

Note that any configuration changes made using this parameter affect *all* entries in the host key table. For example, use this command to deactivate all the hosts and users:

```
> config services ssh-client host-key-table row-status notInService
```

You should be very careful when using this parameter because it affects all the SSH clients.

IDLE-TIMEOUT (DEFAULT: 2)

Specifies the timeout interval (in minutes) that the SSH client allows before disconnecting an idle session. Setting this value to 0 (zero) disables the idle timeout feature.

KEYPAIR-NAME (DEFAULT: NONE)

Specifies a particular PKI key pair for use by SSH. If no value is provided, SSH uses the first available key.

SESSION-EVENT-MODE (DEFAULT: ENABLED)

Specifies the overall control of session event generation that is used with the normal system event filters. If the session flag is disabled, no session events will be sent. If the flag is enabled, the normal event filter mechanisms are still used to customize input/output of the events.

SSH CLIENT COMMAND LINE OPTIONS

The default values used to configure the SSH client on the AP are usually reasonable for setting up sessions, but you can use command line options to override the SSH client configuration for a particular session.

To get help on SSH command options, enter this command at the CLI:

```
> ssh -h
```

To get more verbose help, use an upper case H, as follows:

```
> ssh -H
```

Table 26 lists the command options.

Table 26 SSH Command Options

Command Option	Use	Supported Values
-l <name>	Log on as specified user.	User name
-c <cipher>	Use specified cipher.	3des-cbc, des-cbc, or none
-t <time>	Specify the idle timeout value in minutes. Overrides the AP's SSH client configuration.	0 - 1440, default 2
-i <time>	Specify the initialization timeout value in seconds.	10 - 180, default 40
-w <time>	Specify the maximum time (in seconds) to wait for user password input. This value is used to attempt authentication before processing the initialization timeout value (set by the -i option).	10 - 60, default 30
-m <algorithm>	Specify MAC authentication algorithm.	hmac-sha1, hmac-md5, or none
-a <auth mode>	Specify the authentication mode to use.	publicKey, password, or publicKeyORpassword
-C <compression>	Specify compression to use.	zlib or none
-v	Use verbose mode to display command line information.	—
-V	Display only version information.	—
-p <port number>	Specify the TCP port to use.	Port number, default port is 22
-k <key>	Specify the PKI key pair name to use for Public Key authentication. Overrides the AP's SSH client configuration.	Key pair name
-h	Display Help text.	—
-H	Display verbose Help text.	—

Command Option	Use	Supported Values
-P <password>	Specify user password to present to the SSH server. This password is used for the first authentication attempt. If another attempt is needed, the user is prompted for another password.	Password
-r <retry number>	Specify the number of TCP connection retries before timing out. Setting this value to 0 (zero) blocks the TCP connection. This may take up to 75 seconds to time out.	1 - 10, default 3

USING SCRIPTS FOR SSH CLIENT CONFIGURATION

The default values used to configure the SSH client on the AP are usually reasonable for setting up sessions, but you can use internal TCL environment variables to pass optional information automatically into the SSH client. Although these environment variables let you use TCL scripts to set up the SSH client configuration, range checking and error reporting is limited. When you specify incorrect values for variables, the incorrect values are ignored and the default values are used instead.

The following table lists the TCL environment variables.

Variable	Use
SSH_USER	Supply SSH with a specific user.
SSH_PASSWORD	Supply SSH with a password without prompting.
SSH_COMMAND	Specify command to execute on remote end.
SSH_CIPHER	Specify cipher.
SSH_MAC	Specify authentication algorithm.
SSH_COMP	Specify compression.
SSH_AUTHMODE	Specify authentication mode.
SSH_KEYNAME	Specify PKI key pair name used for Public Key authentication.
SSH_IDLETIMEOUT	Specify idle timeout of the client session in minutes. This value overrides the global value.

Variable	Use
SSH_INITTIMEOUT	Specify maximum time (in seconds) allowed for authentication.
SSH_PASSWAIT	Specify maximum time (in seconds) to wait for password input.
SSH_TCPRETRY	Specify the number of TCP connection retries.

About the SSH Server Applications

The AP supports two applications that use the SSH protocol as an underlying transport.

- SSH ftp (SFTP) allows you to securely put files to, or get files from, the AP's storage areas from a remote workstation. See Page 543 for information on using SFTP.
- Secure Copy Protocol (SCP) enables you to securely copy files to and from the AP from a remote workstation, or to copy a file from one of the AP storage areas to another. See Page 548 for information on using SCP.

Using the SSH File Transfer Protocol (SFTP)

The SSH file transfer protocol (SSH ftp, or SFTP) is an FTP-like application that uses SSH as its underlying transport. SFTP works in the same way as the FTP protocol, and can be used to put and get files in similar fashion. The difference is that SFTP sessions are encrypted and authenticated, owing to the use of SSH as an underlying transport.

In similar fashion, another SSH protocol called SCP (Secure Copy Protocol) uses SFTP as its underlying transport. This enables you to use SCP to copy files remotely — and securely — to or from flash or NVRAM, either from another system or from one storage area to another (that is, within-the-box copying). See Page 548 for detailed information on using SCP.

A Note on the AP File System

Typical uses of SFTP include putting files to, or getting files from the AP's storage areas (NVRAM and flash) from a remote system. It is therefore important to understand how the AP file system is structured.

The AP implements a simple file system. The root directory is /ap. Under this

directory are two storage directories: /ap/flash and /ap/nvram. You can only store configuration and TCL files in /flash or /nvram. You cannot store files under the root directory /ap.

In addition, SFTP only supports 'flat' local file names. In other words, you cannot use SSH ftp to put or get files that contain a slash (/).

CONFIGURING SFTP ACCESS ON THE AP

As mentioned earlier, SFTP runs over SSH. You can see whether SSH is running by entering this command:

```
> show services ssh-server config
```

Check to see that Oper Status is *upEnabled*. In addition, check to make sure that a PKI key pair exists on the server by issuing this command:

```
> show services pki key-pairs summary
```

If SSH is running, configuring SFTP involves these three steps:

1. Setting the *global-access-mode* parameter to *sftpRead* or *sftpReadWrite*.
2. Setting the *user-privilege-validation* parameter to *enabled*.
3. Configuring SFTP access for individual users.

SETTING THE GLOBAL ACCESS LEVEL FOR SFTP

The *global-access-mode* parameter applies to all users. For SFTP access to be permitted to anyone, the parameter must be set to either *sftpRead* (for read only access) or *sftpReadWrite* (for read/write access).

This parameter's default value is *session*. With this value, only SSH session access is allowed, and all SFTP attempts are dropped. To set the parameter to allow read/write access for all users, use this command:

```
> config services ssh-server global-access-mode sftpReadWrite
```

Keep in mind that this is a global parameter, rather than a setting that applies to individual users. Setting the parameter as shown here is only the first step needed to allow SFTP. The next step is to configure an access level for individual users, as explained below.

ENABLING USER PRIVILEGE VALIDATION

The *user-privilege-validation* parameter controls whether the SSH server checks the SSH privileges configured for SSH client users. By default, this parameter is disabled, in which case the server performs no checking. If you enabled the parameter, the server checks the SSH privileges assigned to each SSH user.

To enable the parameter, use this command:

```
> config services ssh-server user-privilege-validation enabled
```

You *must* enable this parameter and configure the appropriate SFTP access for specific users to allow them to use the SFTP application.

CONFIGURING SFTP ACCESS FOR INDIVIDUAL USERS

You can define an SFTP access level either while defining users under the authentication subsystem, or later. To do this, set the *ssh-privileges* parameter to either *sftpRead* or *sftpReadWrite*. By default, this parameter is set to *noAccess*. This sample command adds a user named *bill* with the password *rogers.* His SSH privileges are set to *sftpReadWrite*. Provided the global access mode is set appropriately and user privilege validation is enabled, Bill receives SFTP read/write access to the SSH server.

```
> config services authentication user.bill ssh-privileges sftpReadWrite \
  password rogers
```

An access level of *sftpRead* allows a user to get a file from a remote system; *sftpReadWrite* access allows a user to get files from, and put files to a target system. The default access level (*noAccess*) disallows SSH connections to the target system. Since SFTP uses SSH as its transport, this implicitly denies all SFTP connections as well. Provided that the global access mode has been set properly, the level of access granted to Bill allows him to put and get files to and from the AP's storage areas.

VERIFYING USER CONFIGURATION

You can verify the information associated with any user with this command. Notice that user Davidc_old has session level access. Session level access allows Davidc_old to connect to the AP login screen. He cannot read or write files using SFTP. In the screen shown here, only Bill, Chris, and Davidc_new have SFTP privileges. Their privileges allow SFTP read only (*sftpRead*) rather than read/write (*sftpReadWrite*):

```
> show services authentication user-table summary
```

```
                        SERVICES AUTHENTICATION USER-TABLE

            NAME|        Password|     Client Group| Authentication Method|
                |   Ssh Privileges|       Row Status|
----------------|-----------------|-----------------|----------------------|
       (default)|         --------|        (default)|  radiusAuthentication|
                |         noAccess|           active|
            bill|         --------|        (default)|           alwaysAccept|
                |         sftpRead|           active|
           chris|         --------|        (default)|internalAuthentication|
                |         sftpRead|           active|
       davidc_new|         --------|        (default)|internalAuthentication|
                |         sftpRead|           active|
       davidc_old|         --------|        (default)|internalAuthentication|
                |          session|           active|
             jim|         --------|        (default)|internalAuthentication|
                |         noAccess|           active|
            mike|         --------|        (default)|           alwaysReject|
                |         noAccess|           active|
           ralph|         --------|        (default)|  radiusAuthentication|
                |         noAccess|           active|
```

STARTING AN SFTP SESSION (UNIX SYSTEM)

To start an SFTP session, issue this command. In this example, chris is the user name (which the AP authenticates during connection establishment) and 198.234.5.6 is the IP address of the AP. If user chris were configured to be authenticated by public key, the server would not prompt for a password.

```
> sftp 198.234.5.6 chris

  local path     :  /home/chart

chris's password:

  remote path  : /ap

sftp>
```

If the SSH server prompts for a password, failure to provide it generates a failure message like the following:

```
Disconnected : authentication error (Authentication method disabled.).

Remote path  : error : connection was lost.
```

Once you establish an SFTP session, you can change your remote path to either of the two storage areas, and put or get files securely from your client's

local path. Table 27 provides a list of the basic commands to use. They are similar to those employed when using ordinary FTP.

Table 27 Common SFTP Commands

Command	Use
pwd	Show current directory.
cd	Change to named directory.
cd..	Back up one directory level.
ls	List contents of the AP directory.
get	Get a copy of a file from the AP to the local directory.
put	Put a copy of a file to the AP.
lpwd	Local machine, show current directory.
lcd	Local machine, change directory.
lls	Local machine, list directory contents.
rm	Remove a file.

EXAMPLE: PUTTING AND GETTING FILES

The AP contains two main storage areas for files: flash and NVRAM. Establishing an SFTP session places you in the top level directory, ap. Here is a typical session showing a directory change to flash, and put and get operations for several files.

```
sftp> pwd
/ap
sftp> ls
flash    nvram
sftp> cd flash
sfftp>pwd
/ap/flash
sftp>ls
admin.tcl      fullboot.bin     test.tcl
config.file    mips.2.2.bin     test1.tcl
```

These commands move you to the directory from where you launched SFTP, and show the contents of the directory. To copy test.tcl to your home directory, use this command:

```
sftp>get test.tcl
```

You will see a message like the following as the file transfers:

```
Transferring /ap/flash/test.tcl -> /home/chart/test.tcl (1k)
|o.....................................................................................|
|.....................................................................................|
sftp>
```

Binary operating system files and configuration files contain header information that is preserved during file transfers. This information makes it possible to transfer them (via SFTP puts) to the AP. TCL files lack this information, and are treated as flat text files during file transfer operations. If you attempt to use SFTP to put an unexpected file type to flash or NVRAM, the operation may fail because these files lack the header information needed to determine the amount of space the AP needs to allocate for them in storage.

USING THE SECURE COPY PROTOCOL (UNIX SYSTEM)

The SSH secure copy protocol (SCP) uses SFTP as its underlying transport. As a result, SCP operations are controlled by the access levels configured for SFTP users.

This sample command shows how to copy a TCL file from a client's local directory to NVRAM on a target system. If password authentication is enabled, you are prompted for a password as shown:

```
> scp test.tcl chris@198.202.232.159:/ap/nvram/test.tcl
chris@198.202.232.159's password:

Transfering test.tcl -> 198.202.232.159:/ap/nvram/test.tcl  (1k)
|.....................................................................................|
280 bytes transferred in 0.05 seconds [4.94 kb/sec].
```

To copy a file from one storage area to another, use this command. Since the command opens and closes two SCP sessions, you are prompted twice for a password (provided password authentication is enabled).

```
> scp chris@198.202.232.159:/ap/flash/test.tcl chris@198.202.232.159:
  /ap/nvram/test4.tcl
chris@198.202.232.159's password
```

```
chris@198.202.232.159's password:
Transfering 198.202.232.159:/ap/flash/test.tcl -> 198.202.232.159:/ap/nvram/test
4.tcl  (1k)
|*****************************************************************************|
280 bytes transferred in 1.08 seconds [0.25 kb/sec].
```

CONFIGURING IP MULTICAST

IP multicast (multicast) is a routing technique that allows the sending of IP packets from one or more sources to multiple destinations. Using multicast, a single IP packet can be sent to all members of a multicast group. The concept of *group* is fundamental to multicast. In this context, a group is Class D addresses in the range 224.0.0.0 to 239.255.255.255 (as described in RFC 2336) that serves as a destination address for multicast packets.

ABOUT MULTICAST

Multicast uses IGMP (Internet Group Membership Protocol) to dynamically register and remove hosts from multicast groups. Hosts identify themselves as members of a multicast group by sending IGMP messages. Multicast routers listen for these messages, and periodically send out queries to see which groups are active on specific LANs.

Multicast routers can use any of several protocols to communicate with one another and build routes for each group. These protocols include DVMRP.

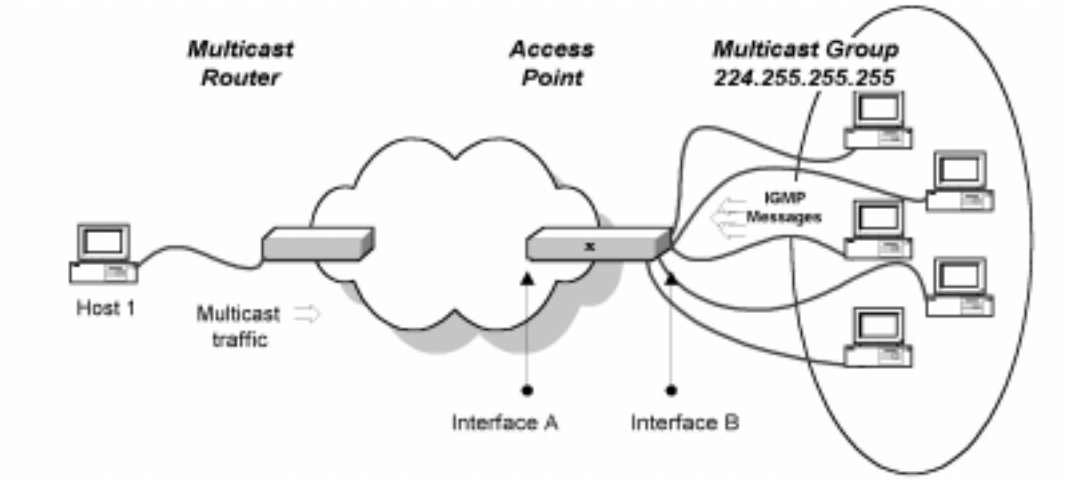
The AP supports multicast traffic over all supported networks, and also over IP tunnels, as discussed on Page 552.

CONFIGURING MULTICAST

When running with multicast enabled, the AP listens on certain interfaces for IGMP messages sent by external hosts. These messages are announcements telling the AP to include or remove a given host from specific multicast group. The AP keeps track of the IGMP requests it receives, and communicates with other multicast-enabled routers to share this information.

To configure multicast, you must enable the protocol globally and configure DVMRP for each IP instance on which you intend to forward multicast traffic. Figure 75 shows a basic multicast configuration. For the AP in this example to forward multicast traffic sent by Host 1 to the multicast group shown, multicast must be enabled on interfaces A and B.

Figure 75 Basic Multicast Configuration



You can enable the protocol globally with this command:

```
> config ip multicast mode enabled
```

Next, use this command to turn on DVMRP and enable multicast on each interface.

```
> config ip.1 multicast protocol dvmrp mode enabled
```

MULTICAST ON MULTINETTED INTERFACES

On each interface, multicast uses a single IP address as the source address for IGMP and DVMRP packets. When multiple addresses exist on an interface, multicast selects the first active address to forward traffic. All other addresses on the interface are ignored. You can explicitly specify the address multicast uses with this command:

```
> config ip.1 multicast address <IP address>
```

If you explicitly configure a multicast address, be sure to configure and enable DVMRP on the interface as explained earlier.

DEFINING MULTICAST SCOPE

You can define the scope of multicasting on an interface. This provides a way of limiting the groups to which multicast traffic will be broadcast. To do this, specify an IP address and a mask. This definition specifies the local scope as specified in RFC 2365:

```
> config ip.1 multicast boundary.239.192.0.0 mask.255.252.0.0
```

LIMITING MULTICAST SCOPE WITH A TTL VALUE

You can limit transmission of multicast packets by configuring a TTL threshold on multicast interfaces. This is especially useful in cases when you do not want packets intended for local recipients to be multicast outside the LAN.

To configure a TTL threshold for an interface, use this command.

```
> config ip.1 multicast ttl 16
```

The default TTL value is 0. The value can be set as high as 255. If you set the value to 16, as in this example, multicast packets with a TTL lower than 16 will not be forwarded out of the interface.

CONFIGURING DVMRP

You may want to configure a default DVMRP route to control announcements to neighbors. In addition, you can also configure a receive policy to define reception of DVMRP routes.

ANNOUNCING A DVMRP DEFAULT ROUTE

You can configure DVMRP to announce a default route to neighbors on a specific interface. This route would take precedence over other routes. To do this, use this command:

```
> config ip.1 dvmrp default <disabled|originate|only>
```

Use a value of disabled if you do not want to announce a default route. A value of originate announces a default route. A value of only only announces the default route.

CONFIGURING A RECEIVE POLICY FOR DVMRP

You can configure a receive policy to define reception of DVMRP routes with this command.

```
> config ip policy-receive.dvmrp name.<name> peer <address>
  address-start <address> address-end <address> receive-action drop
```

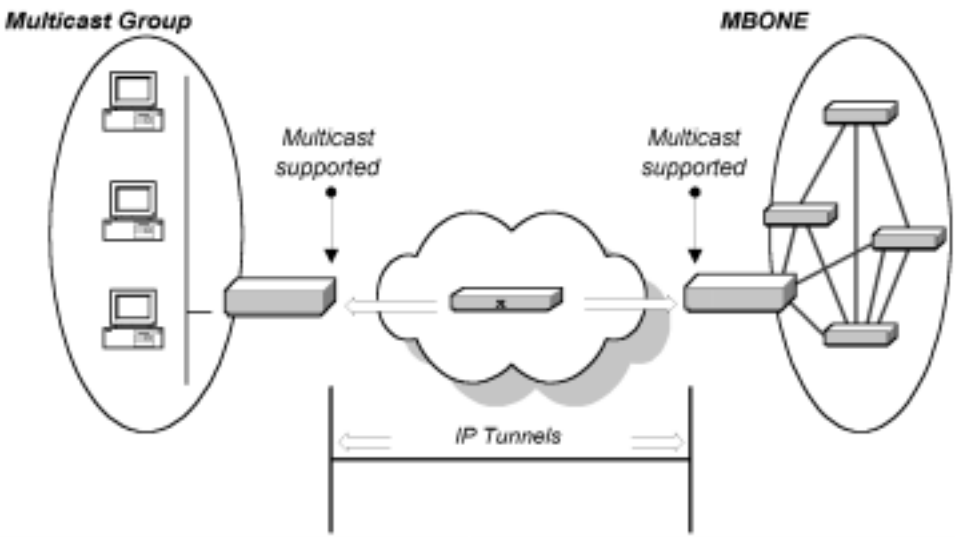
In this context, the peer attribute matches the DVMRP neighbor address.

MULTICAST OVER IP TUNNELS

You can configure the AP to send multicast traffic over an IP tunnel to another multicast-capable router. Doing so enables multicast routers to communicate with one another through the tunnel. In this case, all multicast IP traffic is encapsulated by the AP before delivery to the destination at either end of the tunnel.

Tunneling multicast traffic in this manner enables users connected to a multicast-capable router to communicate with the MBONE, even if intervening routers do not support multicast. Figure 76 shows a general picture of IP tunnel use. In this scenario, members of a multicast group can establish a virtual tunnel connection with a multicast router on the MBONE.

Figure 76 IP Multicast Via IP Tunnel



Multicast is supported over IP tunnel subinterfaces. To create an IP tunnel, begin by adding an IP tunnel and an IP tunnel subinterface, as follows:

```
> stack virtual cbq.5 iptnl.1 iptnl.1.1 ip.6
```

Next, add an unnumbered IP address to the IP forwarder associated with the subinterface. This source address is the local endpoint of the tunnel.

If you want to ensure that this address is up at all times, you can use the loopback address as its source address. You can also use the router address as a source address.

Note that if the router address already exists and you do not specify a different source address, the AP automatically makes the router address the source address of any unnumbered interface added. In this example, 222.44.55.66 is the loopback address.

```
> add ip.6 address.0.0.0.0 src-addr 222.44.55.66
```

```
> add ip.7 address.0.0.0.0 src-addr 222.44.55.66
```

Next, specify the encapsulation method used by the tunnel, and the remote endpoint of the tunnel.

```
> config iptnl.1.1 encapsulation native remote-ip <remote endpoint address>
```

Encapsulation types native and GRE are supported for IP tunnels. Native encapsulation stuffs the packet inside a normal IPv4 header and unicasts it to the remote tunnel endpoint. GRE encapsulation inserts a GRE header between the outer header and the payload header.

INTEROPERATING WITH MROUTED

When sending packets through an IP tunnel, you can configure the AP to unicast DVMRP messages to the remote tunnel end point. This enables the AP to interoperate with older versions of mrouted or with Cisco routers in operating in dvmrp encapsulation mode. To do this, use the following command:

```
> config ip.<instance> dvmrp unicast-tunnel enabled
```

MONITORING MULTICAST

The AP provides monitoring capabilities for IGMP, the multicast forwarding cache and next-hop tables, and DVMRP.

MONITORING IGMP

Issue these commands to get information about IGMP counters:

```
> show ip igmp counters
```

Here is a sample display:

```
                                IP IGMP COUNTERS

          In Msgs  = 2738
        In Errors  = 0
      In Reports   = 730
   In V1 Reports   = 0
       In Leaves   = 1
      In Queries   = 32
    In Unknowns    = 8
       Out Msgs    = 40
```

Use this command to show information on IGMP memberships on interfaces:

```
>show ip igmp cache-table summary
```

Following is a sample display:

```
                          IP IGMP CACHE-TABLE SUMMARY

        ADDRESS|      INTERFACE|                Self|        Last Reporter|
               |               |              Uptime|          Expiry Time|
               |               |              Status|          Version1 Timer|
---------------|---------------|--------------------|----------------------|
       224.0.0.2|           ip.3|               false|       198.202.232.94|
               |               |227041 (0 hours 37 m)|20259 (0 hours 3 min)|
               |               |              active|                     0|
       224.0.1.1|           ip.3|               false|      198.202.232.200|
               |               |227170 (0 hours 37 m)|25730 (0 hours 4 min)|
               |               |              active|                     0|
      224.0.1.24|           ip.3|               false|       198.202.232.64|
               |               |227075 (0 hours 37 m)|20325 (0 hours 3 min)|
               |               |              active|                     0|
      224.0.1.75|           ip.3|               false|      198.202.232.223|
               |               |227081 (0 hours 37 m)|20219 (0 hours 3 min)|
               |               |              active|                     0|
```

Like an IP forwarding table, the multicast forwarding cache contains information that shows how traffic will be forwarded by multicast. This information is periodically and dynamically updated by DVMRP.

MONITORING THE MULTICAST CACHE AND NEXT-HOP TABLES

Use these commands to display information from the multicast cache and next-hop table:

> show ip multicast cache-table summary

> show ip multicast next-hop-table summary

A sample multicast cache table is shown here:

IP MULTICAST CACHE-TABLE SUMMARY

GROUP	SOURCE	MASK	Upstream Interface Uptime Expire Different Protocol
224.0.1.1	198.202.232.200	255.255.255.0	198.202.232.155 ip.3 19033 (0 hours 3 min> 9167 (0 hours 1 min > 0 dvmrp
224.0.1.11	12.0.1.45	255.255.255.0	38.9.74.2 ip.6 214142 (0 hours 35 m> 505858 (1 hours 24 m> 0 dvmrp

MONITORING DVMRP

Use this command to show the global state of DVMRP:

> show ip dvmrp summary

```
                                IP DVMRP SUMMARY

          Version  = Version 3
   Generation Id   = 931953458
Number Of Routes   = 3
Reachable routes   = 3
```

You can show the state of DVMRP on each interface with this command:

> show ip.<instance> dvmrp summary

```
                                IP.6 DVMRP SUMMARY

         Address*  = 198.202.232.155
          Metric*  = 1
           State   = up
         Default*  = disabled
  Unicast Tunnel*  = enabled
Rcv Bad Packets    = 0
 Rcv Bad Routes    = 0
```

Information on the state of DVMRP neighbors is available with this command:

> show ip dvmrp neighbors summary

```
                        IP DVMRP NEIGHBORS SUMMARY

          INTERFACE|         ADDRESS|              Up Time|          Expiry Time|
                   |                 |      Generation Id|       Major Version|
                   |                 |      Minor Version|        Capabilities|
                   |                 |         Rcv Routes|      Rcv Bad Packets|
                   |                 |     Rcv Bad Routes|                State|
-------------------|-----------------|-------------------|--------------------|
               ip.6|       38.9.74.2|3400 (0 hours 0 min >|400 (0 hours 0 min 4>|
                   |                 |          931524343|                   3|
                   |                 |                  8|                0x06|
                   |                 |                  0|                   0|
                   |                 |                  0|             oneway|
              ip.10| 208.218.164.222|1511600 (4 hours 11 >|3000 (0 hours 0 min >|
                   |                 |          910956251|                   3|
                   |                 |                255|                0x06|
                   |                 |              30754|                   0|
                   |                 |                  0|              active|
```

CONFIGURING VIRTUAL LANS

The AP supports virtual LANs (VLANs) based on the 802.1p and 802.1q LAN bridge standards. The 802.1p standard is used to prioritize the queueing of VLAN traffic. The 802.1q standard uses certain criteria to determine VLAN membership.

You can configure the AP to handle VLAN traffic in the following situations:

- Bridged traffic
- Routed traffic
- Prioritized traffic

SETTING UP YOUR VLAN CONFIGURATION

You can configure the AP to perform these tasks when setting up your VLAN configuration:

- Receive VLAN traffic
- Drop VLAN traffic
- Use VLAN IDs in static or dynamic configurations
- Strip off 802.1q tags

RECEIVING VLAN TRAFFIC

Use this command to configure the AP to receive any VLAN traffic:

```
> config eth.<instance> admit-8021q-tagged-frames true
```

DROPPING VLAN TRAFFIC

Use this command to configure the AP to drop any VLAN traffic it receives:

```
> config eth.<instance> admit-8021q-tagged-frames false
```

USING VLAN IDS

When setting up your VLAN configuration, you can specify how the AP uses VLAN IDs in static configurations. You can also specify that the AP advertises its VLAN ID for dynamic configurations.

USING STATIC CONFIGURATION

The AP can assign a VLAN ID to a packet that is being routed onto a LAN based upon the destination IP subnet (static configuration). See "Setting Up Ethernet Subinterfaces" on page 559 for more information about assigning VLAN IDs.

By default, the AP only routes traffic for devices with configured VLAN IDs. To change this parameter so the AP routes traffic for devices with VLAN IDs that have not been explicitly configured on the AP:

```
> config eth.<instance> admit-configured-vlans-only false
```

USING DYNAMIC CONFIGURATION

The AP supports end-station GARP VLAN Registration Protocol (GVRP) so that connected switches running GVRP do not have to be manually configured with a VLAN ID. Although the AP can route traffic, the AP will not be able to learn information about the VLANs with which connected switches have been configured.

By default, GVRP for dynamic configuration between VLAN devices is enabled. GVRP can be enabled and disabled for each Ethernet integral interface. To disable GVRP:

```
> config eth.<instance> enable-gvrp false
```

STRIPPING OFF 802.1Q TAGS

You can configure the AP to strip off the 802.1q tag when bridging VLAN traffic. The AP only bridges VLAN traffic over Ethernet integral interfaces.

To configure the AP to strip off the 802.1q tag before transmitting any bridged VLAN traffic:

```
> config eth.<instance> strip-bridged-8021q-tags true
```

Note that when you configure the AP to strip off 802.1q tags, *only* 802.1q tags are stripped off before transmission. Any 802.1p tags configured by the AP (see "Prioritizing VLAN Traffic" on page 562) are not stripped off even though they use the same header as the 802.1q tags.

DISPLAYING THE VLAN CONFIGURATION

To display the parameters set for the routing and bridging of VLAN traffic, use this command:

```
> show eth.<instance> vlan
```

This sample command displays the VLAN configuration for the eth.1 instance:

```
> show eth.1 vlan
```

```
  Admit 8021q Tagged Frames* = true
Admit Configured VLANs Only* = true
   Strip Bridged 8021q Tags* = false
               Enable GVRP* = true
            GARP Join Time* = 20
```

SETTING UP ETHERNET SUBINTERFACES

Access Point systems support the creation of subinterfaces for Ethernet data links. While integral interfaces have one index, subinterfaces have two indices. For instance, eth.1 is an integral interface, but eth.1.1 and eth.1.2 are subinterfaces. Each subinterface uses a separate IP instance. The IP instance can be multinetted.

Ethernet subinterfaces classify VLAN traffic based upon the VLAN ID. Once the VLAN ID has been assigned, VLAN traffic is handled in the following manner:

- Incoming IP traffic that is contained in 802.1q tagged packets is forwarded to the correct Ethernet subinterface based on the VLAN ID.
- Incoming IP traffic with a VLAN ID that has not been configured on the AP is forwarded over the Ethernet integral interface, if it exists, or dropped. You can also specify that routed traffic from an unknown VLAN ID be dropped (see “Using Static Configuration” on page 558).
- Outgoing IP traffic routed over an Ethernet subinterface is tagged with the VLAN ID before transmission.
- Outgoing IP traffic routed over the Ethernet integral interface is not tagged. If a tag exists, it is stripped off. Note that 802.1p tags configured by the AP (see “Prioritizing VLAN Traffic” on page 562) are not stripped off.

To set up Ethernet subinterfaces, perform these tasks:

1. Create the Ethernet subinterface.
2. Assign a VLAN ID.

Creating Ethernet Subinterfaces

To create an Ethernet subinterface, use the *stack* command to place it over the data link. For example, this command creates an Ethernet subinterface and stacks it above the integral interface (eth.2):

```
> stack slot.4.1 cbq.3 eth.2 eth.2.new
```

As mentioned earlier, this subinterface requires an additional IP forwarder, which can be either numbered or unnumbered. To add the subinterface and forwarder at the same time, proceed as follows:

```
> stack slot.4.1 cbq.3 eth.2 eth.2.new ip.new
```

If you want to add 802.1p bits to prioritize traffic (see "Prioritizing VLAN Traffic" on page 562), you can add the CBQ instance for the sub interface at creation time as follows:

```
> stack slot.4.1 eth.2 eth.2.new cbq.new ip.new
```

The keyword "new" causes the AP to supply the next available instance of the object added with the *stack* command.

To display the Ethernet subinterfaces and CBQ instances above the Ethernet integral interface, use the *status* command.

This partial display shows the results of entering the preceding *stack* commands one right after the other:

SYSTEM STACK

ROW		Layer	Status
1	slot.1.1 Ethernet Driver	enabled	up
2	slot.1.1 cbq.1	enabled	up
3	slot.1.1 eth.1	enabled	up
4	slot.1.1 ip.1	enabled	up
5	slot.3.1 LMC5200 HSSI Driver	enabled	up
6	slot.3.1 cbq.2	enabled	up
7	slot.3.1 hssi.1	enabled	up
8	slot.3.1 frame-relay.1	enabled	up
9	slot.3.1 bridge.1	enabled	up
10	slot.3.1 ip.2	enabled	up
11	slot.4.1 Ethernet Driver	enabled	down
12	slot.4.1 cbq.3	enabled	down
13	slot.4.1 eth.2	enabled	down
14	slot.4.1 ip.3	enabled	down
15	slot.4.1 eth.2.1	enabled	down
16	slot.4.1 eth.2.2	enabled	down
17	slot.4.1 ip.9	enabled	down
18	slot.4.1 eth.2.3	enabled	down
19	slot.4.1 cbq.6	enabled	down
20	slot.4.1 ip.10	enabled	down

ASSIGNING VLAN IDS TO SUBINTERFACES

Ethernet subinterfaces do not come up until they have been configured with a valid VLAN ID. To add a VLAN ID to a subinterface:

```
> config eth.<instance>.<instance> vlan-id <number>
```

where <instance> specifies the subinterface and <number> indicates the 12-bit VLAN ID used in the VLAN tag header as a value from 0 - 4094 (0 is the null VLAN ID). For example:

```
> config eth.2.1 vlan-id 12
```

To display Ethernet subinterface summary information:

```
> show eth.2.1 summary
```

ETH.2.1 SUMMARY

```
      Description  = Ethernet Datalink slot: 4 port: 1
             Type  = iso88023Csmacd
            Speed  = 100000000 bits per second
      MAC Address  = 00:00:00:00:00:00
             Mode* = up
            State  = down
Last State Change  = 42603155 (118 hours 20 min 31 secs)
          VLAN ID* = 12
```

PRIORITIZING VLAN TRAFFIC

To tag packets with 802.1p priority bits using CBQ, use the *stack* command to add an instance of CBQ between each Ethernet interface or subinterface that is expected to transmit 802.1p tagged packets and the bridge or IP forwarder. For example, use the following command to add an Ethernet subinterface, the CBQ instance, and the IP forwarder all at once.

```
> stack slot.<instance>.<instance> eth.<instance> eth.<instance>.new
  cbq.new ip.new
```

The keyword “new” causes the AP to supply the next available instance of the object added with the *stack* command.

Once a traffic class definition has been created to classify traffic based upon existing criteria, you can set the *datalink-traffic-index* parameter to add 802.1p priority bits to the classified traffic.

To configure the *datalink-traffic-index* parameter for cbq.3 and its traffic class called asap:

```
> config cbq.3 traffic-class.asap datalink-traffic-index <number>
```

where <number> indicates priority as a value from 0 - 7 (7 being the highest priority).

To display the CBQ datalink parameter configuration:

```
> show cbq.3 traffic-class.foo datalink
```

```
                        CBQ.3 TRAFFIC-CLASS.FOO DATALINK

Datalink Traffic Index= - 4
Datalink Traffic Flags= = 0x00
```

NOTE *When you add an 802.1p tag and configure the AP to strip off 802.1q tags for bridged traffic, the 802.1p tag that has been configured by the AP is not stripped off even though it uses the same header as the 802.1q tag.*

CONFIGURING THE AP STACK WITH SNMP

You can use raw SNMP sets to add new entries to the AP Interfaces table, and connect and disconnect entries from the stack.

All objects used in the explanations that follow are documented in the Xedia MIB, available at *ftp://ftp.xedia.com/MIBS*. Unless it is necessary to use an SNMP agent, you should use the CLI or Access View to configure the AP.

CREATING A NEW ENTRY IN THE INTERFACES TABLE

Every entry in the interfaces table has an ifIndex value. You need to assign an ifIndex value when creating a new entry. While you can assign any unused ifIndex value to a new entry, it is recommended that you assign the next available value as explained below.

ASSIGNING THE NEXT IFINDEX VALUE TO A NEW ENTRY IN THE IFSTACKTABLE

Let's say you want to create a new instance of Frame Relay or CBQ. While you can supply any ifIndex value, it is preferable to discover and use the next value of ifIndex.

Use this command to discover the value of xifNextIndex. Use of the .0 is specific to this object and describes how it is instantiated.

```
get xifNextIndex.0
  value = 32
```

Next, set the value of xifNextIndex to 32. Doing this reserves the next value of 32 for your use.

```
set xifNextIndex.0 32
```

Provided that this set produces a success message, you can assign an ifName and rowStatus to the ifIndex. This set command assigns the ifName and instance number CBQ.5 to row 32, with a rowStatus of createAndGo.

```
set ifName.32 cbq.5 xifRowStatus.32 createAndGo(4)
```

See the following section (or the Xedia MIB) for valid ifName formats. The

row assignments created with this set can be represented as follows:

	ifIndex	*ifName*	*rowStatus*
Row	*32*	*cbq.5*	*createAndGo*

THE FORMAT OF IFNAME VALUES

In the preceding example, the ifName assigned to ifIndex 32 is CBQ.5. Whether you choose to provide a instance-number value at random or use the next available value, the value you provide for ifName must conform to the following general form:

```
<component-name>.<instance-number>.[<sub-interface-number>]
```

As when using the *stack* command at the CLI, you can use the keyword *new* in place of either <instance-number> or <sub-interface-number>. Following are five sample ifName objects:

- cbq.new
- frame-relay.1
- frame-relay.1.new
- frame-relay.1.5
- slot.1.1

CONNECTING AND DISCONNECTING IFENTRIES

To connect and disconnect existing ifEntries, use the ifStackStatus object to modify the ifStackTable. Each entry in this table specifies a connection between two ifEntries. You can use the xsysIfNameIndexTable to get the ifIndex values. Just enter the index of ifName to get the associated ifIndex value. For example, you can get the ifIndex value for eth.1 and cbq.5 as follows:

```
get xsysIfIndex.eth.1
  value = 20

get xsysIfIndex.cbq.5
  value = 32
```

Once you know the ifIndex values, you can issue a *set* command to connect

each entry in the ifStack table. This *set* command connects ip.1 to eth.1. Notice that the ifIndex value for ip.1 precedes the value of eth.1. When working with MIB objects, you must specify values from the highest level (in this case, IP) to the lowest (Ethernet).

In this case, note that both createAndGo and 4 (the enumerated value associated with createAndGo) are specified. Whether you use the createAndGo or its value depends on the SNMP manager you are using.

```
set ifStackStatus.20.32 createAndGo (4)
```

To disconnect these two ifName entries, use a *set* command and a rowStatus of Destroy as shown here. Again, you must use either destroy or 6, depending on your SNMP manager.

```
set ifStackStatus.24.20 destroy (6)
```

NOTES ON SNMP CONFIGURATION

Following are two rules of thumb to follow when configuring the AP stack.

First, remember to verify the changes you make to the ifName table. For instance, if you do a *set ifname.<x> xifrowstatus.<index> 4*, you should do a *get ifname.<x>* to make sure it was really created.

You should also verify changes to the ifStack table. For instance, if you do a *setifstackstatus.<upper>.<lower> 4* (OR *6*), follow this command with a *getifstackstatus.<upper>.<lower>* to make sure the connection was created (or destroyed) as requested. This step is important because if the connection does not exist, the result contains Empty instead of the instance information.

INDEX

A

Access Point
- about 1
- architecture 2
- as access router 4
- as edge device 4, 16, 17
- attribute information, showing 29
- bandwidth management capabilities 1
- command line interface, managing with 22
- date and time, changing 46
- interface architecture 3, 27
- interface group information, showing 29
- interface objects, obtaining information about 27
- interfaces table, configuring with SNMP 563
- locating in your network 15
- logging in 18
- management 7
- managing the system 37
- managing with Access View 457
- modes of deployment 15
- network configurations 1
- POP deployment 17
- products 1
- resetting the software 47
- RFC support 8
- settable attributes 31
- SNMP support 38
- stack command, making dynamic changes to
  - configuration with 33
- status, checking 19
- system name, changing 46
- watchdog timer, configuring 46
- Web-hosting deployment 16

Access View
- about 7
- assigning a system name 458
- bandwidth
  - allocation, displaying 469, 472
  - utilization, displaying 473
  - utilization,displaying 469
- browser requirements 7
- CBQ traffic classes, configuring 465
- CBQ tree structure, displaying 471
- Class-Based Queuing, configuring 463
- DNS client, configuring 460
- Frame Relay, configuring 462
- frames support 7
- getting help 456
- HSSI, configuring 461
- HTML support 7
- Java support 7
- logging in 454
- managing the Access Point system 457
- overview 453
- setting data and time 458
- slot configuration information

changing 456
displaying 456
SNMP communities, setting up 459
starting up 454
system configuration information
changing 456
displaying 456
system messages, displaying 457
Traffic Class Graphing wizard, using 482
traffic classes, adding 465
Traffic Status application 469
accessing 469
bandwidth utilization, displaying 469
description of frames 470
display preferences, setting 479
help information, displaying 477
VPN configuration information, changing 456
window description 455
Current Page frame 456
Information Log frame 457
Management Navigator frame 456
Administrative community, creating 42
Architectural overview 2
ARP table size
configuring 99, 138, 178
ATM
adaptation layer 118
ARP table size
configuring 138
ATM layer 120
ATM-DS3 layer 120
ATM-SONET layer 120
CBQ layer 120
CBQ, configuring to select a virtual connection 141
cell structure 118
CLP bit, setting 142
configuring bridging over 264
datalink-traffic-flag parameter, setting 143
DS3 layer
attributes 123
circuit ID 124
line coding 124
line length 124
line type 124
loopback config 124
send code 124
transmit clock source 124
configuring 123
current interval, showing 125
line length, setting 124
prior intervals, showing 125
status, showing 125
attributes
line 125
loopback 125
valid and invalid intervals 125
implementation in Access Point systems 120
interface, configuring 136
cell delay variation tolerance 135
cell-align 136
empty-cell 136
IPOA layer 121
adding a CBQ instance over 141
overview 117
Quality of Service 120
categories 131
CBR 131
UBR 131
VBR 131
setting up interfaces 122
about traffic descriptors and service categories 130
CBQ, configuring to select a virtual connection 141
CLP bit, setting 142
data link layer, configuring 128
datalink-traffic-flag parameter, setting 143
IP address
adding to each ATM interface 139
IPOA layer
adding a CBQ instance over 141
configuring 137
mapping traffic classes to virtual connections 141
physical interface, configuring 123
predefined descriptors, using 132
service categories 133
CBR 134

UBR 134
VBR.1 133
VBR.2 133
VBR.3 134
subinterfaces, configuring 139
traffic descriptors
adding 130
defining 132
VCLs
adding 129
adding at IPOA layer 138
configuring 129
SONET interface
attributes 126
circuit ID 127
loopback config 127
medium type 127
transmit clock source 127
configuring 126
current interval, showing 128
physical layer information 126
prior intervals, showing 128
status, showing 127
loopback 127
section, line, or path 127
valid and invalid intervals 127
subinterfaces
adding circuits to 140
configuring 139
setting up 139
switching 119
virtual connection mapping 141
Authentication
by a remote router 173
configuring
PPP 172
RADIUS 507
SSH server 517
configuring PPP for 173
AutoClasses
creating 6
enabling 325
limiting classes created on an interface 319
naming conventions 329
naming restrictions 330
invalid name prefixes 331
parameters
auto-class-child-dest-ip-divisor 326
auto-class-child-max-bw-alloc 325
auto-class-child-src-ip-divisor 326
auto-class-sub-tree-depth 327
removing at specified intervals 317
sample configuration 325
saving AutoClass children 333
subtree class definitions, editing 333
subtree depth, specifying 327
using 324

B

Bandwidth management 14
capabilities 1
CBQ and link sharing 3
overview 3
BGP
activating 219
AS number, configuring 219
AS path regular expressions 226
classifying traffic on AS Path 295
regular expressions, using 295
classifying traffic on BGP Community 293
configuring 218
deactivating 219
decision process 223
default route policy 222
for redundancy and load balancing 245
in complex networks 247
overview 218
peers
displaying information about 219
removing 220
peers, adding 219
receive policy actions 224
accept routes 224
drop (filter) routes 224
filter by AS path 224
filter by community 224
set local preference 224
route information, displaying 221
route policy 222

route reflector server 221
configuring
clusters 222
servers 221
client and non-client 221
full configuration 221
route selection 223
send policy actions 225
aggregate routes 225
import routes 225
never send routes 225
prepend AS 225
send routes 225
set community 225
set Multi-Exit Discriminator (MED) 225
set next-hop-self 225
static routes 244
Boot order, configuring 55
Bridge encapsulation, routing over 266
Bridging
and routing 259
configuring redundancy 267
configuring VBRP 267
converting a bridge to a router 262
converting a router to a bridge 260
creating a bridge-router configuration 261
non-IP traffic 261
over ATM 264
over Frame Relay 263
over PPP 264

C

CBQ
See also Class-Based Queuing
Certificate Revocation Lists (CRL)
configuring 449
disabling CRLs 451
loading CRLs 450
system time, configuring 450
viewing certificates 451
viewing CRLs 451
Class-Based Queuing
as a bandwidth manager 272
as a firewall 273
AutoClasses
enabling 325
naming conventions 329
naming restrictions 330
invalid name prefixes 331
parameters
auto-class-child-dest-ip-divisor 326
auto-class-child-max-bw-alloc 325
auto-class-child-src-ip-divisor 326
auto-class-sub-tree-depth 327
sample configuration 325
saving AutoClass children 333
subtree class definitions, editing 333
using 324
classification 14
classifying traffic
by domain name 289
by forwarding profile 249
CBQ class, defining 252
on applications 285
stateful pairs 285
on AS Path 295
regular expressions, using 295
on BGP Community 293
on datalink traffic 298
802.1p 299
802.1q 300
on host range 284
on port range 287
on protocol range 286
on ToS values 335
classification-tos 335
classification-tos-mask 335
operators 288
configuring with Access View 463
connection accounting, configuring 365
description of 14
directionality 4
firewalls 340
flexibility of 5
general purpose 3
H.323 traffic, classifying 316
hierarchical classification 273
hierarchy

AutoClasses classes 283
components 281
default classes 282
null classes 283
root classes 281
traffic classes 282
how it works 273
IP fragments,classifying 372
managing 317
AutoClasses, removing at specified intervals 317
displaying statistics 321
aggregates 321
borrow attempts 323
borrowed bytes 323
bytes, aggregate bytes 321
cbq-stats 321
dropped bytes 321
dropped packets 322
flow 321
for all classes 321
for selected classes 321
high water 322
idle 322
over limits 322
packets, aggregate packets 322
q-elasticity factor 290
queued packets 322
throttles 323
limiting classes created on an interface 319
statistics collection
controlling 320
controlling rate of 320
for all classes 319
for selected classes 320
monitoring stateful connections 361
multimedia traffic, classifying 315
overview 272
packet matching 4
parameters 4
parameters for writing to the ToS field of packets
tos-borrow-mask 337
tos-borrow-value 337
tos-mask 336
tos-value 336
RED
configuring 291
monitoring 293
parameters
q-elasticity-factor 291
red-average-queue-size 293
red-dropped-packets 293
red-minimum-queue-value 291
red-probability 293
RTSP traffic, classifying 316
scheduling 14
stateful classification 305
about 306
advantages of 307
configuring 308
FTP, configuring 313
positioning classes to protect end systems 310
with firewalls 311
without firewalls 312
supported applications 308
unsupported applications, classifying 309
with TCP 314
with UDP 314
stateless classification 305
how it works 306
SYN flood protection 377
ToS value on packets, setting
that borrow bandwidth 338
that do not borrow bandwidth 337
traffic classes
bandwidth threshold 290
configuring 277
managing queues 291
peer classification order 301
enhancing security 302
filtering specific traffic types 301
sample class definition 277
sample configuration 278
traffic parameters, using in combination 300
traffic classification 4
about 273
CBQ trees 273
class relationships 274
on input or output 276
traffic parameters 283

Classification
and queuing 4
and scheduling 14
Client tunnels
authentication
by RADIUS 440
locally 440
authentication, disabling 441
configuring an authentication method 405
digital certificates, assigning 405
preshared key, assigning 405
router address
designating 393, 404
security profile parameters 398
security profiles, configuring 394, 405
user authentication 409, 439
user password authentication 440
user password authentication, enabling 95
Command Line Interface
about 7
command completion 24
command recall 25
commands 22
add 22
colwidth 22
config 22
describe 22, 26
disable 22
dump-config 56
enable 22
gen-script 85
help 22, 26
log 22, 59
more 25
ping 20, 23
remove 23
reset 23
show 23, 29
stack 23, 33
status 19, 23, 27, 57
sync 23, 86
traceroute 21, 23
where 32
help, getting 26
screen display area, changing 25
settable attributes 31
Configuration
saving 54
script commands, generating 85
summary 56
changes, displaying 56
status, displaying 57
connections from a remote router 172
CRLs
configuring 449

D

datalink traffic
classifying traffic on 298
802.1p 299
802.1q 300
Date and time
setting 46
setting with Access View 458
Default classes
preventing dropped packets 6
Default route
adding 191
configuring 191
Demilitarized zone
defined 354
describe command 26
DHCP Relay
about 499
configuring 499
destinations, displaying 502
parameters 500
relaying requests by interface to a particular DHCP server 501
server destinations, configuring 501
Digital certificates
certificate request, configuring 447
configuring 446
generating and loading certificate 448
generating PKI key pairs 447
DMZ
define 354

DNS client
configuring 62
configuring with Access View 460
DNS parameter settings, displaying 64
domain name, configuring 62
enabling and disabling the client 63
managing the cache 63
resetting the client 63
Dropped packets, preventing 6
DS3 layer
ATM, configuring for 123
attributes 123
circuit ID 124
line coding 124
line length 124
line type 124
loopback config 124
send code 124
transmit clock source 124
current interval, showing 125
line length, setting 124
prior intervals, showing 125
status, showing 125
attributes
line 125
loopback 125
valid and invalid intervals 125
DVMRP
announcing a default route 551
configuring 551
displaying information about 556
receive policy, configuring 552

E

E1
about 103
interface, configuring 103
setting up the interface 108
clock source 108
data link facilities 108
examining E1 defaults 109
for Fractional E1 112
interface type 109
line coding 108
line type 108
loopback mode 110
forcing local into 110
local analog loopback 110
local framer loopback 110
remote line loopback 110
remote payload loopback 110
echo interval
configuring for PPP 169
Equal-Cost multipath load balancing
RIP 193
Ethernet
ARP table size
configuring 99
autonegotiation 100
duplex, changing 98
flow control 100
Gigabit Ethernet, configuring 100
interface, configuring 98
IP address, adding 99
parameters 98
speed, changing 98
Event management 73
console display, controlling 82
displayable subsystems 77
event filtering configuration, sample 84
events
sending as SNMP traps 82
sending to syslog servers 82
filtering options
displaying 76
setting
by date and time 75
by input event type 75
by level 74
by output event type 75
by subsystem 76
subsystem, displaying by 77
maximum non-persistent event log size, setting 81
maximum persistent event log size, setting 81
parameters
configuring 81
displaying 83

System Event Log table
clearing 81
displaying 73

F

File transfer
interpreting errors 53
ordinary transfer 53
secure transfer 52
File transfer table 47
about 47
creating table entries 49
parameters 50
action 51
dest 50
dest-file-name 50
encryption-password 50
server address 50
source 50
source-file-name 50
displaying 51
security 49
Firewalls 340
connection accounting, configuring 365
DMZ
defined 354
ICMP packets
classifying 366
controlling types sent and received 367
filtering example 368
statefully classifying 366
IP fragments
classification parameters 376
classifying 372, 375
denial of service attacks, foiling 373
displaying information about 374
dropping all fragments 375
forwarding 375
how classifying fragments works 372
limiting fragment handling 374
monitoring stateful connections 361
connection state information
about 361
sorting by time 363
where command 363
summary information 365
viewing by flow 362
spoofing 350
stateful classification 340
and bandwidth management 342
how it works 340
LAN to LAN configuration, example 345
LAN to WAN configuration, example 348
securing the DMZ configuration, example 353
using established classes 341
using initiated classes 341
SYN flood protection 377
active mode 377
passive mode 378
Forwarding policy
configuring 249
Forwarding profile
classifying traffic on 249
Fractional T1/E1, configuring 112
Frame Relay
Access Point systems, configuring for 163
Access View, configuring with 462
bridging, configuring over 263
CBQ
adding an instance over the Frame Relay layer 167
configuring to select a DLCI 167
DLCIs, configuring 167
interface, setting up 164
mapping traffic classes to DLCIs 167
PPP connections 164, 168
PVCs, setting up 164
checking configured PVCs 165
validating Frame Relay connectivity 165
replacing with PPP connections 168
subinterfaces, setting up 165
assigning DLCIs to subinterfaces 166
configuring Frame Relay subinterfaces 166

G

Gigabit Ethernet

autonegotiation 100
configuring 100
flow control 100
interface, configuring 98
parameters 98

H

H.323
classifying multimedia traffic 316
help command 26
HSSI
configuring with Access View 461
interface
configuring 101
setting up 101

I

IGMP, displaying information about 554
IKE Keepalive protocol 414
configuring 415
how it works 414
parameters 416
Integrated Services Digital Network
See also ISDN
Interface architecture 27
Intermediate System
See also IS
Internet Key Exchange (IKE)
about 387
IP
address
adding 184
adding to an interface 178
displaying 181
configuring 178
ip.sink address, using 181
loopback interface address
source address 180
using 180
memory allocation
changing 189
configuring 189
multicast, configuring 549
multinetting 184
multiple IP addresses to an interface, adding 184
proxy ARP 185
router address as default source address 184
routes to a specific destination, displaying list of 188
unnumbered interface
adding 179
deleting 180
using 179
unnumbered IP addresses, using 184
verifying the source of a packet 185
IP address
adding for Ethernet 99
WAN connections, adding 19
IPSec
about 386
Authentication Header (AH) 386
client tunnel
authentication 391
establishing a session 390
how it works 389
overview 388
Encapsulating Security Payload (ESP) 386
Internet Key Exchange (IKE) 386
site-to-site tunnel overview 387
Transport Mode connections, configuring 417
Transport Mode overview 392
ISDN
configuring 144
dial services, configuring 151
dial back service 155
dial backup service 155
dial on demand service 152
permanently connected service 151
interface, configuring 144
integral interface 144
local interface 144
remote interface 147
subinterface 147
ISDN configuration, displaying 147
local interface, configuring
directory number 145

displaying the configuration 147
Service Profile Identifier 146
SPID 146
subaddress 145
switch type 146
remote interface, configuring
call state 151
current call information, displaying 151
dial services 151
IP forwarder 148
peers 149
parameters
answer-dn 149
control 149
cug 149
dial-backup 149
failure-delay 149
idle-timer 149
info 149
max-duration 150
min-duration 150
originate-dn 150
peer-subaddress 150
permission 150
retries 150
retry-delay 150
speed 150
subinterface configuration, configuring 150
subinterfaces 147
S/T interface module
verifying proper operation 157
subinterface configuration, displaying 150
U interface module
configuring 144
directory numbers 145
local interface, configuring 144
SPID 146
switch type 146
verifying proper operation 158
version, displaying 147
ISIS
addresses, summarizing 214
configuring 211, 213
addresses 214
adjacencies 215
area address 213
enabling 214
IP interface 215
routing level 214
system ID 213
displaying
adjacencies 215
link state database 216
level 1 routing 211
level 2 routing 211
sample configuration 216

L

L2TP
about 424
configuring 431
client users 431
IP interface 431
IP LAN interface 431
L2TP tunnel domains 431
tunnel domains 432
tunnels 424
verifying layers are enabled 431
how it works 426
tunnel information, displaying 436
Virtual Tunnel Transport stack 384, 427
Layer components
SNMP agent
adding with 563
removing with 563
stack command
adding with 33
modifying with 36
removing with 33

M

magic number negotiation
configuring for PPP 169
Managing Access Point systems 7, 37
MIB object details, showing 26
more command 25
MSSI

interface
configuring 101
setting up 102
Multicast
about 549
configuring 549
defining scope 551
displaying
IGMP information 554
information about 554
DVMRP, configuring 551
interoperating with Mrouted 553
IP tunnels, configuring over 552
limiting scope with TTL value 551
monitoring multicast cache 555
monitoring next-hop tables 555
on multinetted interfaces 550
Virtual Tunnel Transport stack 384
Multicast cache, displaying information 555

N

Network Address Translation (NAT)
configuring 487
dynamic bindings 491
LSNAT pools 488
maximum number of sessions 492
NAPT pools 488
private networks 489
static bindings 491
disabling 493
displaying
bindings 492
configured private networks 490
LSNAT pools 490
NAPT pools 490
NAT pools 490
translation sessions 492
enabling 493
instances, adding 489
overview 486
sample configurations 494
session timers
displaying 493
modifying 493
using 493
Network Time Protocol (NTP)
adding servers 505
calculating correct time 503
configuring 503
enabling 504
overview 503
Next-Hop tables
displaying information 555

O

OSPF
area, assigning an interface to 201
authentication 210
MD5 210
simple password 210
components 196
area 196
area border router 196
AS boundary router 196
backbone 196
configuring 196
area aggregates 208
areas 200
interface types 202
interfaces 199
NBMA neighbors 203
normal areas 207
not-so-stubby areas 207
passive interfaces 201
point-to-multipoint neighbors 205
stub area metrics 208
stub areas 207
virtual links 206
connectivity problems, debugging 209
default types 202
defaults, displaying 199
enabling 201
import-as-extern parameter, using 207
interface cost, setting 209
interface types, setting 202
interfaces, reassigning to different areas 206

NBMA configuration 203
normal areas 207
not-so-stubby-areas 207
planning 198
stub areas 207
summary LSAs, blocking import of 208

P

Packet over SONET
interface, configuring 159, 161
attributes 159
circuit ID 160
clock source 160
loopback config 160
medium type 160
physical layer, configuring 159
SONET path labels 162
ping command
description 20
options 20
delay 20
IP address of source host 20
length of packets 20
number of packets sent 20
range of request packet lengths 20
send one request per second 20
timeout 20
PKI certificates
configuring 446
Access Point's subject distinguished name 446
certificate request 447
generating and loading certificate 448
generating PKI key pairs 447
Planning your routing setup 12
POP deployment 17
POS
See also Packet over SONET
PPP
authentication, configuring 172
authentication by a remote router 173
connections from a remote router 172
configuring bridging over 264
echo interval, configuring 169
interface, configuring 164, 168
magic number negotiation, configuring 169
managing bandwidth on a PPP MP link 172
Multilink, configuring 170
PPP MP instance, adding 171
replacing Frame Relay with 168
validating connectivity 170
PPP MP
configuring 170
instance, adding 171
managing bandwidth on a PPP MP link 172
replacing Frame Relay with PPP 170
Proxy ARP 185

R

RADIUS authentication
configuring 507
RADIUS client 511
RADIUS server 508
server entries for the client parameters 513
user authentication method 512
displaying
accounting log information 515
statistical information for the servers 516
summary information for the client 516
user information 515
enabling 512
parameters 513
Random Early Detection
See also RED
RED
configuring 291
monitoring 293
parameters
q-elasticity-factor 291
red-average-queue-size 293
red-dropped-packets 293
red-minimum-queue-value 291
red-probability 293
Regular expressions 226, 295
Resetting the software
from the CLI 47

to the factory default configuration 47
to use current configuration 47
Reverse path verification 185
RFCs and working drafts 8
RIP
activating 193
changing implicit policies 193
action type
receive 194
send 194
configuring 192
deactivating 193
displaying information 194
equal-cost multipath load balancing 193
explicit policies, configuring 195, 233
default behavior 195
Route policy 227
actions 232
receive-action 232
send-action 232
administrative distance 239
BGP-specific capabilities 234
aggregate routes 234
AS prepend 235
AutoMed 234
community 235
local preference 234
MED 234
next-hop-self 235
parameters 234
description 227
execution order 237
individual routing protocols, controlling 227
ISIS-specific capabilities 233
OSPF-specific capabilities 233
overview 227
peer groups 230
adding 230
displaying information about 231
removing 230
sharing policy definitions with 230
peer matching 229
policy execution, controlling with policy names 236
policy matching 228
prefix matching criteria 240
receive policies 228
RIP-specific capabilities 233
routing protocol interaction, controlling 227
sample configurations
BGP multihomed network 245, 247
AS prepend, using 248
community attribute, using 248
local preference attribute, using 246
MED attribute, using 246
setting up IBGP peers 248
BGP stub network 244
RIP POP network 241
receive policies 243
send policies 242
send policies 228
Route selection 223
Router
converting bridge to router 262
converting to bridge 260
Routing
BGP memory allocation, changing 190
BGP, configuring 218
considerations 12
creating a bridge-router configuration 261
default network, configuring 188
default route
adding 191
configuring 191
explicit RIP policies, configuring 233
forwarding directed broadcasts 187
ICMP redirect, using 187
IP
proxy ARP 185
verifying the source of a packet 185
IP address, adding 184
IP memory allocation
changing 189
configuring 189
IP multinetting 184
IP parameters (general), displaying 187
ISIS, configuring 211
NAT, configuring 487

OSPF planning 198
OSPF, configuring 196
over bridged encapsulation 266
peer groups 230
adding 230
displaying information about 231
removing 230
sharing policy definitions 230
policy matching 228
receive policies 228
redundancy, configuring 253
RIP, configuring 192
route allocation, changing 189
route policy 227
BGP-specific capabilities 234
description 227
individual routing protocols, controlling 227
ISIS-specific capabilities 233
OSPF-specific capabilities 233
peer matching 229
RIP-specific capabilities 233
routing protocol interaction, controlling 227
route policy actions 232
receive-action 232
send-action 232
route policy overview 227
router address as default source address 184
send policies 228
setup, planning 12
static route
adding 191
configuring 191
displaying 192
supported IP protocols 190
unnumbered IP addresses 184
VRRP, configuring 253
RTSP
classifying multimedia traffic 316

S

Saving the configuration 54
Scheduler
about passes 69
batch job parameters 67
batch jobs 68
command parameters 66
environment 66
function 66
parameters 66
log messages, clearing 66
logging, enabling or disabling 66
Scheduler, turning on or off 66
sleep interval, setting 66
status, displaying 68
system time, effects of changing 69
timing jobs 69
using 66
Searching by attributes, using where command 32
service categories
CBR 134
UBR 134
VBR.1 133
VBR.2 133
VBR.3 134
show command 29
Site-to-site tunnels
configuring 399, 401
dynamically 402
security profiles 394, 399
static security associations 403
statically 402
wildcard tunnels 403
IKE Keepalive protocol 414
preshared keys, assigning 400
remote gateway 400
router address
designating 393, 399
security profile parameters 396
subinterfaces 399
SMDS
ARP table size
configuring 178
configuration, displaying 177
configuring 175
ARP request address 177
CRC 177
group address 176, 178

individual address 176
subnet 176
interface, adding 175
SMDS and MAC address mappings, displaying 177
SNMP communities
about 38
adding 40
administrative community, designating 42
community names, setting up 38
community table 38
default community users 38
managing 38
modifying 40, 41
removing 41
setting up with Access View 459
SNMP trap destinations, configuring 45
SNMP, configuring the AP stack with 563
SNMPv3
authentication and privacy applications, specifying 44
authentication and privacy keys, generating 44
CLI, restricting user access to 44
configuring a user 43
engine ID, configuring 43
user access
restricting 44
SONET interface
attributes
circuit ID 127
loopback config 127
medium type 127
transmit clock source 127
attributes and values 126
configuring 126
current interval, showing 128
physical layer information 126
prior intervals, showing 128
status, showing 127
loopback 127
section, line, or path 127
valid and invalid intervals 127
SSH
client
command line options 540
configuration settings 539
configuring 534
host keys 537
host authentication mode 535
key data, adding 538
public key authentication 537
sample session 538
scripts for configuration, using 542
TCL environment variables 542
TCP session retries, controlling 536
user authentication 535
server
configuration
settings 529
tuning 533
configuring 518
authentication 517
host list authentication 525
SSH users 521
user authentication 520
displaying information about 534
PKI key pairs, creating 518
sample session (UNIX) 528
SSH connection process 526
server applications 543
secure copy protocol (UNIX), using 548
SFTP 543
SFTP access
user configuration, verifying 545
SFTP access, configuring 544
SFTP session (UNIX), starting 546
SSH server
user password authentication, enabling 95
stack command 33
components, adding 33
components, removing 33
getting help about 36
making dynamic changes 36
Static route
adding 191
configuring 191
displaying 192
status command 19, 27

Subject distinguished name
configuring 446
Switched Multimegabit Data Service
See also SMDS
sync command 86
accessing target systems 86
components, transferring with security features 86
help, displaying 88
modes of operation 87
session variables, customizing 87
System management
administrative community, designating 42
authentication and privacy applications, specifying 44
authentication and privacy keys, generating 44
boot order, configuring 55
CLI, restricting user access to 44
command output, saving in a file 59
components, transferring to another Access Point 86
configuration
changes, displaying 56
saving 54
script commands, generating 85
status, displaying 57
summary, displaying 56
DNS client, configuring 62
engine ID, configuring 43
file transfer 51
ordinary transfer 53
secure transfer 52
managing events 73
NVRAM compression state, displaying 58
pre-login banner,customizing 60
Scheduler, using 66
SNMP communities, setting up 38
SNMP traps, configuring 45
SNMPv1 and SNMPv2 connections, disabling 44
SNMPv3
configuring a user for 43
parameters, configuring 43
using to secure the AP 43
system name, changing 46
system time, changing 46
TCL scripts, creating 70
Telnet, restricting user access to 46
transferable components 86
transferring files 47
interpreting errors 53
user access, restricting 44
watchdog timer, configuring 46
Web, restricting user access to 46
System name
assigning with Access View 458
changing 46

T

T1
about 103
interface, configuring 103
setting up the interface 105
clock source 106
data link facilities 106
examining physical layer defaults 105
examining T1 defaults 107
for Fractional T1 112
idle code 107
interface type 106
invert HDLC 106
line coding 106
loopback mode 110
forcing local into 110
forcing remote into 111
local analog loopback 110
local framer loopback 110
remote line loopback 110
remote payload loopback 110
T3
interface, configuring 115
displaying physical layer defaults 115
parameters 116
circuit ID 116
line coding 116
line length 116
line type 116
loopback config 116
send code 116

transmit clock source 116
payload scrambler, using 117
setting line length 116
TCL scripts
autosourcing 71
creating 70
running 71
running automatically 71
sample script 72
scripts, displaying available 71
transferring to Access Point systems 70
traceroute command
description 21
options 21
destination host 21
IP address of source host 21
maximum number of hops 21
number of packets to send 21
packet length 21
print hop addresses numerically 21
time to wait 21
type of service 21
UDP port number 21
verbose output 21
Traffic Class Graphing wizard
using 482
Traffic classes
adding with Access View 465
bandwidth threshold 290
CBQ, configuring to select a virtual connection 141
CLP bit, setting 142
configuring 277
datalink-traffic-flag parameter, setting 143
IPOA layer
adding a CBQ instance over 141
mapping to specific Frame Relay DLCIs 167
mapping traffic classes 141
peer classification order 301
enhancing security 302
filtering specific traffic types 301
sample class definition 277
sample configuration 278
traffic parameters, using in combination 300
Traffic classification
directionality 4
inbound and outbound 4
Traffic descriptors
adding to virtual channel links 130
defining 132
Traffic Status application
accessing 469
bandwidth
allocation, displaying 469, 472
utilization, displaying 469, 473
CBQ tree structure, displaying 471
description of frames 470
Bandwidth Allocation 472
CBQ Tree 471
Traffic Equalizer 473
display preferences, setting 479
help information, displaying 477
overview 469
Transferring files 47, 51
security 49
Tunnels
client tunnels, configuring 404
authentication
by RADIUS 440
locally 440
authentication method 405
authentication, disabling 441
digital certificates, assigning 405
preshared key, assigning 405
router address
designating 393, 404
user authentication 409, 439
user password authentication 440
user password authentication, enabling 95
configuring client tunnels 406
displaying
general subinterface information 411
information about tunnel counters 413
information about tunnel security associations 410
summary information 409
dynamic site-to-site tunnels, configuring 402
L2TP, configuring 424

monitoring 409
security profile parameters 396, 398
security profiles, configuring 394, 399, 405
site-to-site tunnels, configuring 399, 401
IKE Keepalive protocol 414
preshared keys assigning 400
remote gateway 400
router address
designating 393, 399
static security associations 403
subinterfaces 399
wildcard tunnels 403
static site-to-site tunnels, configuring 402
Transport Mode connections, configuring 417
client 420
site-to-site 417
user authentication, configuring 89
user password authentication, enabling 95

U

User authentication
configuring 89
default user
about 90
enabling 95
user table
about 89
configuring users 90
specifying authentication method 91
specifying client groups 93
specifying password 90
specifying SSH privileges 93
specifying user name 90

V

VBRP
about 267
configuring 267, 269
description 267
key terms 268
backup 268
master 268
options 256, 270
authentication-key 256, 270
hello-interval 256, 270
mode 256, 270
priority 256, 270
protocol 256, 270
Virtual LANs (VLANs)
configuration, displaying 559
configuring 557
Ethernet subinterface 559
dropping traffic 557
dynamic configuration, using 558
prioritizing traffic 562
receiving traffic 557
static configuration, using 558
stripping off 802.1Q tags 558
VLAN IDs
assigning to Ethernet subinterface 561
using 557
Virtual Private Networking
advantages 380
cost 381
scalability 382
security 381
client tunnels
authentication
by RADIUS 440
locally 440
authentication, disabling 441
configuring
security profiles 394, 405
user authentication 409, 439
configuring an authentication method 405
digital certificates, assigning 405
preshared key, assigning 405
router address
designating 393, 404
security profile parameters 398
user password authentication 440
common applications 382
configuring
client tunnels 404
L2TP tunnels 424
site-to-site tunnels 399
defined 380
displaying

general subinterface information 411
information about tunnel counters 413
information about tunnel security associations 410
summary information 409
Internet Key Exchange (IKE)
about 387
IPSec
about 386
Authentication Header (AH) 386
client tunnel
authentication 391
establishing a session 390
how it works 389
overview 388
Encapsulating Security Payload (ESP) 386
Internet Key Exchange (IKE) 386
site-to-site tunnel overview 387
Transport Mode connections, configuring 417
Transport Mode overview 392
L2TP
about 424
configuring 431
IP interface 431
IP LAN interface 431
tunnel domains 432
verifying layers are enabled 431
how it works 426
tunnel information, displaying 436
Virtual Tunnel Transport stack 384, 427
multicast
Virtual Tunnel Transport stack 384
site-to-site tunnels
configuring 401
dynamically 402
security profiles 394, 399
static security associations 403
statically 402
wildcard tunnels 403
IKE Keepalive protocol 414
preshared keys assigning 400
remote gateway 400
router address
designating 393, 399
security profile parameters 396
subinterfaces 399
tunnel types 383
Virtual Tunnel Transport stack 384
Virtual Tunnel Transport stack
capabilities 384
components 384
VPN
See also Virtual Private Networking
stack command, configuring with 33
VRRP
about 253
configuring 253, 255
description 253
key terms 254
backup 255
master 255
owner 254
options 256, 270
authentication-key 256, 270
hello-interval 256, 270
mode 256, 270
priority 256, 270
protocol 256, 270
priority for backup routers, setting 256
tracking an IP interface 257

W

Watchdog timer
configuring 46
where command 32