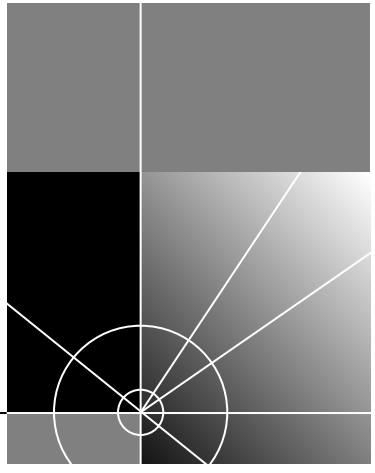
CoreBuilder® 3500 Implementation Guide

Release 3.0



http://www.3com.com/

Part No. 10013506
Published November 1999

3Com Corporation
5400 Bayfront Plaza
Santa Clara, California
95052-8145

3Com Corporation reserves the right to revise this documentation and to make changes in content from time to time without obligation on the part of 3Com Corporation to provide notification of such revision or change.

3Com Corporation provides this documentation without warranty, term, or condition of any kind, either implied or expressed, including, but not limited to, the implied warranties, terms, or conditions of merchantability, satisfactory quality, and fitness for a particular purpose. 3Com may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time.

If there is any software on removable media described in this documentation, it is furnished under a license agreement included with the product as a separate document, in the hardcopy documentation, or on the removable media in a directory file named LICENSE.TXT or !LICENSE.TXT. If you are unable to locate a copy, please contact 3Com and a copy will be provided to you.

UNITED STATES GOVERNMENT LEGEND

If you are a United States government agency, then this documentation and the software described herein are provided to you subject to the following:

All technical data and computer software are commercial in nature and developed solely at private expense. Software is delivered as "Commercial Computer Software" as defined in DFARS 252.227-7014 (June 1995) or as a "commercial item" as defined in FAR 2.101(a) and as such is provided with only such rights as are provided in 3Com's standard commercial license for the Software. Technical data is provided with limited rights only as provided in DFAR 252.227-7015 (Nov 1995) or FAR 52.227-14 (June 1987), whichever is applicable. You agree not to remove or deface any portion of any legend provided on any licensed program or documentation contained in, or delivered to you in conjunction with, this User Guide.

Unless otherwise indicated, 3Com registered trademarks are registered in the United States and may or may not be registered in other countries.

3Com, the 3Com logo, CoreBuilder, Dynamic*Access*, NETBuilder II, PACE, SmartAgent, SuperStack, and Transcend are registered trademarks of 3Com Corporation. 3Com Facts is a service mark of 3Com Corporation.

Postscript is a registered trademark of Adobe Systems, Inc. AppleTalk is a registered trademark of Apple Computer, Incorporated. Banyan and VINES are registered trademarks of Banyan Worldwide. DEC, DECnet, and PATHWORKS are registered trademarks of Compaq Computer Corporation. OpenView is a registered trademark of Hewlett-Packard Company. AIX, IBM, and NetView are registered trademarks and NetBIOS is a trademark of International Business Machines Corporation. Internet Explorer, Microsoft, MS-DOS, Windows, and Windows NT are registered trademarks of Microsoft Corporation. Netscape, Netscape Navigator, and the Netscape N and Ship's Wheel logos are registered trademarks of Netscape Communications Corporation in the United States and other countries. IPX, Novell, and NetWare are registered trademarks of Novell, Inc. Sun and SunNet Manager are trademarks of Sun Microsystems, Inc. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. Xerox and XNS are trademarks of Xerox Corporation.

All other company and product names may be trademarks of the respective companies with which they are associated.

Guide written by 3Com MSD Technical Publications, Marlborough, MA, USA.
We welcome your comments on this guide. Send e-mail to: sdtechpubs_comments@ne.3com.com

CONTENTS

ABOUT THIS GUIDE

Conventions 22
CoreBuilder 3500 Documentation 24
  Paper Documents 24
  Software and Documents on CD-ROM 26
Documentation Comments 26
Year 2000 Compliance 26

1 CONFIGURATION OVERVIEW

System Configuration Procedure 27
  Procedure Summary 27
  Configuration Procedure 28

2 MANAGEMENT ACCESS

Management Access Overview 31
  Administration Console Overview 32
  SNMP-Based Network Management Overview 34
Key Concepts 34
  OSI Protocols 34
  Protocols 35
Key Guidelines for Implementation 38
  Access Methods 38
Administration Console Access 40
  Password Levels 40
  Terminal Port Access 41
  Modem Port Access 41
Web Management Access 42
  Browser Requirements 42
SNMP Access 43

3 SYSTEM PARAMETERS

System Parameters Overview 46
Features 46
Benefits 47
Key Concepts 47
Key Guidelines for Implementation 48
File Transfer 48
Implementing FTP 48
Implementing TFTP 49
Security 49
Security Options 50
Important Considerations 51
Software Update 52
Important Considerations 52
nvData Operations 53
Saving nvData 53
Restoring nvData 54
Resetting nvData 55
Viewing nvData 55
Simple Network Time Protocol (SNTP) 56
SNTP Overview 56
Implementing SNTP 57
Standards, Protocols, and Related Reading 57

4 PHYSICAL PORT NUMBERING

Port Numbering Overview 59
Numbering Rules 59
Supported Module Types 60
Key Guidelines for Implementation 61
Examples of Port Numbering 62
Example 1: Fully Loaded System 62
Example 2: Empty Slot in the System 63
Example 3: Gigabit Ethernet Module with Other Modules 64
Example 4: FDDI Module with Other Modules 65
Effects of Removing a Module 66
Port-Numbering Changes 66
VLAN Changes 66
Trunk Changes 67

Effects of Replacing Modules 68
Replacing Modules of the Same Type or Same Number of Ports 68
Replacing Modules of Different Types 68

5 ETHERNET

Ethernet Overview 72
Features 72
Benefits 73
Key Concepts 74
Ethernet Frame Processing 76
Key Guidelines for Implementation 78
Link Bandwidths 78
Trunks 78
Port Enable and Disable (Port State) 79
Important Considerations 79
Port Labels 79
Labeling Ports 79
Autonegotiation 80
Important Considerations 80
Port Mode 82
Important Considerations 82
Flow Control 83
Important Considerations 83
PACE Interactive Access 84
Important Considerations 84
Standards, Protocols, and Related Reading 84
Ethernet Protocols 84
Media Specifications 85
Related Reading 85

6 FIBER DISTRIBUTED DATA INTERFACE (FDDI)

FDDI Overview 88
- Features 88
- Benefits 88

Key Concepts 89
- Related Standards 89
- FDDI Network Topologies 91
- Nodes and Attachments 93
- Dual Homing 97
- FDDI Stations 97
- Primary and Secondary Paths 99
- Media Access Control 99
- Ports 100

Key Guidelines for Implementation 101

FDDI Stations 102
- Setting the Connection Policies 102
- Setting Neighbor Notification Timer 104
- Enabling and Disabling Status Reporting 104

FDDI Paths 104
- Setting tvxLowerBound 104
- Setting tmaxLowerBound 105
- Setting maxT-Req 105

FDDI MACs 106
- Setting the Frame Error Threshold 106
- Setting the Not Copied Threshold 106
- Enabling and Disabling LLC Service 107

FDDI Ports 107
- Setting lerAlarm 107
- Setting lerCutoff 108
- Setting Port Labels 108

Station Mode (DAS and SAS) 109
- Single Attachment Station (SAS) 109
- Dual Attachment Stations 109

Sample FDDI Configurations 109

Standards, Protocols, and Related Reading 111
- Requests For Comments (RFCs) 111
- Standards Organizations 111
- Related Reading 111

7 BRIDGE-WIDE AND BRIDGE PORT PARAMETERS

Bridging Overview 114
- Benefits 114
- Features 115

Key Bridging Concepts 116
- Learning Addresses 116
- Aging Addresses 116
- Forwarding, Filtering, and Flooding Packets 116
- Spanning Tree Protocol 117

How the Spanning Tree Protocol Works 119
- CBPDUs at Work 119
- How a Single Bridge Interprets CBPDUs 123
- How Multiple Bridges Interpret CBPDUs 124
- Spanning Tree Port States 129
- Reconfiguring the Bridged Network Topology 131

Key Guidelines for Implementation 132

STP Bridge and Port Parameters 134
- Administering Bridge-wide STP Parameters 134
- Administering STP Parameters on Bridge Ports 136

Frame Processing 137

MAC Address Table 138
- Aging Time 138
- Address Threshold 138
- Important Considerations 138

IP Fragmentation 139

IPX SNAP Translation 139

Broadcast and Multicast Limit for Bridge Ports 140
- Important Considerations 140

GARP VLAN Registration Protocol (GVRP) 141
- Important Considerations 141

Standards, Protocols, and Related Reading 142

8 TRUNKING

Trunking Overview 144
- Features 144
- Benefits 144

Key Concepts 145
- Port Numbering in a Trunk 145
- Trunk Control Message Protocol (TCMP) 146

Key Guidelines for Implementation 147
- General Guidelines 147
- Trunk Capacity Guidelines 148

Defining Trunks 150
- Important Considerations 150

Modifying Trunks 152
- Important Considerations 152

Removing Trunks 153
- Important Considerations 153

Standards, Protocols, and Related Reading 154

9 VIRTUAL LANS

VLAN Overview 156
- Need for VLANs 156
- Benefits 157
- Features 158

Key Concepts 159
- Related Standards and Protocols 159
- VLAN IDs 160
- Terminology 161

Key Guidelines for Implementation 163
- Network-based VLANs vs. Multiple Interfaces per VLAN 163
- VLANs Created by Router Port IP Interfaces 164
- Number of VLANs 165
- General Guidelines 167

VLAN allOpen or allClosed Mode 169
- Important Considerations 169
- Modifying the VLAN Mode 170
- Mode Requirements 171

Ignore STP Mode 172
Important Considerations 172
VLAN Aware Mode 174
Port-based VLANs 175
The Default VLAN 175
Static Port-based VLANs 178
Dynamic Port-based VLANs Using GVRP 182
Protocol-based VLANs 186
Important Considerations 186
Selecting a Protocol Suite 187
Establishing Routing Between VLANs 189
Network-based IP VLANs 192
Important Considerations 192
Example of Network-based VLANs 193
Rules of VLAN Operation 195
Ingress Rules 195
Egress Rules 198
Examples of Flooding and Forwarding Decisions 200
Rules for Network-based (Layer 3) VLANs 202
Modifying and Removing VLANs 206
Monitoring VLAN Statistics 207

10 PACKET FILTERING

Packet Filtering Overview 210
What Can You Filter? 210
When Is a Filter Applied? — Paths 211
Path Assignment 212
Key Concepts 213
Standard Packet Filters 213
Custom Packet Filters 214
Important Considerations 215
Managing Packet Filters 215
Tools for Writing Filters 217
ASCII Text Editor 217
Built-in Line Editor 217
Web Management Filter Builder Tool 219

Downloading Custom Packet Filters 221
Download with Filter Builder 221
Download an ASCII File 222
The Packet Filtering Language 224
Principles for Writing a Custom Filter 224
How the Packet Filter Language Works 224
Procedure for Writing a Custom Filter 225
Packet Filter Opcodes 228
Implementing Sequential Tests in a Packet Filter 233
Common Syntax Errors 235
Custom Packet Filter Examples 237
Destination Address Filter 237
Source Address Filter 237
Length Filter 237
Type Filter 238
Ethernet Type IPX and Multicast Filter 238
Multiple Destination Address Filter 238
Source Address and Type Filter 239
Accept XNS or IP Filter 239
XNS Routing Filter 240
Port Group Filter 240
Limits to Filter Size 241
Using Port Groups in Custom Packet Filters 242
Port Group Packet Filter Example 242
Port Group Filter Operation 242
Port Group Management and Control Functions 245
Defining Port Groups 245
Long Custom Filter Example 247
Filtering Problem 247
Packet Filter Solution 247
Optimizing the Filter with Accept and Reject Commands 254

11 INTERNET PROTOCOL (IP)

Routing Overview 258
- Routing in a Subnetted Environment 259
- Integrating Bridging and Routing 260
- IP Routing Overview 261
- Features and Benefits 262

Key Concepts 263
- Multiple IP Interfaces per VLAN 263
- Media Access Control (MAC) Address 263
- Network-Layer Address 264
- IP Addresses 264
- Variable Length Subnet Masks (VLSMs) 268
- Router Interfaces 271
- Routing Table 272

Routing Models: Port-based and VLAN-based 274
- Role of VLANs in IP Routing 275
- Port-based Routing 276
- VLAN-based Routing 280

Key Guidelines for Implementing IP Routing 282
- Configure Trunks (Optional) 282
- Configure IP VLANs (VLAN-based Routing) 282
- Establish Your IP Interfaces 283
- Enable IP Routing 285
- Administering IP Routing 285

Address Resolution Protocol (ARP) 286
- Important Considerations 288

ARP Proxy 288
- Important Considerations 288
- Example 288

Internet Control Message Protocol (ICMP) 290

ICMP Redirect 292
- Important Considerations 292
- Example 293

ICMP Router Discovery 294
- Important Considerations 294
- Example 294

Broadcast Address 295
- Important Considerations 295

Directed Broadcast 295
Important Considerations 295
Routing Information Protocol (RIP) 296
Basic RIP Parameters 296
RIP Mode 297
Compatibility Mode 297
Cost 297
Poison Reverse 298
Advertisement Address 298
Route Aggregation 299
RIP-1 Versus RIP-2 299
Important Considerations 300
Routing Policies 300
How Routing Policies Work 301
Important Considerations 303
Implementing RIP Routing Policies 303
Setting Up RIP Routing Policies 306
Creating RIP Routing Policies 307
Example 308
Domain Name System (DNS) 310
Important Considerations 310
User Datagram Protocol (UDP) Helper 311
Configuring Overlapped Interfaces 312
Important Considerations 313
Standards, Protocols, and Related Reading 313
Requests For Comments (RFCs) 313
Standards Organizations 314
Related Reading 314

12 VIRTUAL ROUTER REDUNDANCY PROTOCOL (VRRP)

VRRP Overview 316
Router to Router 316
Host to Host and Host to Gateway 316
Example 317
Key Concepts 318
How VRRP Works 319
Virtual Router Decision-making 320
Important Considerations 322

Implementing VRRP 323
Create VLANs 324
Configure IP Interfaces 324
Configure the Router Protocol 325
Enable Routing 325
Configure VRRP 325
Enable VRRP 326
VRRP and Other Networking Operations 326
Spanning Tree Protocol (STP) 327
Dynamic Routing Protocols (RIP, RIP-2, OSPF) 327
IGMP Queries 328
ICMP Redirect 329
Quality of Service 329
IP Routing Policies 329
Dynamic Host Configuration Protocol (DHCP) 329
Standards, Protocols, and Related Reading 329

13 IP MULTICAST ROUTING
IP Multicast Overview 332
Unicast Model 332
Broadcast Model 332
Multicast Model 332
Benefits of IP Multicast 333
How a Network Supports IP Multicast 334
IP Multicast Routing 334
IP Multicast Tunnels 335
IP Multicast Filtering 336
Internet Support for IP Multicast 337
Key Concepts 338
Traffic Movement 338
IP Multicast Groups 338
Source-Group Pairs 338
Multicast Addresses 339
How IGMP Supports IP Multicast 341
Electing the Querier 341
Host Messages 341
Role of IGMP in IP Multicast Filtering 342

How DVMRP Supports IP Multicast 343
Spanning Tree Delivery 343
Managing the Spanning Tree 344
DVMRP Interface Characteristics 346
Key Guidelines for Implementation 347
Configuration Procedure 347
Impact of Multicast Limits 348
Impact of IEEE 802.1Q on Multicasts 348
Protocol Interoperability 348
Configuring IGMP Options 349
Querying and Snooping Modes 349
Important Considerations 349
Configuring DVMRP Interfaces 349
Important Considerations 349
Configuring DVMRP Tunnels 350
Important Considerations 350
Configuring DVMRP Default Routes 352
How Default Routes Work 352
How to Configure A Default Route 352
Viewing the DVMRP Routing Table 353
Viewing the DVMRP Cache 353
Using IP Multicast Traceroute 354
Standards, Protocols, and Related Reading 355

14 OPEN SHORTEST PATH FIRST (OSPF)

OSPF Overview 358
Features 358
Benefits 360
Key Concepts 363
Autonomous Systems 363
Areas 363
Neighbors and Adjacency 363
Router Types 364
Protocol Packets 365
How OSPF Routing Works 366
Key Guidelines for Implementing OSPF 369
Autonomous System Boundary Routers 370
Configuring an ASBR 370

Areas 372
Types of Areas 373
Area Border Routers 375
Routing Databases 375
Configuring Route Summarization in ABRs 376
Important Considerations 376
Default Route Metric 379
OSPF Interfaces 380
Mode 380
Priority 380
Area ID 381
Cost 381
Delay 382
Hello Interval 383
Retransmit Interval 383
Dead Interval 384
Password 384
Statistics 384
Important Considerations 385
Link State Databases 387
Router Link State Advertisements 387
Network Link State Advertisements 388
Summary Link State Advertisements 389
External Link State Advertisements 389
Important Considerations 391
Neighbors 392
Neighbor Information 392
Static Neighbors 395
Important Considerations 395
Router IDs 396
Important Considerations 396
OSPF Memory Partition 397
Default Memory Allocation 397
Running Out of Memory — Soft Restarts 398
Manual Memory Allocation 399
System Memory Allocation 399
Stub Default Metrics 400
Important Considerations 400

Virtual Links 401
Important Considerations 402
OSPF Routing Policies 403
Important Considerations 404
Implementing Import Policies 405
Implementing Export Policies 408
OSPF Statistics 416
Standards, Protocols, and Related Reading 417

15 IPX ROUTING

IPX Routing Overview 419
Features 420
Benefits 420
Key Concepts 421
How IPX Routing Works 421
Terminology 426
Key Guidelines for Implementation 427
Procedural Guidelines 427
General Guidelines 427
IPX Interfaces 428
Important Considerations 428
Per-Interface Options 429
IPX Routes 430
Important Considerations 430
Primary and Secondary Routes 431
Static Routes 431
Dynamic Routes Using RIP 431
Routing Tables 432
Selecting the Best Route 433
IPX Servers 434
Important Considerations 434
Primary and Secondary Servers 435
Static Servers 435
Dynamic Servers Using SAP 435
Maintaining Server Information 435
Server Tables 436
IPX Forwarding 437
Important Considerations 437

IPX RIP Mode 438
Important Considerations 438
RIP Policies 439
IPX SAP Mode 441
Important Considerations 441
SAP Policies 441
IPX Statistics 443
Standards, Protocols, and Related Reading 444

16 APPLETALK
AppleTalk Overview 445
Features 446
Benefits 447
Key Concepts 448
AppleTalk Protocols 448
AppleTalk Network Elements 454
Terminology 455
Key Implementation Guidelines 457
AppleTalk Interfaces 458
Important Considerations 459
AppleTalk Routes 460
Important Considerations 460
AppleTalk Address Resolution Protocol (AARP) Cache 462
AppleTalk Zones 464
Important Considerations 465
Changing Zone Names 466
Forwarding AppleTalk Traffic 468
Enabling Forwarding 468
Disabling Forwarding 468
Important Considerations 468
Checksum Error Detection 469
Important Considerations 469
AppleTalk Echo Protocol (AEP) 469
AppleTalk Statistics 470
Datagram Delivery Protocol 470
Routing Table Maintenance Protocol 471

Zone Information Protocol 472
Name Binding Protocol 473
Standards, Protocols, and Related Reading 474

17 QoS AND RSVP

QoS Overview 476
Features 476
Benefits 476
Methods of Using QoS 477
Key Concepts 478
Related Standards and Protocols 478
Terminology 479
Key Guidelines for Implementation 482
Procedural Guidelines 482
General Guidelines 482
QoS Classifiers 483
Important Considerations 483
Using Predefined Classifiers 483
Assigning Flow and Nonflow Classifier Numbers 484
Defining Flow Classifiers 485
Defining Nonflow Classifiers 488
QoS Controls 489
Important Considerations 489
Assigning Control Numbers 490
Specifying Rate Limits 492
Specifying Service Levels 493
Specifying TCP Drop Control 494
Setting the QoS Timer Control 495
Examples of Classifiers and Controls 497
Example 1: Traffic to and from a Specific Server 497
Example 2: Filtering Traffic to a Destination 499
Example 3: Using Two Classifiers to Filter Traffic 501
Example 4: Assigning High Priority to Specific Traffic 504
Example 5: Nonflow Multimedia Tagged Traffic 506
Example 6: Bridged Nonflow IP Unicast Traffic 508
Modifying and Removing Classifiers and Controls 510
Important Considerations 510

QoS Excess Tagging 511
Example of QoS Excess Tagging 511
Transmit Queues and QoS Bandwidth 513
LDAP 514
Important Considerations 514
Operation 515
RSVP 516
RSVP Terminology 517
Example of RSVP 518
Setting RSVP Parameters 520

18 DEVICE MONITORING

Device Monitoring Overview 522
Key Concepts and Tools 522
Administration Console 522
Web Management Tools 522
Network Management Platform 523
SmartAgent Embedded Software 523
Other Commonly Used Tools 523
Event Logging 524
Important Consideration 524
Displaying the Event Log Configuration 524
Configuring the Output Devices 524
Configuring the Services 524
Baselining 525
Important Considerations 525
Displaying the Current Baseline 525
Setting a Baseline 525
Enabling or Disabling Baselines 525
Roving Analysis 526
Key Guidelines for Implementation 527
Important Considerations 527
Ping 530
Important Consideration 530
Using Ping 530
Ping Responses 530
Strategies for Using ping 531

traceRoute 532
Using traceRoute 532
traceRoute Operation 532
SNMP 533
SNMP Overview 533
Setting Up SNMP on Your System 538
Remote Monitoring (RMON) 541
Overview of RMON 542
RMON Benefits 543
RMON in Your System 544
3Com Transcend RMON Agents 545
Important Considerations 546
RMON-1 Groups 547
RMON-2 Groups 552
Management Information Base (MIB) 556
MIB Files 556
Compiler Support 558
MIB Objects 559
MIB Tree 560
MIB-II 562
RMON-1 MIB 563
RMON-2 MIB 564
3Com Enterprise MIBs 565

A TECHNICAL SUPPORT
Online Technical Services 567
World Wide Web Site 567
3Com Knowledgebase Web Services 567
3Com FTP Site 568
3Com Bulletin Board Service 568
3Com Facts Automated Fax Service 569
Support from Your Network Supplier 569
Support from 3Com 569
Returning Products for Repair 571

INDEX

About This Guide

This guide describes information that you need to use features of the CoreBuilder® 3500 system after you install it and attach the system to your network. Before you use this guide:

- Verify that your system is installed and set up using the *CoreBuilder 3500 Getting Started Guide*.
- Become familiar with the *Command Reference Guide*, which documents the commands that are used to configure and manage your system.
- Read Chapter 1 for an overview of the configuration process.

This guide is intended for the system or network administrator who is responsible for configuring, using, and managing the CoreBuilder 3500 system. It assumes a working knowledge of local area network (LAN) operations and familiarity with communications protocols that are used on interconnected LANs.

If the information in the release notes differs from the information in this guide, follow the instructions in the release notes.

The most current versions of guides and release notes are available in Adobe Acrobat Reader Portable Document Format (PDF) or HTML from the 3Com World Wide Web site:

http://www.3com.com

Conventions

Table 1 and Table 2 list conventions that are used throughout this guide.

Table 1 Notice Icons

Icon	Notice Type	Description
	Information note	Information that describes important features or instructions
	Caution	Information that alerts you to potential loss of data or potential damage to an application, system, or device
	Warning	Information that alerts you to potential personal injury
	Layer 2 switch	In figures, a switch that can perform Layer 2 functions
	Layer 3 switch	In figures, a switch that can perform both Layer 2 and Layer 3 functions

Table 2 Text Conventions

Convention	Description
`Screen displays`	This typeface represents information as it appears on the screen.
`Syntax`	The word "syntax" means that you evaluate the syntax provided and then supply the appropriate values for the placeholders that appear in angle brackets. Example: To set the system date and time, use the following syntax: `CCYY-MM-DDThh:mm:ss`
`Commands`	The word "command" means that you must enter the command exactly as shown and then press Return or Enter. Commands appear in bold. Example: To remove an IP interface, enter the following command: **`ip interface remove`**
The words "enter" and "type"	When you see the word "enter" in this guide, you must type something, and then press Return or Enter. Do not press Return or Enter when an instruction simply says "type."
Keyboard key names	If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press Ctrl+Alt+Del.

Table 2 Text Conventions (continued)

Convention	Description
Words in *italics*	Italics are used to: ■ Emphasize a point ■ Denote a new term at the place where it is defined in the text ■ Identify menu names, menu commands, and software button names. Examples: From the *Help* menu, select *Contents*. Click *OK*.

CoreBuilder 3500 Documentation

The following documents comprise the CoreBuilder 3500 documentation set. Documents are available in one of two forms:

- Paper Documents

  The paper documents that are shipped with your system and components are listed in the next section.

- Software and Documents on CD-ROM

  The System Software and Documentation CD contains online versions of the paper documents, this *Implementation Guide*, and the *Command Reference Guide*, as well as the CoreBuilder 3500 system software.

To order additional copies of the paper documents and the CD-ROM, contact your sales representative.

Paper Documents

These documents are shipped with your system:

- *CoreBuilder 3500 Unpacking Instructions*

  How to unpack your system. Also, an inventory list of all the items that are shipped with your system.

- *CoreBuilder 3500 Software Installation and Release Notes*

  Information about the software release, including new features, software corrections, and known problems. It also describes any changes to the documentation.

- *CoreBuilder 3500 Quick Installation Guide*

  Quick reminders and information for system installation. For greater detail on installation procedures, see the *CoreBuilder 3500 Getting Started Guide*.

- *CoreBuilder 3500 Getting Started Guide*

  All the procedures necessary for getting your system up and running, including information on installing, cabling, powering up, configuring, and troubleshooting the system.

- *CoreBuilder 3500 Command Quick Reference*

  All of the Administration Console commands for the system.

- *Web Management User Guide for the CoreBuilder 3500*

  Overview, installation, and troubleshooting information for the suite of applications that help you manage your system over the Internet.

In addition, each module and field-replaceable component contains a guide:

- *CoreBuilder 3500 System Processor Removal and Replacement Guide*

  Provides overview information and removal and replacement instructions for the CoreBuilder system processor.

- *Module Installation Guides*

  An overview, LED status information, and installation instructions for each module.

- *GBIC Transceiver Installation Guide*

  Installation instructions for the Gigabit Ethernet Interface Converter transceiver.

- *CoreBuilder 3500 Power Supply Assembly Removal and Replacement Guide*

  Overview information and removal and replacement instructions for the CoreBuilder power supplies.

- *CoreBuilder 3500 Fan Tray Removal and Replacement Guide*

  Overview information and removal and replacement instructions for the fan tray.

- *PCMCIA Flash Card User Guide*

  Information on using the PCMCIA card to save and restore system configuration settings.

- *Blank Faceplate Installation Guide*

  Instructions for covering empty slots with the blank faceplate.

Software and Documents on CD-ROM

The compact disc that comes with your system contains:

- System software
- Online versions of the paper guides that are shipped with your system, modules, and field-replaceable components
- *CoreBuilder 3500 Implementation Guide* (this guide)
- Multiplatform *Command Reference Guide*

  Information about the commands used to configure the system. This guide documents commands for the CoreBuilder 3500 as well as other 3Com systems.
- Help system for the Web Management suite of applications

  Online Help system for the CoreBuilder 3500 Web Management software. See the *Web Management User Guide* for information about Web Management and the related Help system.

Documentation Comments

Your suggestions are very important to us. They help us to make our documentation more useful to you.

Please send e-mail comments about this guide to:

sdtechpubs_comments@ne.3com.com

Please include the following information when commenting:

- Document title
- Document part number (found on the inside title page of this guide)
- Page number

Example:

CoreBuilder 3500 Implementation Guide

Part Number 10013506

Page 25

Year 2000 Compliance

For information on Year 2000 compliance and 3Com products, visit the 3Com Year 2000 Web page:

http://www.3com.com/products/yr2000.html

1 CONFIGURATION OVERVIEW

This chapter provides the configuration procedure for the first time that you install a CoreBuilder® 3500 Layer 3 High-Function Switch.

To upgrade the software on an existing switch, see the Software Installation and Release Notes *for configuration information.*

System Configuration Procedure

Software is installed on each system at the factory. Because the software boots from flash memory when you power on the system, the system is immediately ready to configure according to your network needs.

3Com recommends that you use the following procedures the first time that you set up your system and every time that you modify its configuration.

Procedure Summary

These steps are described in detail in the next section:

1. Establish management access.
2. Choose a subsequent management access method.
3. Choose a subsequent management interface.
4. Configure parameters related to the network infrastructure. These include system, bridge-wide and bridge-port, Ethernet, FDDI, and trunking parameters.
5. Define all VLANs.
6. Define routing protocol interfaces and set related parameters.
7. Configure more advanced traffic control features, such as packet filters and Quality of Service (QoS).
8. Monitor the system and analyze network activity.

These steps are described in detail in the next section.

Configuration Procedure

Follow the steps that apply to your network needs and ignore the steps that do not apply.

1 Establish management access.

To perform configuration or management tasks, you must initially:

a Connect to the system through its terminal serial port or modem serial port.

For information about the required settings for the serial ports, see Chapter 2 in this guide.

b Use the Administration Console as the management interface.

The Administration Console is a menu-driven command line interface that is embedded in the system software. For specific menu and command information, see the *Command Reference Guide*.

2 Choose a subsequent management access method.

You can continue to access your system through a local serial connection, or you can use one of two other local access methods — any in-band port on a media module or the out-of-band 10BASE-T port on the system processor module. To manage the system through either access method, you must first configure an IP address:

- **To configure an IP address for an out-of-band port** — Using the serial port connection from step 1, configure an IP address through the `management ip interface` menu. For more information, see Chapter 2.
- **To configure an IP address for an in-band port** — Using the serial port connection from step 1, configure an IP address by defining an IP VLAN (through the `bridge vlan` menu; see Chapter 9) and an associated IP interface (through the `ip interface` menu; see Chapter 11).

3 Choose a subsequent management interface.

After you configure an IP address, you have additional management interface options:

- **Administration Console** — You can now access this interface from a remote Telnet connection.
- **Web Management software** — From your Web browser, you can access a suite of HTML-based applications that are embedded in the software. For more information, see the *Web Management User Guide*.
- **SNMP-based applications** — One example is 3Com Transcend® Network Control Services software. To manage the system in-band from SNMP-based applications, set the SNMP parameters through the `snmp` menu. For more information, see Chapter 2 and Chapter 18 in this guide, as well as application-specific documentation.

4 Configure parameters related to the network infrastructure.

One or more of the following topics may apply to your system, depending on your network requirements:

- **System parameters** — To choose the file transfer protocol, administer nonvolatile data (nvData), perform system software updates, and display your system configuration, see Chapter 3.
- **Physical port numbering** — To learn the port numbering rules and understand the effects of adding or removing modules, see Chapter 4.
- **Ethernet** — To label Ethernet ports, set the port mode, enable flow control, and control autonegotiation and other settings, see Chapter 5.
- **FDDI** — To configure stations, paths, MACs, and ports, see Chapter 6.
- **Bridge-wide and bridge port parameters** — To set parameters for Spanning Tree Protocol (STP), GARP VLAN Registration Protocol (GVRP), IPX SNAP translation, and IP fragmentation, see Chapter 7.
- **Trunks** — To increase the bandwidth and resiliency between two points, you can aggregate many individual links into a single logical link called a trunk. You must configure trunks before you define VLANs. For more information, see Chapter 8.

5 Define all VLANs.

To create logical workgroups, which are generally equivalent to Layer 2 broadcast domains or Layer 3 networks, you can define port-based, protocol-based, and network-based VLANs, and set related modes in the system. You must define VLANs after you define trunks and before you define routing interfaces. For more information, see Chapter 9.

6 Configure routing interfaces and set related parameters.

You can use the following protocols to configure routing interfaces and set related parameters:

- **IP** — See Chapter 11.
- **VRRP** — See Chapter 12.
- **IP multicast** — See Chapter 13.
- **OSPF** — See Chapter 14.
- **IPX** — See Chapter 15.
- **AppleTalk** — See Chapter 16.

7 Configure more advanced traffic control features:

- **Packet filters —** To improve LAN performance, shape traffic flows, or implement security controls with standard, custom, predefined, and port group packet filters, see Chapter 10.
- **Quality of Service (QoS) and the Resource Reservation Protocol (RSVP) —** To classify, control, and prioritize traffic where available bandwidth is low and your network is carrying time-sensitive or business-critical information, use the QoS and RSVP features. For more information, see Chapter 17.

8 Monitor the system and analyze network activity.

You can use the system's device monitoring features such as event logging, baselining, roving analysis, and RMON to record and analyze your network periodically and identify potential network problems before they become serious problems. To test and validate paths in your network, use tools like ping and traceRoute. SNMP and MIBs provide ways to collect performance data on your network. For more information on these features, see Chapter 18.

2 MANAGEMENT ACCESS

This chapter explains the different methods used to configure management access to the system. It describes the different types of applications and the underlying communication protocols that are used to deliver data between your end-station device and the system. It also contains information about connecting to the system directly through one of two serial connections, or through an Ethernet port to an IP (Internet Protocol) interface to run network management applications.

This chapter covers the following topics:

- Management Access Overview
- Key Concepts
- Key Guidelines for Implementation
- Administration Console Access
- Web Management Access
- SNMP Access

Management Access Overview

The system provides you with the flexibility to access and manage your system using several different methods. You can administer your system using:

- The Administration Console
- Web Management suite of applications
- An external SNMP-based network management application such as 3Com's Transcend Network Control Services

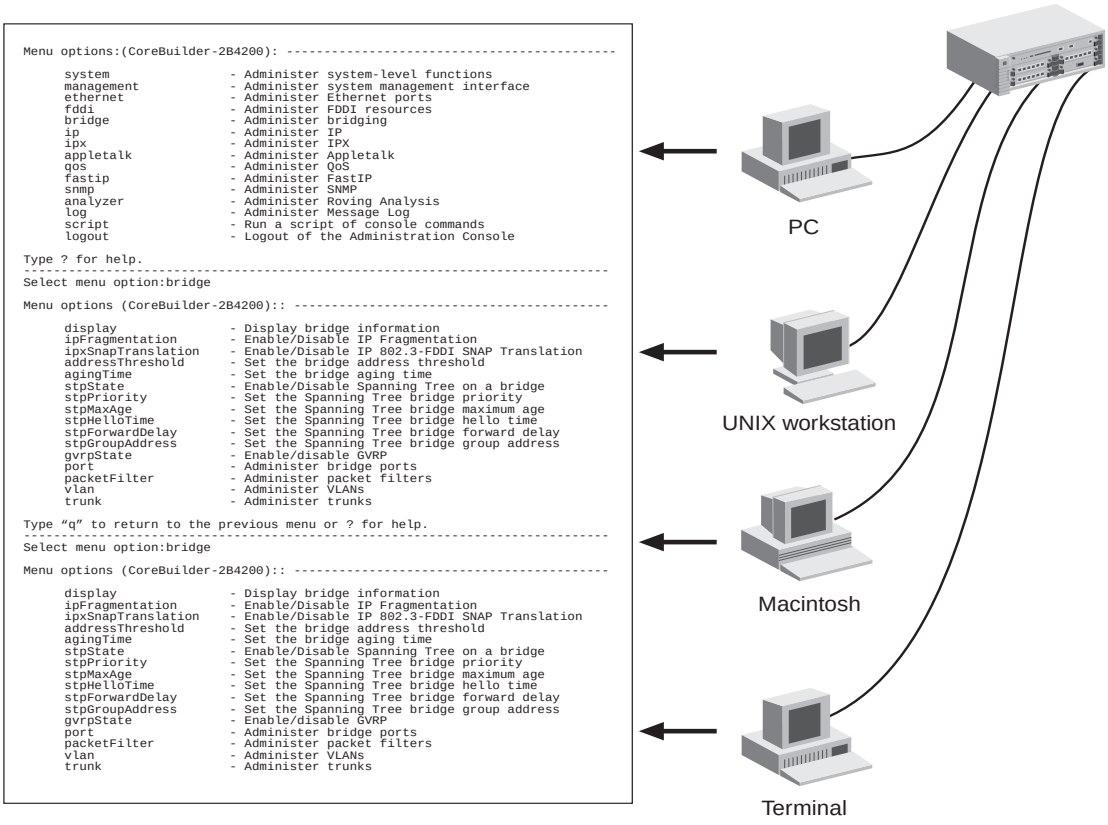
The Administration Console and most of Web Management are embedded parts of the software and are available for immediate use on your system.

Administration Console Overview

The Administration Console is an internal character-oriented, menu-driven, user interface for performing system administration such as displaying statistics or changing option settings. You can view the Administration Console from a terminal, a PC, a Macintosh, or from a UNIX workstation. You can access the Administration Console through a terminal or modem serial port, or through an Ethernet port using an Internet Protocol (IP) interface.

Figure 1 shows a sample output of menu options that can be viewed from the various devices.

Figure 1 Viewing the Administration Console



Web Management Overview

The Web Management software consists of embedded Web Management applications and installable tools:

- **Embedded Web Management applications** — Use the embedded Web Management applications for most of your device configuration and management tasks. You can manage a single port or device, or, using multiple windows, you can manage multiple devices. This software, which is part of the system software image, contains:
  - **WebConsole** — An HTML-based set of configuration forms.
  - **DeviceView** — A Java-based application that displays a real-time image of the device. You can manage each port, module, or system by clicking the part of the image that you want to manage.
  - **Performance features** — Dynamic monitoring through graphing of QoS statistics and Ethernet interfaces.
  - **Help** — Access to the configuration form on which you set up the installable Help files as well as access to links to support information on the 3Com Web site.
- **Installable tools** — Install these optional tools on your workstation from the *Software and Documentation CD-ROM* or from the 3Com Web site:
  - **DeviceView accessories** — To set up e-mail notification for Status Logging
  - **WebManage Framework** — To group your access links to the devices that you manage
  - **Filter Builder** — To create and test filters for packets on your switch
  - **Form-specific Help** — To get more information about WebConsole, DeviceView, and Performance forms

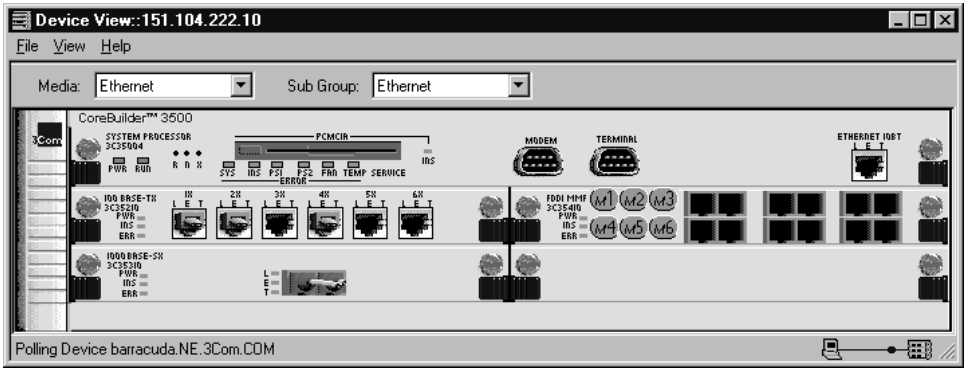
For details about this software, see the *Web Management User Guide.*

SNMP-Based Network Management Overview

For more complete network management, you can use an external SNMP-based application such as 3Com's Transcend Network Control Services or another network management application. You access external applications through an Ethernet port using an IP interface.

Figure 2 shows an example of a Device View screen.

Figure 2 Sample Transcend Network Control Services Device View



Key Concepts

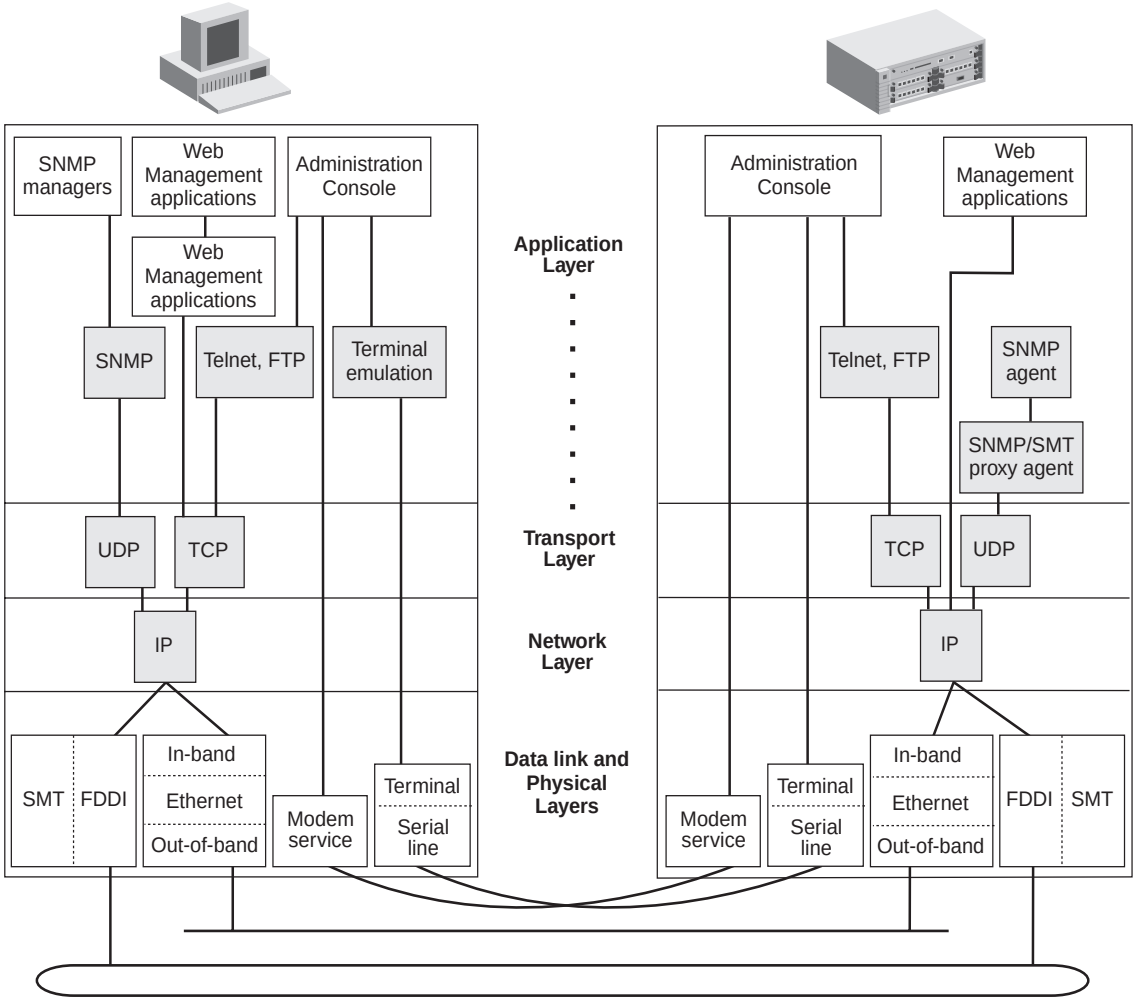
This section describes the relationship between the methods of management access described in the previous sections and how they fit into established networking protocols. It also introduces the concepts of in-band and out-of-band management using IP.

OSI Protocols

Management and administration on the system occur through the layers of the Open Systems Interconnection (OSI) reference model.

Figure 3 shows how the different management access methods fit into the OSI model.

Figure 3 OSI Protocols for the CoreBuilder 3500



Protocols The system supports the following protocols:

- Virtual terminal protocols, such as Telnet
- Simple Network Management Protocol (SNMP)
- FDDI Station Management (SMT) protocol

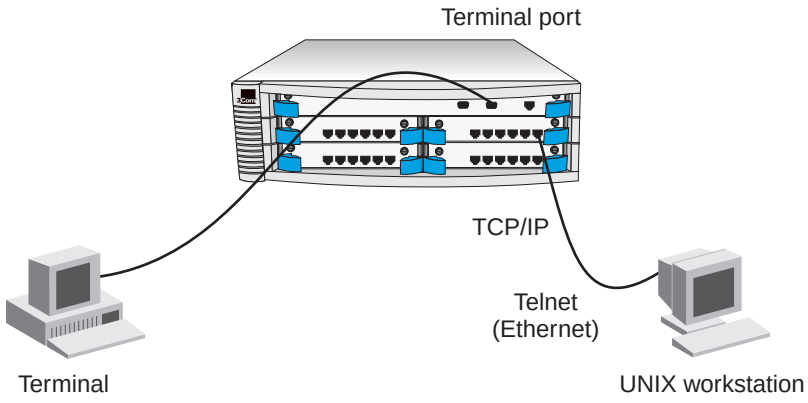
Virtual Terminal Protocols

A *virtual terminal protocol* is a software program, such as Telnet, that allow you to establish a management session from a Macintosh, a PC, or a UNIX workstation. Because Telnet runs over TCP/IP, you must have at least one IP address configured on the system before you can establish access to it with a virtual terminal protocol. Within the Administration Console, you configure an IP address by defining an IP interface. See the *Command Reference Guide* for additional information about defining IP addresses for in-band or out-of-band management.

Terminal emulation differs from a virtual terminal protocol in that you must connect a terminal directly to the serial port.

Figure 4 shows a UNIX workstation connected to the system through a virtual terminal protocol, and a terminal connecting directly to a serial port through a null modem cable.

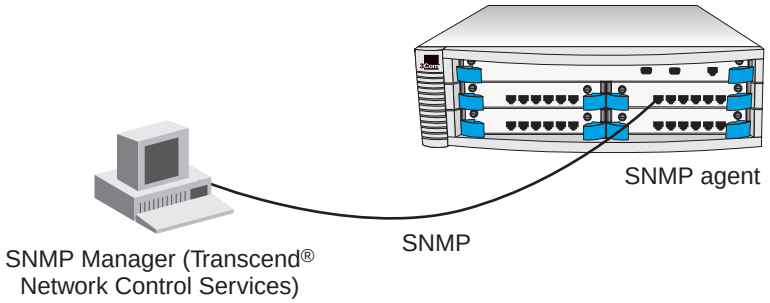
Figure 4 Administration Console Access



Simple Network Management Protocol

Simple Network Management Protocol (SNMP) is the standard management protocol for multi-vendor IP networks. SNMP supports transaction-based queries that allow the protocol to format messages and to transmit information between reporting devices and data-collection programs. SNMP runs on top of the User Datagram Protocol (UDP), offering a connectionless-mode service. Figure 5 shows a PC connected to the system through an Ethernet port.

Figure 5 SNMP Manager Access



See Chapter 18 for additional information about SNMP.

IP Management Concepts

Both in-band and out-of-band management have advantages and disadvantages.

In-Band Management If you manage your system and its attached LANs over the same network that carries your regular data traffic, then you are managing your network *in band*. This kind of management is often the most convenient and inexpensive way to access your system. The disadvantage is that, if your data network is faulty or congested, you may not be able to diagnose the problem because management requests are sent over the same network.

Out-of-Band Management If you are using a dedicated network for management data, then you are managing your network *out of band*. Although this is a more expensive way to access your network, you are able to diagnose problems even when your data network is faulty.

Key Guidelines for Implementation

This section describes guidelines for the different ways to access your system.

Access Methods

There are several ways you can access your management application on the system; locally through a terminal connection, or remotely using a modem or an IP connection.

Table 3 describes these different methods.

Table 3 Management Access Methods

Access Method	Access Description	Interface
Terminal	Connect directly to the Administration Console and stay attached during system reboots.	Terminal serial port (see "Terminal Port Access").
Modem	Access the Administration Console by dialing in from remote sites.	Modem serial port (see "Modem Port Access").
IP	Access the Administration Console with up to four Telnet sessions. Use Web Management or an external SNMP management application to communicate with the CoreBuilder SNMP agent.	In-band or out-of-band Ethernet port assigned to an IP interface (see "In-Band Management" and "Out-of-Band Management").

Setting Up the Terminal Port

Use the Administration Console to set the baud rate to match the speed of your terminal.

Baud setting changes take effect immediately after you confirm the change. You must adjust the baud setting of your terminal or terminal emulator to match your management interface port before you can reestablish communication using the terminal port. When you change the baud rate to something other than 9600, the new setting becomes the new default, even after you issue a `system nvdata reset` *command.*

You can use the `system serialPort terminalSpeed` *command through the terminal serial port or through an IP interface. However, if you change the terminal speed while in a telnet session, you must reboot the system for the change to take effect.*

Setting Up the Modem Port

Use the Administration Console to match your external modem speed. Then configure the external modem by establishing a connection between your current Administration Console session and the modem port.

You must establish a connection to the modem by issuing the `system serialPort connectModem` *command after you change the modem speed and before dialing in. This sequence allows the modem to synchronize its baud rate with the system.*

See the *CoreBuilder 3500 Getting Started Guide* for terminal port and modem port pin-outs. For additional information about modem port settings, see the *Command Reference Guide*.

IP Management Interface

An Internet Protocol (IP) management interface allows you to manage the system in-band through an Ethernet port on a module, or out-of-band through the out-of-band Ethernet port. You can access the system through an IP interface in one of the following ways:

- You can use Telnet to connect up to four concurrent remote sessions to the Administration Console using a terminal program from a host computer.
- You can run Web Management to access its management applications to manage and monitor your system.
- You can run an SNMP-based network management application to manage and monitor your system.

IP is a standard networking protocol that is used for communications among various networking devices. To gain access to the system using TCP/IP or to manage the system using SNMP, you must set up an IP interface for your system. How you set up the IP interface depends on whether you plan to manage the system in band (with your regular network traffic) or out of band (with a dedicated network).

For Telnet, Web Management, or SNMP access, you must first define an IP interface. You can use either an out-of-band or in-band port for the IP interface, but do not assign the same IP address to both the out-of-band and in-band ports. Also, be sure not to assign an out-of-band port IP address that is on the same subnet as any of the in-band IP interfaces.

- **In-Band Management** — If you are managing your network in-band, you need to set up an IP routing interface and at least one VLAN. See Chapter 9 for information about defining a VLAN, and Chapter 11 for information about setting up an IP routing interface. See "In-Band Management" for additional information about in-band management.
- **Out-of-Band Management** — If you are managing your system out of band, you need to assign an IP address and subnet mask for the out-of-band Ethernet port on your system through the `management` menu. The out-of-band Ethernet port is the 10BASE-T port on the system processor module and is not associated with a port number. See Chapter 11 for background information on IP addresses and subnet masks. See "Out-of-Band Management" for additional information about out-of-band management.

Administration Console Access

The first time that you access the Administration Console, access the system at the *administer* level and press the Return key at the password prompt. The initial password is null. Subsequent access is described next.

Password Levels

The Administration Console supports three password levels, allowing the network administrator to provide different levels of access for a range of users, as described in Table 4.

Table 4 Password Access Levels

Access Level	For Users Who Need to	Allows Users to
Administer	Perform system setup and management tasks (usually a single network administrator)	Perform system-level administration (such as setting passwords, loading new software, and so on)
Write	Perform active network management	Configure network parameters (such as setting bridge aging time)
Read	Only view system parameters	Access only *display* menu items (display, summary, detail)

Passwords are stored in nonvolatile (NV) memory. You must enter the password correctly before you can continue.

When you access the Administration Console, the top-level menu appears. You manage and monitor your system by selecting options from this menu and from others below it. Each menu option is accompanied by a brief description.

For additional information about using the Administration Console, see the *Command Reference Guide*.

Terminal Port Access

Direct access to the Administration Console through the terminal serial port is often preferred because you can remain on the system and monitor it during system reboots, and certain error messages are sent to the serial port, regardless of the interface through which the associated action was initiated. A Macintosh or PC attachment can use any terminal emulation program for connecting to the terminal serial port. A workstation attachment under UNIX can use an emulator such as TIP.

Modem Port Access

You can access the Administration Console from your PC or Macintosh using an external modem attached to the modem serial port.

When you have configured the external modem from the Administration Console, the system transmits characters that you have entered as output on the modem port. The system echoes characters that it receives as input on the modem port to the current Administration Console session. The console appears to be directly connected to the external modem.

Web Management Access

Web Management applications are an embedded part of the CoreBuilder 3500 Enterprise Switch. They include WebConsole, DeviceView, and Performance monitoring tools. Additional installable applications include Help.

After you have set up your IP address for the CoreBuilder 3500 system, you can access Web Management applications directly in your Web browser by entering the IP address of the system.

In the installable WebManage Framework window, you can list and manage all your devices from one central location. You can easily add and delete devices and group the devices in ways that make sense to you, for example, by location or subnetwork.

For more information, see the *Web Management User Guide*.

Browser Requirements

Web Management requires either Microsoft Internet Explorer 4.01 or later, or Netscape Navigator 4.03 or later.

- **Netscape Navigator** — If you are using Netscape Navigator 4.03 or 4.04, be sure to install the Netscape JDK 1.1 Patch. You can download the patch from the following location:

  `http://help.netscape.com/filelib.html#smartupdate`

  If you encounter problems accessing Help files when you use Netscape, clear the browser memory cache and disk cache and restart the browser.

- **Internet Explorer** — If you are using Internet Explorer, install the latest 4.01 Service Pack 1. This service pack makes Internet Explorer Year 2000 compliant and fixes other product support issues. You can download the 4.01 Service Pack 1 from the following location:

  `http://www.microsoft.com/msdownload/iebuild/ie4sp1_win32/en/ie4sp1_win32.htm`

  If the above link is unavailable, you can download the service pack from the Microsoft home page:

  `http://www.microsoft.com`

See the *Web Management User Guide* for additional information about Web Management.

SNMP Access

You can use an external SNMP-based application such as 3Com Transcend Network Control Services to access your system through an Ethernet port using an IP interface. SmartAgent® intelligent agents are the foundation of the Transcend architecture. SmartAgent software and RMON work together to provide automatic network-wide monitoring, analysis, and reporting. For additional information about Transcend Network Control Services, see the 3Com Web page at:

http://www.3com.com

3 System Parameters

This chapter guidelines and other information about the system parameters that you can configure.

This chapter covers these topics:

- System Parameters Overview
- Key Concepts
- Key Guidelines for Implementation
- File Transfer
- Security
- Software Update
- nvData Operations
- Simple Network Time Protocol (SNTP)
- Standards, Protocols, and Related Reading

You can manage system parameters in either of these ways:

- *From the* `system` *menu on the Administration Console. See the* Command Reference Guide.
- *From the System folder of the Web Management software. See the* Web Management User Guide.

System Parameters Overview

On the Administration Console, you use the `system` menu to set or modify values for system parameters or functions. For many of these parameters, you can also use the configuration forms in the System folder of the Web Management suite of software applications.

Features

You can set or modify the values for when you perform the following tasks:

- Display your system's current configuration
- Take a snapshot of your system's current system configuration and status
- Create and modify passwords
- Create and maintain a statistics baseline

  See Chapter 18 for details.
- Set and administer your system's serial port baud rates

  See Chapter 2 for details.
- Modify your system's date and time

See the *Command Reference Guide* for descriptions of the commands that you use to set and modify these system parameters.

You can also set options for the following, as discussed in these sections later in this chapter:

- File Transfer
- Security
- Software Update
- nvData Operations
- Simple Network Time Protocol (SNTP)

Benefits Using the options on the `system` menu:

- Provides an easy method for setting and modifying system parameters.
- Provides added security by limiting IP and Web Management access to your system.
- Decreases the time and cost of modifying your system configuration. You do not need to make frequent changes from the same source and then reboot your system to apply the changes.
- Provides an easy method of communicating remotely through the fileTransfer option.
- Reduces the cost of software upgrades by providing an easier process for remote upgrade operations.
- Provides an easy method for changing your system time, changing time zones, and resetting for daylight savings time through SNTP.

Key Concepts

Review these terms and key concepts for system parameters:

- **FTP** — File Transfer Protocol. You can send files from one system to another with this protocol.
- **TFTP** — Trivial File Transfer Protocol. Designed to function over the User Datagram Protocol (UDP), this protocol reads and writes files to and from a remote server. It is smaller and easier to operate than FTP, but it lacks most of the FTP features.
- **Save** — Use this option on the `nvData` menu to save nvData to a file on a remote system.
- **Restore** — Use this option on the `nvData` menu to restore data from a file on a network host.
- **Examine** — Use this option on the `nvData` menu to examine a previously saved nvData file header.
- **Simple Network Timing Protocol (SNTP)** — SNTP is an adaptation of the Network Time Protocol (NTP). NTP is used to synchronize computer clocks in the global Internet. For more detailed information on NTP and how it is used in your system, see "Simple Network Time Protocol (SNTP)" later in this chapter.
- **Trusted IP Client** — One or more clients that you can allow to have management access to your system. You can configure up to 5 IP addresses or 5 subnetworks on this access list.

Key Guidelines for Implementation

This section briefly explains how to set and modify the values for system parameters that you can set.

The system sets most of the parameter values during power-on. To set parameters that are not defined by the system or to modify predefined values, use one of the following methods:

- The `system` menu on the Administration Console's top-level menu
- The System folder of Web Management software

To set or modify system parameter values, follow these basic steps:

1. Access the menu or form that governs a system parameter.
2. Specify a value.

File Transfer

From the system menu or folder, you can select which protocol you want the system to use to transfer data between systems. Choose either File Transfer Protocol (FTP) or the Trivial File Transfer Protocol (TFTP), which is the default.

Implementing FTP

FTP meets the following file transfer objectives:

- Transfers data reliably and efficiently through an IP connection
- Provides security by ensuring that the person who attempts to use FTP has a valid username and password combination

Important Consideration

- All file transfers using FTP are sent over an IP connection. Before you use FTP, you must configure an IP address for the system. For more information on IP, see Chapter 11.

Implementing TFTP

The Trivial File Transfer Protocol (TFTP) is simpler to use than FTP but has less functionality. TFTP uses UDP as its transport protocol, with a simple stop-and-wait acknowledgment system. Because TFTP has an effective window of only one 512-octet segment, its performance cannot match that of FTP. The most common application for TFTP is bootstrapping a host over a local network.

Important Considerations

Consider the following guidelines before you select TFTP:

- TFTP does not provide access control or security, so use TFTP only when authentication and directory visibility are not required.
- Because TFTP provides no user authentication, you must give *loose* permission to files that are located on your system, that is, make files publicly readable and writable. Otherwise, the TFTP server does not grant requests for file access.
- You must create two files when you are using the save nvData option over TFTP. See "Saving nvData" in this chapter.

For more information on TFTP, see your TFTP server documentation.

Security

You can limit IP management access to your system through the Administration Console or the Web Management software as follows:

- On the Administration Console, you can limit IP management access through the `system console security` menu.
- On the Web Management software, use a security option in the WebManage folder on the Web console.

To limit IP management access, you can use the `system console security` option to configure up to 5 IP addresses or 5 subnetworks, called *trusted IP clients*. If an IP address or subnet is not on the trusted IP client list, the IP address or subnet cannot be used to access the system using the Web Management software, Administration Console, or SNMP.

If you do not configure trusted IP clients on the system, a user with the appropriate password at a remote device can access the system.

Security Options

To configure trusted IP clients from the Administration Console, use the following options:

- **Display** — Shows the IP address and subnet mask of each trusted IP client.
- **Define** — Allows you to supply the IP address and subnet mask of a trusted IP client.
- **Remove** — Removes an IP client from the trusted list.
- **Message** — Controls the message that is displayed when access is denied.
- **Access** — Enables or disables checking for trusted IP clients. By default, checking for trusted IP clients is disabled.

The Web Management software offers these security options:

- **Display** — Displays the trusted IP clients and indicates whether checking for trusted IP clients is enabled or disabled.
- **Configuration** — Allows you to enable or disable checking for trusted IP clients and control the message displayed to a user when access is denied.
- **Add Trusted Client** — Defines a trusted IP client.
- **Remove Trusted Client** — Removes a trusted IP client from the list.

Important Considerations

Consider the following guidelines *before* you configure trusted IP clients on your system.

Procedures

Configure trusted IP clients in this order:

1. Define the trusted IP clients.
2. Display the list of configured trusted IP clients to verify that you have configured them correctly.
3. Enable the checking for trusted IP clients (using the access option on the Administration Console or the System Configuration form in the Web Management software).

CAUTION: *Be careful when you define trusted IP clients. If you specify an incorrect IP address or subnetwork, you can affect your ability to access the system, as follows:*

- *For Web Management access, the change is immediate. Therefore, an incorrect IP address or subnet forces you to reestablish local access via the serial port.*
- *For Telnet access, the change takes effect at your next login.*

Additional considerations

- If you modify a trusted IP client definition through the Web Management software, the change also affects Telnet and SNMP access to the system. If you modify a trusted IP client definition through Telnet access to the Administration Console, the change also affects SNMP and Web Management access to the system.
- Use the subnet mask to allow all addresses on a particular subnetwork to have trusted access. For example, the IP address 158.101.112.219 with a subnet mask of 255.255.255.0 allows all addresses on the 158.101.112 subnetwork to have trusted access, whereas the same IP address with a subnet mask of 255.255.255.255 only allows only access by 158.101.112.219.
- The trusted IP client information is retained, that is, saved in nvData after a system reboot.

Software Update

You can load a new or updated version of the system software into your system's flash memory or to a PCMCIA flash memory card, with softwareUpdate option on the System menu through the Administration Console. Depending on your network load, loading software into flash memory can take approximately 10 to 15 minutes to complete.

Important Considerations

Consider the following guidelines *before* you update the system software:

- Before you attempt to install the system software, verify that you have extended memory installed on your system. For information on how to verify your system's memory see the *Getting Started Guide*.
- You can load the system software into flash memory while the system is operating. The system does not have to be powered off.
- Verify that you have defined an IP address on your system.
- To guard against failure during the software upgrade, be sure to save the software to nvData *before* you perform the system software upgrade.

Consider the following points *after* you upgrade the system software:

- If the executable software image that is stored in flash memory is corrupted (for example, if a power failure occurs during the update), contact 3Com Technical Support.
- You can continue to run the old software after you perform a system software upgrade. When it is convenient, reboot your system to use the upgraded software.

nvData Operations

All of the system's configurable parameters are saved in nonvolatile memory. When you work with nonvolatile data (nvData), you can:

- Save and restore your system configuration for backup.
- Examine a saved nvData file header.
- Reset system data to its factory default values, if necessary.

Saving nvData

When you enter commands to save nvData, the system copies data that is stored in nonvolatile memory to a disk file location that you specify. You can use the `system nvdata save` option to save nvData from your system to a:

- File on another system remotely through FTP or TFTP.
- PCMCIA flash card indirectly.

See the PCMCIA Flash Card User Guide *for a detailed explanation about how to save nvData to a PCMCIA flash memory card.*

Important Considerations

Consider the following guidelines before you perform an `nvData save` operation:

- When you use TFTP, *before* you save data to the file, you have to create two files on the TFTP server. The screen display appears as follows:

```
Select menu option: system nvdata save
Host IP Address [158.101.100.1]: 158.101.112.34
NV Control file (full pathname): [/tftpboot/mecca]
Enter an optional file label {?}: mecca2
Control File:mecca
Data File: mecca.nvd
Saving system ...
```

- You must supply the host IP address and specify the file where you want to save the data according to requirements of your TFTP and FTP implementation.
- Some TFTP implementations require that you store the file in the same directory where the TFTP daemon (server) is running on a remote host.
- Because TFTP does not provide user authentication, give the file loose permissions to make it both readable and writable. TFTP does not grant requests for file access.

Restoring nvData

Use the nvData restore option on the `system nvData` menu to restore a previous configuration that you have saved to an external file.

Effects and Consequences

Consider the following guidelines before you restore nvData:

- Do not confuse nvData restore with nvData reset. You use `nvData reset` *only* to reset your system configuration values to their factory default settings.
- After you restore nvData, the software presents a proposal for how to restore the data based on the following restoration rules:

Rule 1

- **Exact match** — The system IDs and revisions (if applicable) all match between the saved configuration and the configuration of the system on which you are restoring the image.

Rule 2

- **System ID mismatch** — System IDs do not match between the saved configuration and the target system. In this case, the system informs you of the mismatch and then prompts you to continue.

If neither of these rules succeeds, you cannot apply the saved configuration to your system.

- Before you restore a system with mismatched system IDs, consider the following issues that might cause problems after the nvData is restored:
  - Management IP addresses (which are defined in IP interface configurations) are saved as nvData and restored. Restoring management IP addresses can cause duplicate IP address problems. To avoid these problems, change the IP addresses of any defined interfaces before you connect the restored system to the network.
  - Statically configured MAC addresses are saved as nvData. After a successful restore operation, verify that you have no duplicate addresses.

Resetting nvData

To reset the system settings back to their factory default values, use the nvData reset option.

Important Considerations

Consider these points *before* you reset nvData on your system:

- Resetting nvData erases all user-configured data, including all passwords, *except* the terminalSpeed and modemSpeed baud settings and the system boot parameters. Therefore, before you reset all affected values, document your configuration so that you can reconfigure the system after you reset it, or save the existing nvData to a file. See "Saving nvData" earlier in this chapter for details.
- You can reset nvData on a system only when it is directly connected through the Administration Console. You cannot reset nvData through a Telnet connection.

Viewing nvData

To verify that you have successfully saved nvData to the file that you specified, view the header information for that file. The header information shows pertinent product and system information.

Example:

```
Select menu option: system nvdata examine
Host IP Address [158.101.100.1]: 158.101.112.34
NV Control file (full pathname): systemdata
Product ID 4, Product Type 1
System ID 102D00
Saved 1999-05-20T09:24:43 AM Version 2.
Labelled: LabSwitch
```

Simple Network Time Protocol (SNTP)

This section covers:

- SNTP Overview
- Implementing SNTP

SNTP Overview

SNTP is an adaptation of the Network Time Protocol (NTP), which is used to synchronize computer clocks in the global Internet. NTP provides comprehensive mechanisms to access national time and frequency dissemination services, organize the time-synchronization subnetwork, and adjust the local clock in each participating subnetwork peer.

SNTP is a simplified access strategy for servers and clients using NTP version 3. The access paradigm is identical to the User Datagram Protocol (UDP)/TIME protocol, so it is relatively easy to adapt a UDP/TIME client implementation to operate using SNTP. SNTP is designed to operate in a dedicated server configuration with existing NTP and other SNTP clients and servers.

SNTP can operate in either unicast mode (point-to-point), multicast mode (point-to-multipoint), or anycast mode (multipoint-to-point):

- **A unicast client** — Sends a request to a designated server at its unicast address and expects a reply within a specified time frame. From the reply, the unicast client can determine the time and (optional) the round-trip delay and local clock offset relative to the responding server
- **A multicast server** — Periodically sends a unsolicited message to a designated IP local broadcast address or multicast group address and expects no requests from clients.
- **An anycast client** — Sends a request to a designated IP local broadcast address or multicast group address. One or more anycast servers reply with their individual unicast addresses. The anycast client binds to the first reply that it receives and then continues the operation in unicast mode.

Implementing SNTP

The system software provides an SNTP client, which works with distributed SNTP time servers to synchronize the system clock to international time standards.

The SNTP client operates in unicast mode, which means that the client and server end-system addresses are assigned following the usual IP conventions. Although SNTP in these systems supports one server at a time, you can define up to three servers for backup. Therefore, when the client does not receive a response from the first server within a designated time, it sends a request to the next server on the list.

Standards, Protocols, and Related Reading

See the following references for more information on these protocols:

- **RFC 959** — File Transfer Protocol Specification
- **RFC 1350** — Trivial File Transfer Protocol Specification
- **RFC 2030** — Simple Network Time Protocol, v4.0, Specification
- **RFC 1305** — Network Time Protocol, v3.0, Specification
- **RFC 868** — Time Protocol Specification

4 PHYSICAL PORT NUMBERING

The CoreBuilder® 3500 follows a specific set of rules for assigning physical port numbers. This chapter describes the physical port numbering on the system. It covers the following information:

- Port Numbering Overview
- Key Guidelines for Implementation
- Examples of Port Numbering
- Effects of Removing a Module
- Effects of Replacing Modules

Port Numbering Overview

Before you configure your system, read this chapter to become familiar with the physical port numbering scheme on the system. Understanding the port numbering scheme enables you to:

- Manage your bridge ports, especially if you use trunking, as described in Chapter 8.
- Accurately define your virtual LANs (VLANs), as described in Chapter 9.

Numbering Rules

Your system supports up to 24 ports, numbered consecutively:

- Port 1 represents the first port associated with a module in Slot 1 (the top left slot on the system) and continues for the rest of the ports associated with Slot 1.
- Numbering continues for the ports associated with a module in Slot 2 (top right).
- Numbering continues for the ports associated with a module in Slot 3 (bottom left).
- Numbering continues for the ports associated with a module in Slot 4 (bottom right).

See Figure 6 later in this chapter for an example.

Additional rules:

- Port numbering is consecutive, regardless of module type.
- Numbering skips over an empty slot and continues with the ports associated with the next occupied slot.
- Numbering includes unused ports.

For several examples of port numbering, see "Examples of Port Numbering" later in this chapter.

Supported Module Types

The port numbering range depends on the type of modules that you have configured into your system. For example, at Release 2.0.0, the system supported the following modules:

- Up to four 10/100BASE-TX Ethernet modules, each with 6 ports that have RJ-45 connectors
- Up to four 100BASE-FX Ethernet modules, each with 6 ports that have SC connectors
- Up to four 1000BASE-SX or 1000BASE Gigabit Interface Converter (GBIC) Ethernet modules, each with 1 port (up to four Gigabit Ethernet ports per system). The 1000BASE GBIC module requires CoreBuilder 3500 system software at release 1.2.0 or higher. Each Gigabit Ethernet module uses a trunk resource, so keep track of your trunk resources (maximum of 4) when you add a Gigabit Ethernet module. See Chapter 8 for information on trunking and trunking resources.
- Up to four FDDI modules, each with 6 ports
- Any combination of these modules. For example, you can have one Gigabit Ethernet module in Slot 1 (port 1), one FDDI module in Slot 2 (ports 2–7), one 10/100BASE-TX Ethernet module in Slot 3 (ports 8–13), and one 100BASE-FX Ethernet module in Slot 4 (ports 14–19).

Key Guidelines for Implementation

To ensure that you understand the port numbering that the system reports for certain aspects of your configuration (bridging information, trunks, FDDI ports, and VLANs), observe these guidelines when you configure your system:

- Determine your physical port configuration before you attempt to configure any bridging parameters.

Trunking

- If you use *trunking* to group ports, configure your trunks *before* you attempt to configure any VLANs. Be sure that you understand how trunking associates a group of ports with a trunk. (See Chapter 8.) These associations affect the following situations:
  - When you perform an operation for which you must specify bridge ports (for example, when you define VLANs), you must use the lowest-numbered port in each trunk to represent the trunk. The operation that you perform then applies to all ports in the trunk.
  - When you view information that applies to more than one port (for example, bridging displays for trunks), the port number field identifies all ports in the trunk. A VLAN summary display lists all physical ports to indicate which physical system connectors can receive or transmit frames within a VLAN. (You must use the VLAN detail display to see trunk port groups.)

FDDI DAS pairs

- By default, FDDI ports are single-attached station (SAS) M-ports, where each port is selectable as a bridge port. If you configure FDDI ports as *dual-attach station (DAS) pairs*, you associate two FDDI ports with each DAS pair and only the lowest-numbered port in the pair is selectable as a bridge port. Configure the appropriate number of DAS pairs *before* you configure any VLANs. Be sure that you understand how a DAS configuration associates the two FDDI ports in the pair. (See Chapter 6.) These associations affect the following situations:
  - When you perform an operation for which you must specify bridge ports (for example, when you define VLANs), you must use the lowest-numbered port in each DAS pair to represent the DAS pair. The operation that you perform then applies to both ports in the DAS pair.
  - When you view information that applies to more than one port (for example, bridging displays for DAS pairs), the port number field identifies both ports in the DAS pair. A VLAN summary display lists all physical ports to indicate which physical system connectors can receive or transmit frames within a VLAN. (You must use the VLAN detail display to see the DAS pairs.)

The configuration of trunks or DAS pairs does not change the port numbering scheme shown in displays such as Ethernet statistics displays or bridge port displays. If you have created trunks or FDDI DAS pairs, however, be aware that a group of ports is associated with each trunk or DAS pair. Therefore, a display such as a bridge port display groups the ports associated with each trunk or DAS pair. See the Command Reference Guide *for examples of the bridge port display commands.*

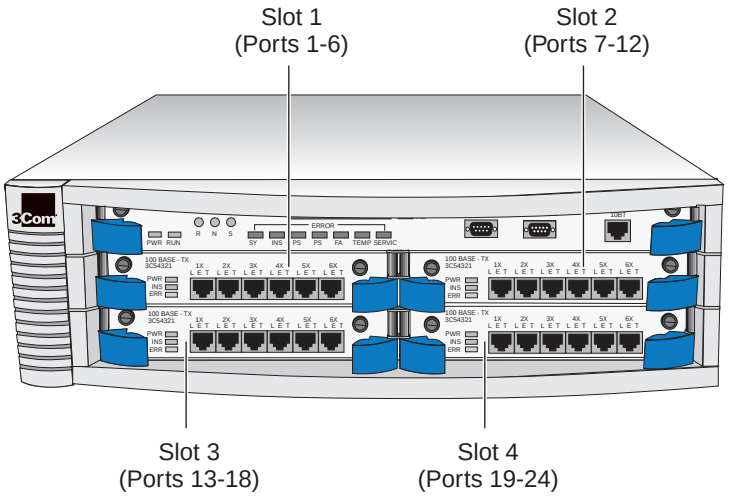
Examples of Port Numbering

This section provides sample configurations that illustrate port numbering on the system.

Example 1: Fully Loaded System

For a fully loaded system (4 occupied slots) with Fast Ethernet ports, the ports are numbered 1 through 24, starting top left to top right, and then continuing bottom left to bottom right, as shown in Figure 6. (The figure shows the 10/100BASE-TX module.)

Figure 6 Port Numbering for a System with Four Fast Ethernet Modules

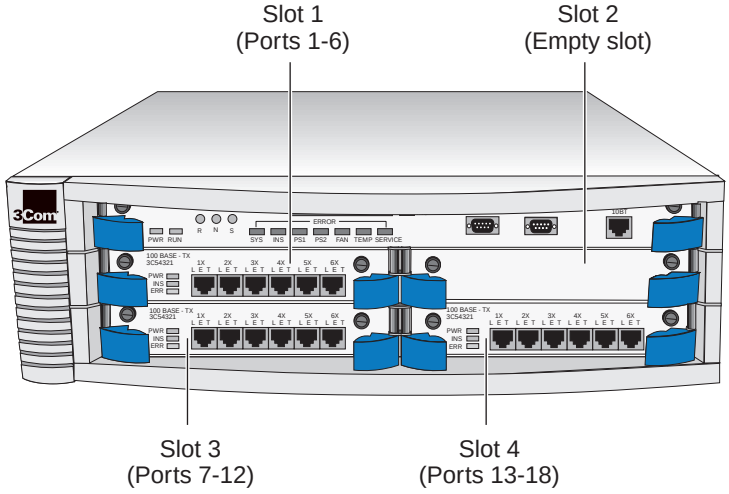


Example 2: Empty Slot in the System

When you have an empty slot, the port numbering includes no ports for that slot. With three Fast Ethernet modules, for example, you have 18 ports, which are numbered according to their position in the system.

For example, if the top-right slot is empty (slot 2), the ports are numbered as shown in Figure 7. (The figure shows the 10/100BASE-TX module.)

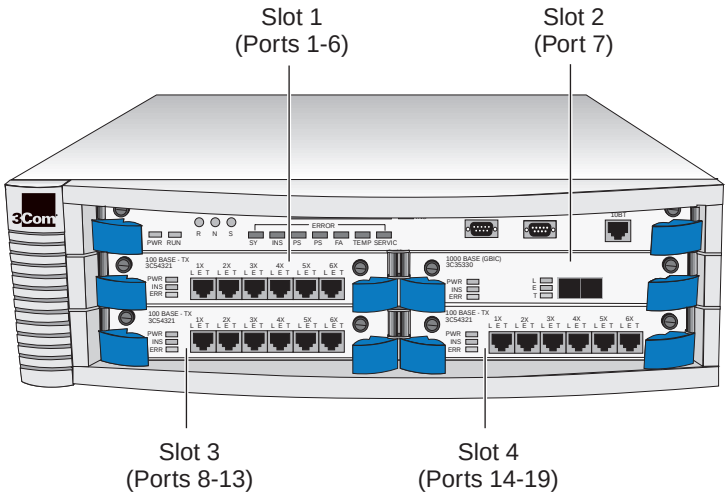
Figure 7 Port Numbering for a System with an Empty Slot



Example 3: Gigabit Ethernet Module with Other Modules

When you have a system with one Gigabit Ethernet module and three Fast Ethernet modules, port numbering accounts for the single port on the Gigabit Ethernet module, as shown in Figure 8.

Figure 8 Port Numbering for a System with a Gigabit Ethernet Module



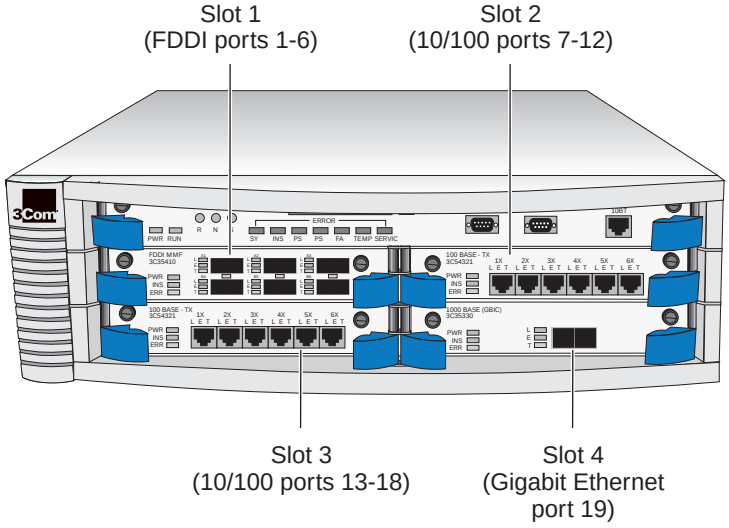
Example 4: FDDI Module with Other Modules

An FDDI module has six FDDI ports (two rows of three ports). Figure 9 shows an FDDI module in slot 1. The top row's ports are numbered 1 through 3 and the bottom row's ports are numbered 4 through 6. Slots 2 and 3 have 10/100 Fast Ethernet modules, and Slot 4 has a Gigabit Ethernet module.

When two FDDI ports are configured as a dual-attach station (DAS) pair, there is one bridge port using two physical (fiber) connectors. The anchor port is the A-port of the DAS port pair. If you configure two FDDI ports as a DAS pair, you must specify the lowest-numbered (anchor) port in the DAS pair and the other port in the pair becomes unselectable.

For example, for the FDDI module shown in slot 1, the three configurable DAS pairs have ports 1 and 4, ports 2 and 5, and ports 3 and 6. When specifying bridge ports (for example, for VLANs), you specify port 1 to represent the first DAS pair, port 2 to represent the second DAS pair, and port 3 to represent the third DAS pair. For more information about FDDI configurations, see Chapter 6.

Figure 9 Port Numbering for a System with an FDDI Module



Effects of Removing a Module

When you remove a module and leave the slot empty, a number of changes occur.

Port-Numbering Changes

The ports are sequentially renumbered when you remove a module from slot 1, 2, or 3. Removing a module in slot 4 does not cause renumbering, only a loss of those ports.

Example If you have a fully loaded system with four 10/100BASE-TX modules and you remove the module in slot 3 (ports 13-18), the ports associated with the module in slot 4 (formerly numbered 19-24) are renumbered to 13-18. (See Figure 6.)

VLAN Changes

When you remove a module, VLAN changes occur as follows:

- If you have a VLAN that contains ports that have been renumbered, the renumbered ports now appear in the VLAN summary display.

  Example If a VLAN contained ports 20 through 22 before you removed the module in slot 3, these ports show up as ports 14 through 16 in the VLAN summary after you remove the module.

- If you have a VLAN that includes ports associated with the removed module, those ports are removed from the VLAN and the VLAN summary display no longer shows those ports. (This change includes trunk ports.)

  Example If a VLAN contained ports 17 through 24 before you removed the module in slot 3, the removal of the module in slot 3 causes the removal of previous ports 17 and 18 from the VLAN. (See Figure 6.) The VLAN then contains the renumbered ports 13 through 18 (previously ports 19-24).

- If there are no remaining ports in the VLAN once you remove the module, the VLAN summary display shows the VLAN without any ports.

See Chapter 9 for more information about VLANs.

Trunk Changes When you remove a module, trunk changes occur as follows:

- If you have a trunk that includes ports associated with the removed module, the trunk display shows that the trunk has `Missing` ports.

Example If you had a trunk on ports 17 through 20 before you removed the module in slot 3, the removal of that module causes the trunk to have two missing ports (17 and 18). (See Figure 6.) It now has renumbered ports 13 and 14 (previously ports 19 and 20).

- If there are no remaining ports in the trunk after the module is removed, the trunk summary display shows the trunk without any ports.

Example If you had a trunk with ports 13 through 16 before you removed the module in slot 3, the trunk summary now shows an empty port list.

See Chapter 8 for more information on trunking.

Effects of Replacing Modules

When you remove a module, a number of changes occur, depending on the replacement module.

Replacing Modules of the Same Type or Same Number of Ports

If you remove a module that does not have any trunks or DAS ports and replace it with another module that has the same number of ports, the following changes occur:

- The port numbering is not affected — 10/100 Ethernet and FDDI modules can be exchanged without affecting the port numbers.

One Gigabit Ethernet module in any slot other than slot 4 must be replaced by another Gigabit Ethernet module to prevent port renumbering.

- The system remembers ports that were members of a VLAN. When another module is inserted into the empty slot, the ports are added back into the VLAN.

Replacing Modules of Different Types

More complicated changes occur when you swap six-port modules and one-port Gigabit Ethernet modules, replace FDDI modules that have DAS port pairs (because a DAS pair uses one bridge port to represent two physical ports), or replace modules on which you have trunks defined (because only the anchor port is used to define a trunk in a VLAN).

Port-Numbering Changes

Swapping six-port 10/100 Ethernet modules and FDDI modules does not cause the system to renumber ports, but swapping six-port modules and Gigabit Ethernet modules *does* cause the system to renumber ports.

Example If you have four 10/100 Ethernet modules, and you replace a 10/100 Ethernet module in slot 1 with a Gigabit Ethernet module, the ports are renumbered as follows: ports 1-6 become port 1, ports 7-12 become ports 2-7, ports 13-18 become ports 8-13, and ports 18-24 become ports 14-19.

VLAN Changes

- If you replace a six-port module with a Gigabit Ethernet module, the ports are renumbered, and any preexisting VLANs now include the Gigabit Ethernet port *only* if the VLANs previously included the first port of the six-port module.

Example If a VLAN contained ports 1 through 12 before you replaced the 10/100 Ethernet module in slot 1 with a Gigabit Ethernet module, the VLAN contains ports 1 through 7 after the change. (Port 1 is the Gigabit Ethernet port.)

- If a VLAN is defined over a Gigabit Ethernet module and you replace the module with a six-port module, only the *first* port of the new six port module is included in the VLAN after the change.

Example If a VLAN is defined over a Gigabit Ethernet module in slot 1 and six ports in slot 2 (that is, the VLAN has ports 1 through 7 configured) and you replace the Gigabit Ethernet module with an FDDI module with all SAS ports, the VLAN contains ports 1,7 through 12 after the change.

- If a VLAN has a DAS pair on an FDDI module, and the stationMode for that DAS port pair is changed to SAS (or the FDDI module is replaced by a six-port module), only the first SAS port of the previous DAS port pair is included in the VLAN after the change.

Example If a VLAN is defined over three DAS ports of an FDDI module in slot 1 and six Ethernet ports in slot 2 (that is, the VLAN has ports 1-12), and you change the FDDI ports from DAS to SAS, the VLAN contains ports 1 through 3 and 7 through 12 after the change.

Trunk Changes

- If you remove a module of a specific type that has trunks and replace it with a module of another type, the new ports do not become part of the trunk. When you define a trunk, the trunk is associated with a specific media type (100 Mb, Gigabit, or FDDI).

Example If you replace a 10/100 Ethernet module in slot 1 (that has a trunk on ports 5 and 6) with an FDDI module, the new FDDI ports 5 and 6 do *not* become part of the trunk. In this case, removing the 10/100 Ethernet module causes the 100 Mb trunk to lose all of its ports, although the trunk itself remains configured on the system.

Special Case: If you have four trunks and you replace a module of a given type with a Gigabit Ethernet module, the system cannot recognize the new Gigabit Ethernet module, because this module type uses a trunk resource. In this case, you must remove one of your trunks before you add the Gigabit Ethernet module (for example, a trunk associated with the removed module).

- If you replace a module that has a trunk spanning another module, after the change, the trunk is missing the ports associated with the removed module.

Example If a trunk spans two 10/100 Ethernet modules (ports 5-8) in slots 1 and 2, and the module in slot 1 is replaced by an FDDI module, after the change, the trunk display shows ports 5 and 6 as `Missing`, but the trunk still has the ports from the other Ethernet module (ports 7 and 8).

For more information on trunking, see Chapter 8. For information on VLANs, see Chapter 9.

5 ETHERNET

This chapter provides guidelines and other key information about how to implement Ethernet ports.

The chapter covers these topics:

- Ethernet Overview
- Key Concepts
- Key Guidelines for Implementation
- Port Enable and Disable (Port State)
- Port Labels
- Autonegotiation
- Port Mode
- Flow Control
- PACE Interactive Access
- Standards, Protocols, and Related Reading

You can manage Ethernet port features in either of these ways:

- *From the* `ethernet` *menu of the Administration Console. See the* Command Reference Guide.
- *From the Ethernet folder of the Web Management software. See the* Web Management User Guide.

Ethernet Overview

Ethernet is a standardized, packet-based network that supports an exponential hierarchy of three line speeds:

- **10 Mbps** — Ethernet
- **100 Mbps** — Fast Ethernet
- **1000 Mbps** — Gigabit Ethernet

All speeds of Ethernet are based on the IEEE 802.3 standard protocol called Carrier Sense Multiple Access with Collision Detection (CSMA/CD), which controls network access. With CSMA/CD, a station that intends to transmit listens for other Ethernet traffic on the network. When the station does not detect network activity, the station transmits.

Features

You can configure these features on Ethernet ports on the CoreBuilder® 3500:

- **Port state** — Whether a port is enabled and connected to a cable (on-line) or disabled (off-line)
- **Port label** — An alphanumeric port identifier
- **Port mode** — Port speed (10 Mbps, 100 Mbps, or 1000 Mbps) and duplex mode (half-duplex or full-duplex)
- **Autonegotiation** — A feature that allows some ports to automatically identify and negotiate speed and duplex mode with a receiving device
- **Flow control** — A Fast Ethernet and Gigabit Ethernet port mode that pauses and resumes transmissions
- **PACE® Interactive Access** — An algorithm that reduces network jitter, provides reliable timing, and optimizes LAN bandwidth use

In addition, some important Ethernet features depend on which Ethernet equipment you use, how you configure it, and how you connect it:

- **Trunking** — Increases bandwidth between switches and servers
- **Trunk Control Message Protocol (TCMP)** — Increases the availability of trunked links by handling physical configuration errors
- **Gigabit Interface Converter (GBIC)** — A Gigabit Ethernet port media type that allows you to hot-swap one media connector without affecting the other connectors

Benefits Ethernet, Fast Ethernet, and Gigabit Ethernet technologies allow you to configure and optimize:

- Link bandwidths
- Link availability

Link Bandwidths

As your network needs to support more users and increasingly bandwidth-intensive applications, you can configure Ethernet networks to keep pace with (or exceed) the capacity demands at two locations:

- **To end stations** — Depending on your application needs and network growth, you can migrate workstation connections from shared 10 Mbps to switched 100 Mbps Fast Ethernet. 3Com's Ethernet network interface cards (NICs) can automatically sense and configure themselves to an upgraded connection speed.
- **Between servers and switches** — Ethernet systems allow you to increase the bandwidth between switches or between servers and switches as your network requires. This increase is accomplished using *trunking* technology (also called *link aggregation*), which works at Open Systems Interconnection (OSI) Layer 2. For more information about trunking, see Chapter 8.

Link Availability

Ethernet technologies also allow you to design high levels of availability into your network through the use of trunking. A trunk enhances network availability because its underlying TCMP technology detects and handles physical configuration errors in point-to-point configurations. For more information about trunking, see Chapter 8.

Other Benefits

The hierarchy of Ethernet, Fast Ethernet, and Gigabit Ethernet technologies offers these additional network benefits:

- Easy configuration and expansion of point-to-point links
- Increased support for workstation moves, adds, changes, and upgrades
- Low-cost expansion of switch-to-switch or switch-to-server bandwidths without having to change device modules or cabling
- With PACE Interactive Access, reduction of network jitter, improved network timing, and optimization of LAN bandwidth use

Key Concepts

These concepts are important to implementing Ethernet:

- **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)** — The standardized Ethernet protocol that controls device access to the network
- **Collision** — When two or more stations attempt to transmit simultaneously
- **Port mode** — An Ethernet port's speed and duplex mode
- **Port speed** — 10 Mbps (Ethernet), 100 Mbps (Fast Ethernet), 1000 Mbps (Gigabit Ethernet)
- **Port state** — Whether a port is enabled and connected to a cable (on-line) or disabled (off-line)
- **Duplex mode** — Whether a port supports one-way (half-duplex) or two-way (full-duplex) transmissions
- **Autonegotiation** — A feature that allows some ports to identify and negotiate speed and duplex mode with a receiving device
- **Flow control** — A Fast Ethernet and Gigabit Ethernet port mode that pauses and resumes transmissions
- **Packet** — The basic unit of communications in Ethernet networks. While packets can vary in size, they have a consistent format.
- **Trunking** — A technology that combines multiple Fast Ethernet or Gigabit Ethernet ports into a single high-speed channel, thereby increasing bandwidth between switches and between servers and switches
- **Trunk Control Message Protocol (TCMP)** — A protocol that detects and handles physical configuration errors in a point-to-point configuration, thereby increasing availability of trunked links
- **Gigabit Interface Converter (GBIC)** — A Gigabit Ethernet port media type that allows you to hot-swap one media connector without affecting the other connectors

- **PACE® Interactive Access** — An algorithm that controls traffic flow on a point-to-point link with an end station. In a typical half-duplex Ethernet connection, you can never achieve high rates of utilization because of the randomness of collisions. If a switch and end station both try to send data, a collision occurs, forces retransmission, and lowers link utilization.

  PACE Interactive Access enables higher link utilization by altering the switch's *back-off* behavior. Instead of continuing to send data after winning a collision, the switch waits, allows the end station to send a packet, and then retransmits. The result is an interleaving of transmissions between the end station and the switch.

  This feature avoids repetitive collisions and prevents an end station from "capturing" the link. (With conventional Ethernet, a packet collision can cause the last station that transmitted successfully to monopolize Ethernet access and cause delays.)

- **Network areas** — 3Com uses a three-tiered framework to describe the functional areas in a LAN:

  - **Wiring closet** — This area provides connections to user workstations. It also includes downlinks into the data center or campus interconnect area.
  - **Data center** — This area receives connections from wiring closets and campus interconnect areas. Most local server farms reside here.
  - **Campus interconnect** — This area appears as a separate location only in larger networks; smaller networks usually have only wiring closets and data centers. The campus interconnect links campus data centers to each other. It may also include an enterprise server farm and connections to a wide area network.

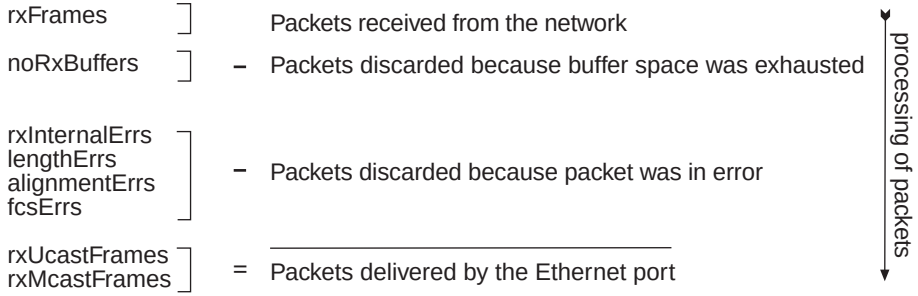
Ethernet Frame Processing

All frames on an Ethernet network are received promiscuously by an Ethernet port. A port can discard frames for either of the following reasons:

- There is no buffer space available.
- The frame is in error.

Figure 10 shows the order in which frame discard tests are made.

Figure 10 How Frame Processing Affects Ethernet Receive Frame Statistics

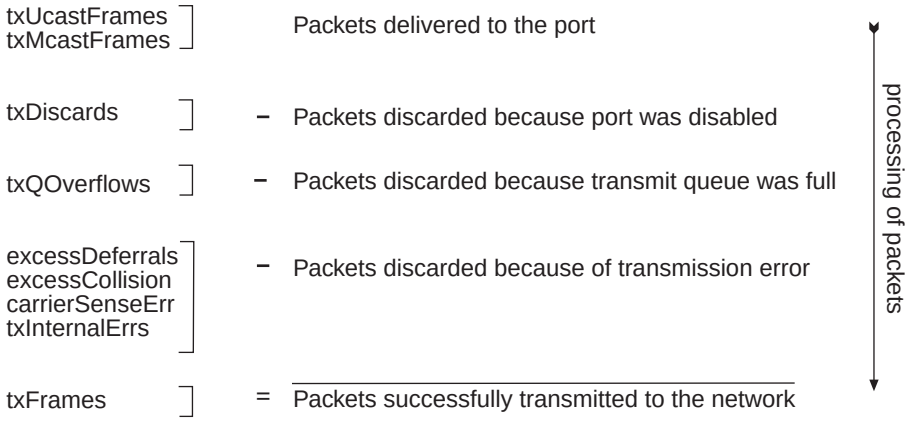


Frames also may be delivered directly to an Ethernet port by bridge, router, or management applications. A transmitted frame can be discarded for any of the following reasons:

- The Ethernet port is disabled.
- There is no room on the transmit queue.
- An error occurred during frame transmission.

Figure 11 shows the order in which these discard tests are made.

Figure 11 How Frame Processing Affects Ethernet Transmit Frame Statistics



Key Guidelines for Implementation

Consider these important factors when you implement and configure Ethernet networks.

Link Bandwidths

Recommended link capacities in a network normally depend on the speed requirements of end-user workstations, as shown in Table 5. In areas that may benefit from 1000 Mbps pipelines, you may be able to substitute trunked Fast Ethernet, subject to the issues raised in Chapter 8.

Table 5 Recommendations for Structuring Bandwidth Across the LAN

	Desktops to Wiring Closet	Wiring Closet to Data Center	Data Center to Campus Interconnect
Mainstream networks	Switched 10 or Shared 10/100	Switched 100	Switched 1000
Power networks	Switched 10/100	Switched 1000	Switched 1000+

Trunks

Consider these important factors when you implement and trunk Fast Ethernet or Gigabit Ethernet links:

- 3Com recommends that you use trunks to increase network availability in the following circumstances:
  - Switch-to-switch connections in the data center and campus interconnect areas
  - Switch-to-server connections in the data center and campus interconnect areas
  - Downlinks from the data center to the campus interconnect area
- When multiple links are trunked, it can be difficult to manage and troubleshoot individual port-to-port connections if a connectivity problem occurs. This issue may not be of concern in a server farm room. But if you use trunking extensively between wiring closets and data centers, the large number of connections involved and their distributed nature may make their management and troubleshooting difficult.

When you work with trunks, be sure that you understand the port numbering for your system. For port-numbering information on the CoreBuilder 3500, see Chapter 4. For more information about trunking, see Chapter 8.

Port Enable and Disable (Port State)

You can enable Ethernet ports (place them online) or disable them (place them off-line).

Important Considerations

- You can use this command to configure the same setting on multiple ports simultaneously. When you specify multiple port numbers, the system prompts you to choose the setting and then applies it to all of the ports.
- Because it stops all network traffic through the port, disabling a port may adversely affect a live network.
- When a port is enabled, the port transmits packets normally. When a port is disabled, the port neither sends nor receives packets.
- The portState is `off-line` for disabled ports and `on-line` for enabled ports that are connected to a network cable.

Port Labels

Port labels serve as useful reference points and as an accurate way for you to identify ports for management applications.

Labeling Ports

- Label Ethernet ports so that you can easily identify the devices that are attached to them (such as LANs, workstations, or servers). For example, you can assign **`engineeringserver`** as a label.
- The new port label appears in system displays the next time that you display information for that port.
- Port labels can include up to 32 ASCII characters, including the null terminator.

Autonegotiation

This feature enables some ports to identify and negotiate speed and duplex mode with a remote device.

Important Considerations

- You can use this command to configure the same setting on multiple ports simultaneously. When you specify multiple port numbers, the system prompts you to choose the setting and then applies it to all of the ports.
- In most cases, if autonegotiation does not properly detect the remote port speed, the vendor of the remote device implemented either autonegotiation or a change in port speed in a noncompliant way. If autonegotiation does not properly detect the port speed, you can manually set the port speed and duplex mode.
- Table 6 lists Ethernet port types on your system, whether they support autonegotiation, and which features they negotiate.

Table 6 Port Types and Autonegotiation Attributes

Port Type	Supports Autonegotiation?	Negotiable Attributes	Default Values for Negotiable Attributes
10/100BASE-TX	Yes	Port speed	10 Mbps
		Duplex mode	Half-duplex
100BASE-FX	No	Not applicable	Not applicable
1000BASE-SX	Yes	Duplex mode	Full-duplex
		Flow control	If autonegotiation is enabled, the system's best effort is On
1000BASE-LX GBIC	Yes	Duplex mode*	Full-duplex*
		Flow control	If autonegotiation is enabled, the system's best effort is On
1000BASE-SX GBIC	Yes	Duplex mode*	Full-duplex*
		Flow control	If autonegotiation is enabled, the system's best effort is On

* LX GBIC, and SX GBIC duplex modes are fixed at full-duplex at this release.

- **10/100BASE-TX ports** — Enabling autonegotiation causes both the port speed and duplex mode attributes to be autonegotiated.
- **100BASE-FX ports** — No autonegotiation of duplex mode occurs. The port speed is fixed at 100 Mbps. The default duplex mode is half-duplex.
- **1000BASE-SX ports** — Both link partners must either enable or disable autonegotiation. As long as autonegotiation is enabled, the system's best effort for handling flow control is `On`.
- When you enable autonegotiation, the system ignores your requested `portMode` information for 10/100BASE-TX ports and your requested `flowControl` information for 1000BASE-SX ports. When you disable autonegotiation, the system recognizes the requested `portMode` values for ports that have portMode options and the requested `flowControl` values for 1000BASE-SX ports.
- Use the `portMode` option to manually configure or modify the port speed and duplex mode. Use the `flowControl` option to manually configure or modify flow control.
- Autonegotiation is enabled by default on the ports that support it.

Port Mode

You can change the port speed and duplex mode for the 10/100BASE-TX ports and the duplex mode for 100BASE-FX ports. You cannot change the port speed or duplex mode for Gigabit Ethernet ports.

Important Considerations

- You can use this command to configure the same setting on multiple ports simultaneously. When you specify multiple port numbers, the system prompts you to choose the setting and then applies it to all of the ports.
- The device that is connected to each port must be configured for the same port mode. If the port speeds differ, the link does not come up. If the duplex modes differ, link errors occur.
- Gigabit Ethernet ports do not support mode options. The value `all` refers only to ports that support port mode options.
- If you change to full-duplex mode on the port, a message indicates that collision detection will be disabled unless you configure the connected device to the same duplex mode.
- Disable autonegotiation on any port on which you are setting a specific port mode.
- Table 7 lists the duplex port mode options available for each port type.

Table 7 Port Mode Options

Port Type	Duplex Port Mode	Resulting Port Mode	[Default]
10/100BASE-TX	100full	100 Mbps, full-duplex	10half
	100half	100 Mbps, half-duplex	
	10full	10 Mbps, full-duplex	
	10half	10 Mbps, half-duplex	
100BASE-FX	100full	100 Mbps, full-duplex	100half
	100half	100 Mbps, half-duplex	

Flow Control

The flow control mode allows a Fast Ethernet or Gigabit Ethernet port to:

- Decrease the frequency with which it sends packets to a receiving device, if packets are being sent too rapidly.
- Send flow control packets to a sending device, to request that the device slow its speed of transmission.

Important Considerations

Table 8 lists the effects of flow control options.

Table 8 Flow Control Options

Flow Control Option	Description	Available on Port Type
on	Port recognizes flow control packets and responds by pausing transmission. The port can generate flow control packets as necessary to slow incoming traffic.	Gigabit Ethernet Fast Ethernet
off	Port ignores flow control packets and does not generate flow control packets.	Gigabit Ethernet Fast Ethernet
rxOn	Port recognizes flow control packets and responds by halting transmission. The port does not generate flow control packets.	Gigabit Ethernet
txOn	Port ignores flow control packets, but it can generate flow control packets, if necessary.	Gigabit Ethernet

- You can use this command to configure the same setting on multiple ports simultaneously. When you specify multiple port numbers, the system prompts you to choose the setting and then applies it to all of the ports.
- The default setting for flow control is off.
- The system does not count flow control packets in receive or transmit statistics.

PACE Interactive Access

PACE Interactive Access prevents excessive network jitter (variation in the timing of packet delivery that can cause garbled sound, jerky images, and delays). PACE technology also improves timing and optimizes LAN bandwidth utilization.

Important Considerations

- You can use this command to configure the same setting on multiple ports simultaneously. When you specify multiple port numbers, the system prompts you to choose the setting and then applies it to all of the ports.
- Use PACE Interactive Access only on half-duplex Ethernet links between a switch and a single end station. (This setting has no effect on full-duplex links.)
- Do not use PACE Interactive Access when a repeater is connected to a switch port.

Standards, Protocols, and Related Reading

The system supports these Ethernet standards:

- **IEEE 802.3** — 10BASE-T Ethernet over unshielded twisted pair (UTP) wiring
- **IEEE 802.3u** — 100BASE-T Fast Ethernet over UTP or fiber-optic cable
- **IEEE 802.3z** — 1000BASE-SX Gigabit Ethernet over multimode fiber-optic cable and 1000BASE-LX Gigabit Ethernet over multimode or single-mode fiber-optic cable

Ethernet Protocols

- **IEEE 802.3** — Carrier Sense Multiple Access with Collision Detection, which controls Ethernet access. A station that intends to transmit listens for network traffic. If it detects none, it transmits.

  If two or more stations transmit at about the same time, their packets experience a *collision* and the colliding data streams do not reach their destinations. The sending stations stop transmitting, send a collision alert to other stations, and wait a random amount of time before trying again.

Media Specifications Table 9 summarizes the system's Ethernet media options.

Table 9 Ethernet Media Specifications

Type	Speed	Media	Connector	Recommended Distance (max)
10/100BASE-TX	10/100 Mbps	Category 5 UTP	RJ-45	100 m
100BASE-FX	100 Mbps	single-mode fiber	SC	20 km
		multimode fiber	SC	412 m (half-duplex) 2 km (full-duplex)
1000BASE-SX	1000 Mbps	multimode fiber	SC	220 m (62.5 micron @ 160 MHz*km modal bandwidth)
				275 m (62.5 micron @ 200 MHz*km modal bandwidth)
				500 m 50 micron @ 400 MHz*km modal bandwidth)
				550 m (50 micron @ 500 MHz*km modal bandwidth)
1000BASE-LX GBIC	1000 Mbps	single-mode fiber	GBIC	5 km (9 micron) (qualified for up to 10 km)
		multimode fiber	GBIC, with duplex SC conditioned launch cable	550 m (62.5 and 50 micron @ all modal bandwidths)
1000BASE-SX GBIC	1000 Mbps	multimode fiber	GBIC	550 m (62.5 and 50 micron @ all modal bandwidths)

Gigabit Ethernet Interface Converter (GBIC) ports are hot-swappable, that is, you can replace one GBIC connector while the other connectors continue to carry traffic.

To ensure optimal compatibility, performance, and regulatory compliance, use only GBIC transceivers and conditioned launch cables that 3Com supports. For information about currently supported GBIC specifications and conditioned launch cables, see the 3Com Web site:

`http://www.3com.com/gigabit_ethernet/gbics`

Related Reading For information about Ethernet media options, see the *CoreBuilder 3500 Getting Started Guide*.

6 Fiber Distributed Data Interface (FDDI)

This chapter provides an overview, key concepts, guidelines, and other key information about how to configure Fiber Distributed Data Interface (FDDI) in your system. This chapter covers these topics:

- FDDI Overview
- Key Concepts
- Key Guidelines for Implementation
- FDDI Stations
- FDDI Paths
- FDDI MACs
- FDDI Ports
- Station Mode (DAS and SAS)
- Sample FDDI Configurations

You can manage FDDI in either of these ways:

- *From the* `fddi` *menu of the Administration Console. See the* Command Reference Guide.
- *From the FDDI folder of the Web Management software. See the* Web Management User Guide.

FDDI Overview

Fiber Distributed Data Interface (FDDI) is a standards-based solution that provides fast and reliable data transfer on a local area network (LAN). FDDI technology, which supports data transfer of 100 million bits per second (100 Mbps), was developed by the American National Standards Institute (ANSI).

Features

FDDI technology:

- Uses optical fiber as its transmission medium, providing security, low signal loss, and high bandwidth data communication.
- Supports simultaneous connection of over 500 nodes on a ring, with up to 2 kilometers (1.2 miles) between adjacent nodes, and up to 200 kilometers (124 miles) of total fiber length.
- Uses a token-passing protocol for access to the network.
- Uses a dual-ring approach: a combination of two independent counter-rotating rings, each running at a data rate of 100 Mbps.
- Is the first LAN technology to provide an embedded network management capability.

Benefits

FDDI offers numerous benefits, many of which originate from the use of fiber-optic cable instead of copper cable.

- The FDDI standard specifies a data rate of 100 Mbps, which allows more data to be sent over optical fiber.
- The distance between nodes using multimode fiber is up to 2 km, which allows for a larger group of network users.
- Radio frequency interference (RFI) or electromagnetic interference (EMI) do not affect fiber-optic cable.
- Fiber-optic cable uses a dual ring topology that:
  - Provides fault tolerance and isolation.
  - Allows for ring wrapping in the event of a fault.
- FDDI uses a token access method that:
  - Supports larger networks.
  - Exploits the cable bandwidth more fully.
  - Eliminates collisions, similar to Carrier Sense Multiple Access/Collision Detect (CSMA/CD).

Key Concepts

Before you implement FDDI in your system, review the following FDDI standards, key concepts, and key terms.

Related Standards

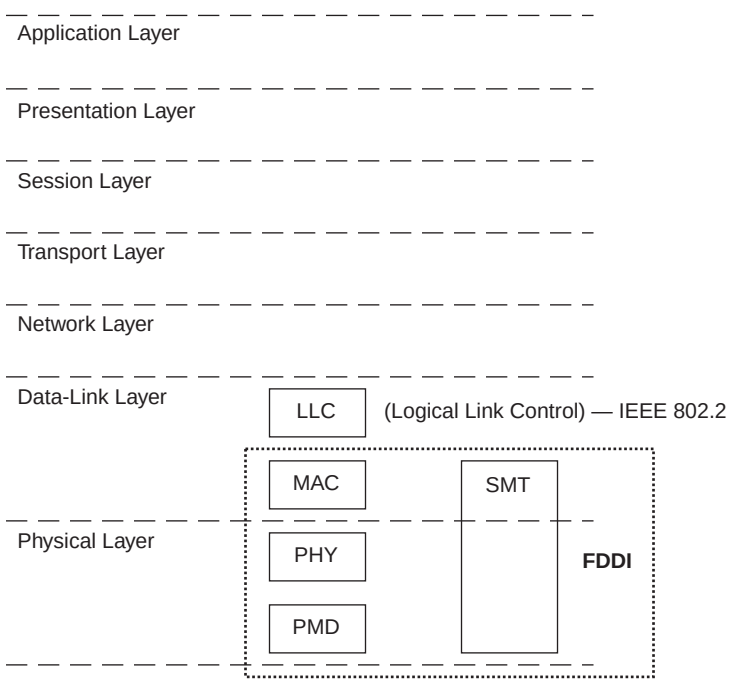
The industry guideline for FDDI technology is divided into four major standards:

- **Physical Medium Dependent (PMD)** — Specifies the characteristics of the fiber-optic medium, the connectors that attach stations to the fiber-optic medium, the transmission wavelength, the power requirements for transmitters, and the methods for optically bypassing inactive stations.
- **Physical (PHY)** — *S*pecifies data encoding and decoding, clock speed and clocking scheme, data framing, and the control symbols used in the network.
- **Media Access Control (MAC)** — *S*pecifies access to the medium, token passing, addressing, data checking, frame generation and reception, error detection and recovery, and the bandwidth allocation among the stations.
- **Station Management (SMT)** — *S*pecifies the FDDI station and ring configurations, initialization and maintenance of station-to-station connections, and the control required for the proper operation of stations in an FDDI ring.

These four standards are always described in relation to the Open Systems Interconnection (OSI) Reference Model. This model was established by the International Standards Organization (ISO) to standardize digital data communications. Each FDDI station is made up of logical entities that conform to the four standards. These entities represent the active services or management elements within OSI.

Figure 12 illustrates the relationship of FDDI entities to the OSI Reference Model. Network attachments communicate with each other using predetermined protocols. The model divides these communication protocols into seven layers, which are defined so that each layer only requires services from the layer below it.

Figure 12 FDDI Relationship to OSI Reference Model

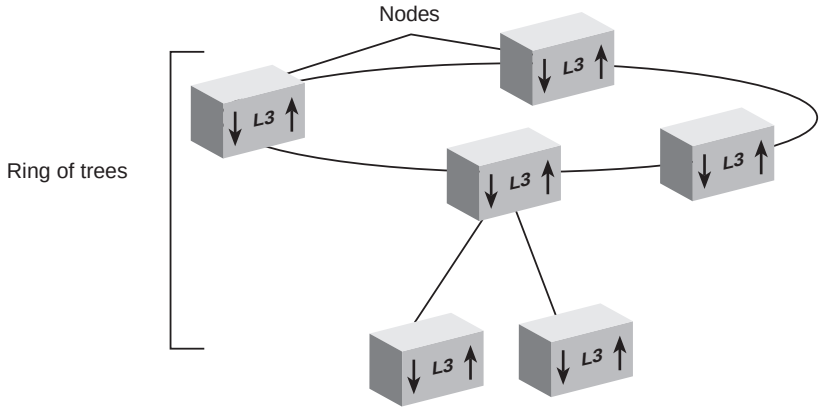


FDDI Network Topologies

The term *network topology* refers to the ways that stations are interconnected within a network. An FDDI network topology may be viewed at two distinct levels:

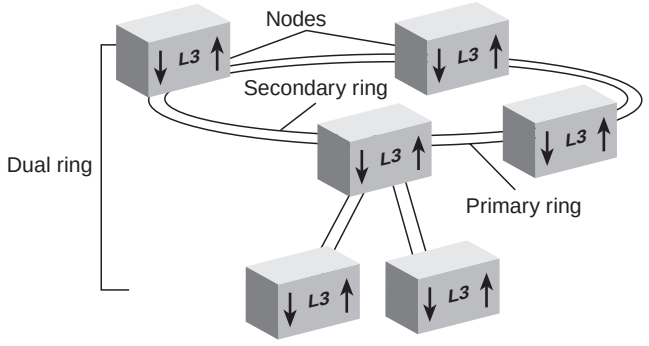
- **Physical topology** — A network's physical topology is defined by the arrangement and interconnection of its nodes. The FDDI physical topology is a *ring of trees*. See Figure 13.

Figure 13 Physical Topology



- **Logical topology** — A network's logical topology is defined by the paths through which tokens and data flow in the network. The FDDI logical topology is a *dual ring.* See Figure 14.

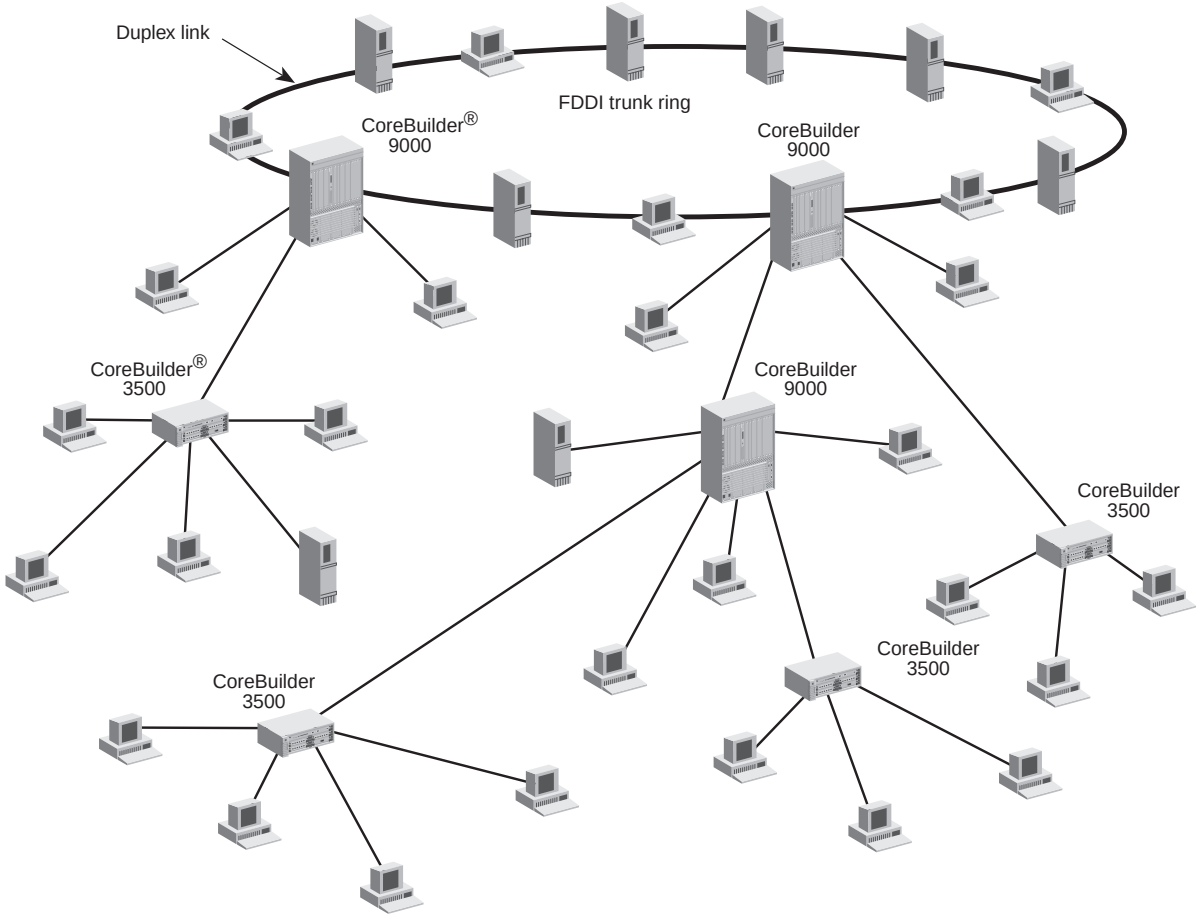
Figure 14 Logical Topology



Physical Topology: A Ring of Trees

The FDDI ring consists of dual-attach stations (DASs) and dual-attach connectors (DACs). The DACs on the ring allow you to attach *trees*. The trees consist of *branches* of single-attach stations (SASs) and DASs that are star-wired off of the concentrators. This kind of network is highly reliable, provides a single, fault-tolerant ring, offers fault isolation, and allows centralized management. See Figure 15.

Figure 15 Ring of Trees



All physical connections in an FDDI topology are *duplex links* (a pair of insulated fiber-optic conductors). Both the FDDI ring and the ring of trees that are created through concentrators are made up of duplex links. Interconnect the nodes in an FDDI network to form at *most* one ring.

If a topology is legal, when physical connections and nodes fail or are removed from the network, one or more legal FDDI topologies are formed. So subsets of legal topologies are also legal. Examples of legal FDDI topologies include the dual ring with trees, the dual ring without trees, and the single tree. For information about legal topologies, see "Setting the Connection Policies" later in this chapter.

Logical Topology: The Dual Ring

A legal FDDI topology consists of at most two separate logical rings: the primary ring and the secondary ring. These logical rings are formed from the physical links that make up the Physical Layer connections. For example, a set of DASs that are connected into a closed loop form an FDDI dual ring (that is, A to B; B to A). Each ring is a logical ring, that is, a separate data path with its own token.

Functionally, the dual ring provides a high degree of reliability to a LAN. When an FDDI network is in normal operation, only the primary ring transmits and receives data. The secondary ring may also carry data, but it is typically used as a backup in case there is a connectivity problem in the primary ring or in one of the nodes on the ring.

When a single fault takes place on an FDDI dual ring, recovery can be made by joining the two rings between the two nodes that are adjacent to the fault. Doing this creates a single logical ring, which results in a wrapped configuration. A wrapped ring is a legal FDDI topology. In the same way, when many faults take place, several disjointed logical rings are created, producing multiple FDDI topologies.

Nodes and Attachments

An FDDI network is made up of stations, concentrators, and switches that contain active services or management elements that conform to the ANSI FDDI standards. These stations and concentrators are connected to optical fiber medium and are attached in the prescribed manner set forth in the FDDI standards to allow reliable data transmission. Connections are made through FDDI ports and are managed by FDDI MACs.

Nodes

An FDDI network is made up of logically connected *nodes*. This generic term is used to refer to any active *station* or *concentrator* in an FDDI network.

- **Station** — Any addressable node on an FDDI network that can transmit, repeat, and receive information. A station contains only one SMT, and *at least one* MAC, one PHY, and one PMD.
- **Concentrator** — An FDDI station with additional PHY/PMD entities, beyond those required for its own connection to an FDDI network. These additional PHY/PMD entities (M ports) connect other FDDI stations, including other concentrators, in a tree topology.

Attachments

Attachments refer to how a node, station, or concentrator is connected to an FDDI network. They are classified as *single attachment* and *dual attachment.* Concentrators can be classified as *null attachment* when the A and B ports are either not present or not used.

- **SAS** — Single Attachment Station. A station or concentrator that has only one physical connection to an FDDI network. The single attachment cannot accommodate a dual (counter-rotating) ring. A single attachment station or concentrator has an S port that attaches to an M port within a concentrator tree.
- **DAS** — Dual Attachment Station. Any station or concentrator that has two physical connections to an FDDI network. This type of attachment can accommodate a dual (counter-rotating) ring. A dual attachment station has one A-B port pair; a dual attachment concentrator has an A-B port pair and at least one M port.

Node Types

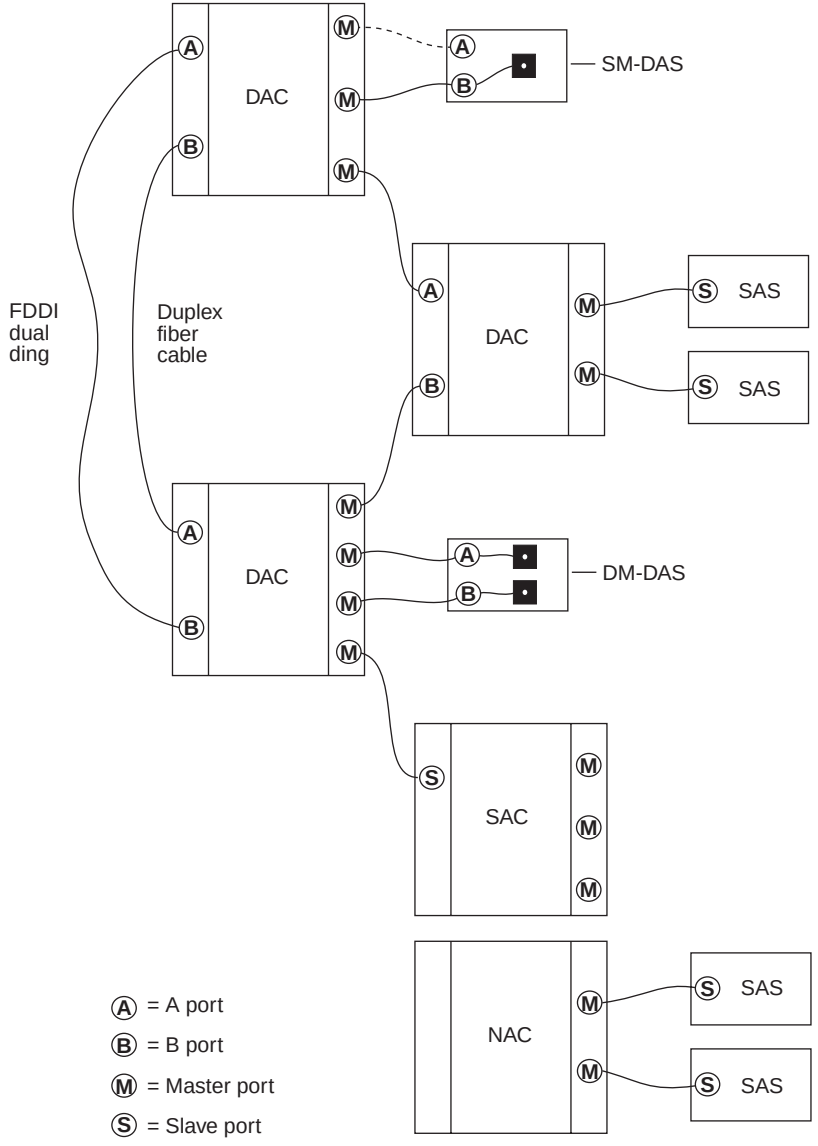
Six station and concentrator types are used to describe station configurations and topologies. Table 10 lists these node types and their abbreviations.

Table 10 Node Types and Abbreviations

Node Type	Abbreviation
Single MAC-Dual Attachment Station	SM-DAS
Dual MAC-Dual Attachment Station	DM-DAS
Single Attachment Station	SAS
Dual Attachment Concentrator	DAC
Single Attachment Concentrator	SAC
Null Attachment Concentrator	NAC

Figure 16 shows how these six node types may connect to an FDDI dual ring.

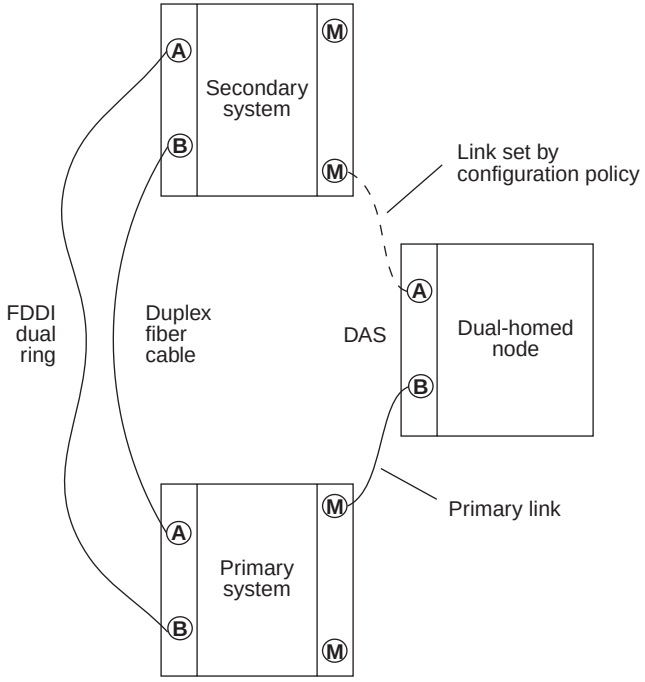
Figure 16 Examples of FDDI Node Types



Dual Homing

When the operation of a dual attachment node is crucial to your network, a configuration called *dual homing* can provide added reliability. Using dual homing you can determine a station's operation by setting the appropriate configuration policy. You can configure the dual-homed station with both links active or with one link active and one connection withheld as a backup. The backup connection becomes active only if the primary link fails. See Figure 17.

Figure 17 Dual Homing



FDDI Stations

Each FDDI station has one Station Management (SMT) entity to provide connection management, ring management, and operational management to the FDDI network. SMT specifies a set of services and signaling mechanisms that are dedicated to FDDI network management. It manages those services of each station on the FDDI network that are specific to the Physical Layer and the MAC portion of the Data Link Layer.

The goal of SMT is to completely define shared medium-management services to guarantee the interoperability of FDDI network equipment from multiple vendors.

SMT Operation

The operation of SMT falls into three broad categories:

- **Physical Connection Management (PCM)** — Establishes and maintains point-to-point physical links between neighboring ports. It provides all the signaling necessary to initialize connections, withhold marginal connections, and support maintenance.
- **Configuration Management (CFM)** — Interconnects PHYs and MACs on paths to achieve proper station configuration and network topology.
- **Ring Management (RMT)** — Manages a MAC's operation in an FDDI ring. RMT detects stations that are *stuck* in the beacon process and initiates the trace function. RMT locates duplicate addresses that might prevent the ring from operating.

FDDI MIB

The FDDI Management Information Base (MIB) defines the collection of information that is available to network management about an FDDI station. The MIB uses an object-oriented approach similar to that used in OSI management standards.

FDDI-managed objects include SMT (that is, the SMT of the station), MACs, paths, and ports. Each of these objects has a collection of attributes such as statistics, error counters, configuration information, event notifications, and actions.

You can access a station's MIB locally through a local management interface or remotely through a management protocol such as Parameter Management Frame (PMF) or Simple Network Management Protocol (SNMP). The SMT standard specifies the meaning and encoding of each MIB attribute.

Frame-based Protocols

SMT provides a number of frame-based services that higher level management functions use to manage stations on the network and to gather information about them. Frame-based protocols:

- Gather network statistics
- Detect, isolate, and resolve faults in the network
- Tune FDDI configuration and operational parameters to meet application and connectivity requirements

SMT has six key frame-based protocols:

- **Neighbor Notification** — Allows SMT to learn the addresses of the logical neighbors of each MAC in a station. This information is useful in detecting and isolating network faults.
- **Parameter Management** — Performs the remote management of station attributes. It operates on all SMT MIB attributes, attribute groups, and actions.
- **Status Reporting** — Allows a station to notify network managers about events such as station configuration changes and network errors.
- **Status Polling** — Provides a mechanism to obtain station status remotely through a request/response protocol.
- **Echo** — Performs loopback testing on the FDDI dual ring.
- **Synchronous Bandwidth Allocation** — Allocates synchronous bandwidth and monitors both synchronous and total bandwidth.

Primary and Secondary Paths

FDDI's dual, counter-rotating ring is made up of a primary and secondary ring. You can be connect FDDI stations to either ring or to both rings simultaneously. Data flows downstream on the primary ring in one direction from one station to its neighboring station. The secondary ring serves as a redundant path and flows in the opposite direction. When a link or station failure occurs, the ring *wraps* around the location of the failure, creating a single logical ring.

Paths represent the segments of a logical ring that pass through a station. An FDDI station can contain two paths:

- **Primary path** — The segment or segments of the primary ring that pass through a station. Conditions may exist in parts of the network that cause the path to be in a different ring. The primary path must be present in all nodes on the network.
- **Secondary path** — The segment or segments of the secondary ring that pass through a station. Conditions may exist in parts of the network that may cause the path to be in a different ring.

Media Access Control

The Media Access Control (MAC) uses a token-passing protocol to determine which station has control of the physical medium (the ring). The MAC delivers frames to their destinations by scheduling and performing all data transfers.

MAC Services

Some of the services that the MAC performs include:

- Frame repetition and reception
- Frame removal
- Frame validity criteria checking
- Token capture
- Token rotation
- Ring initialization
- Beacon process

MAC services are provided by all conforming stations that are attached to the FDDI network.

MAC Operation

The MAC controls access to the physical medium by passing a token around the ring. When a station receives the token, the station may transmit a frame or a sequence of frames. When a station wants to transmit, it removes the token from the ring and transmits the queued frames. After transmission, the station issues a new token, which the downstream station uses.

Stations that are not transmitting only repeat the incoming symbol stream. When repeating, the station determines whether the information was destined for it by comparing the destination address to its own address. If it sees a match, the MAC processes subsequent received symbols or sends them to the Logical Link Control (LLC) in the data-link layer for translation.

Ports

As parts of the Physical Layer, the PHY and PMD entities work together to support each link between FDDI stations. These entities provide the protocols that support the transmission and reception of signals between stations, as well as the optical fiber hardware components that link FDDI stations together. Within an FDDI station, the PHY and PMD entities make up a *port*. Together, they create a PHY/PMD pair that connects to the fiber-optic media and that provides one end of a physical connection with another station.

Ports at both ends of a physical connection determine the characteristics of that physical connection. The protocols that are executed at each port determine whether the connection is accepted or rejected. A connection is accepted if at least one station's policy allows such a connection. A connection is rejected if each station has a policy that disallows the connection.

Each port is one of four types: A, B, M, and S.

- **A port** — Connects to the primary ring on the incoming fiber and the secondary ring on the outgoing fiber. A properly formed FDDI dual ring is composed of a set of stations with the A port of one station connected to the B port of the neighboring station.
- **B port** — Connects to the incoming fiber of the secondary ring and the outgoing fiber of the primary ring.
- **M port** — Used by a concentrator station to provide connections within a concentrator tree. Also referred to as *Master port*.
- **S port** — Used by a single attachment station to provide attachment to an M port within a concentrator tree. Also referred to as *Slave port*.

Key Guidelines for Implementation

Consider the following guidelines when you configure and implement FDDI in your system:

- A high frame error rate often indicates a faulty station on the FDDI ring or a dirty FDDI connector or cable.
- If there is something wrong on your network, you may want to turn off data (user) traffic for a MAC by disabling LLC service. Although you have disabled data traffic from the MAC, the MAC still participates in neighbor notification and is visible to network management.
- The FDDI MAC path selections depend on the stationMode configuration (DAS or SAS).
- FDDI ports can be type A or type B for DAS ports.
- FDDI ports can be type S or type M for SAS ports.

- Before the new FDDI stationMode takes effect, you must reboot your system.
- You cannot modify fddi-stationMode port-pairs when any of the ports in the pair are members of a trunk.
- In a DAS configuration, the Activity LED from the secondary port (channel B) is not used by the port. If network traffic is passing through Channel B, the Activity LED on Channel A indicates port activity.

FDDI Stations

You can set the following FDDI station parameters:

- Connection policies
- Neighbor notification timer
- Status reporting

Setting the Connection Policies

The connectPolicy attribute is a bit string that represents the connection policies that are in effect on a station. A connection's *type* is defined by the types of the two ports involved (A, B, M, or S) in the connection. You can set the corresponding bit for each of the connection types that you want a particular station to reject.

The system's FDDI ports can be of type A or type B. By default, all connections to the systems FDDI ports are valid. Table 11 lists the possible connections to reject and their corresponding bits.

Effects and Consequences

When you set the connection policies, consider the following:

- By default all connections are valid on the system. An M-to-M connection is accepted so that a system port can be connected to another system port.
- Although an M-to-M connection is illegal within the FDDI standard, the CoreBuilder 3500 system allows this connection.

Table 11 Bit to Set for Rejecting a Station Connection

This Connection Is Rejected (System port - Remote port)	If This Bit Is Set	Connection Rules
A-A	0	Undesirable peer connection that creates twisted primary and secondary rings; notify station management (SMT).
A-B	1	Normal trunk ring peer connection.
A-S	2	Undesirable peer connection that creates a wrapped ring; notify SMT.
A-M	3	Tree connection with possible redundancy. The node may not go to Thru state in Configuration Management (CFM). In a single MAC node, Port B has precedence (with defaults) for connecting to a Port M.
B-A	4	Normal trunk ring peer connection.
B-B	5	Undesirable peer connection that creates twisted primary and secondary rings; notify SMT.
B-S	6	Undesirable peer connection that creates a wrapped ring; notify SMT.
B-M	7	Tree connection with possible redundancy. The node may not go to Thru state in CFM. In a single MAC node, Port B has precedence (with defaults) for connecting to a Port M.
S-A	8	Undesirable peer connection that creates a wrapped ring; notify SMT.
S-B	9	Undesirable peer connection that creates a wrapped ring; notify SMT.
S-S	10	Connection that creates a single ring of two slave stations.
S-M	11	Normal tree connection.
M-A	12	Tree connection with possible redundancy.
M-B	13	Tree connection with possible redundancy.
M-S	14	Normal tree connection.
M-M	15	Illegal connection that creates a tree of rings topology.

Setting Neighbor Notification Timer

The T-notify attribute is a timer that the Neighbor Notification protocol uses to indicate the interval of time between the generation of Neighbor Information Frames (NIF). NIF frames allow stations to discover their upstream and downstream neighbors. The T-notify value has a range of 2 through 30 seconds, with a default value of 30 seconds.

Effects and Consequences

When you set the neighbor notification timer, consider the following:

- By setting the T-notify value low, your network reacts quickly to station changes, but more bandwidth is used.
- By setting the T-notify value high, less bandwidth is used, but your network does not react to station changes as quickly.

Enabling and Disabling Status Reporting

The statusReporting attribute controls whether a station generates Status Report Frames (SRFs) to report events and conditions to network management stations. By default, status reporting is `enabled`. If you do not have an SMT management station that listens to these event reports or if you use SNMP to monitor FDDI events on all FDDI end-stations, you can set this attribute to `disabled` so that the station does not generate SRFs.

FDDI Paths

You can display FDDI path information and set the time values of the following attributes:

- tvxLowerBound
- tmaxLowerBound
- maxTreq

Setting tvxLowerBound

The tvxLowerBound attribute specifies the minimum time value of fddiMAC TvxValue that any MAC that is configured on this path uses. A MAC uses its valid transmission timer (TVX) to detect and recover from certain ring errors. If a valid frame has not passed through a MAC during the time indicated by fddiMACTvxValue, the MAC reinitializes the ring.

Effects and Consequences

When you set the tvxLowerBound attribute, consider the following:

- By adjusting the tvxLowerBound value, you specify how quickly the ring recovers from an error. The lower that you set this value, the faster the network reacts to problems, but the ring may reinitialize when there is no problem. The recommended value for tvxLowerBound is `2500` microseconds.
- The higher that you set the tvxLowerBound value, the less chance of frequent reinitializations, but the network takes longer to recover from errors. The recommended value for tvxLowerBound is `2500` microseconds.

Setting tmaxLowerBound

The tmaxLowerBound attribute specifies the minimum time value of fddiMAC T-Max that any MAC that is configured on this path uses. This value specifies the boundary for how high T-Req (the requested token rotation time) can be set.

Setting maxT-Req

The maxT-Req attribute specifies the maximum time value of fddiMACT-Req that any MAC that is configured onto this path uses. T-Req is the value that a MAC bids during the claim process to determine a ring's operational token rotation time, T_Opr. The lowest T-Req bid on the ring becomes T_Opr.

Effects and Consequences

When you set the maxT-req, consider the following:

- When T_Opr is a low value, the token rotates more quickly, so token latency is reduced. However, more of the ring's available bandwidth is used to circulate the token.
- Higher values of T_Opr use less bandwidth to circulate the token, but they increase token latency when the ring is saturated.

FDDI MACs

You can display MAC statistics and configure the following parameters:

- MAC FrameErrorThreshold
- NotCopiedThreshold
- Logical Link Control (LLC) service

Setting the Frame Error Threshold

The FrameErrorThreshold attribute determines when the system generates a MAC condition report because too many frame errors have occurred. A frame error occurs when a frame becomes corrupted.

Station Management (SMT) monitors the ratio of frame errors to all frames that are transmitted within a certain period of time. The FrameErrorThreshold setting determines at what percentage the frame errors are significant enough to report to network management. The threshold value is expressed in a percentage based on 65536 (which is 100 percent). For example, to set the threshold at 1 percent, the value is 655 (the system default). The lower that you set the percentage, the more likely it is for SMT to report a problem.

Effects and Consequences

When you set the frame error threshold, consider the following:

- A high error rate often indicates a faulty station on the FDDI ring or a dirty FDDI connector.

Setting the Not Copied Threshold

The NotCopiedThreshold attribute determines when the system generates a MAC condition report because too many frames could not be copied. Not-copied frames occur when there is no buffer space available in the station (which in turn indicates congestion in the station).

SMT monitors the ratio of frames that are not copied to all frames that are transmitted within a certain period of time. The NotCopiedThreshold setting determines at what percentage the number of frames that are not copied is significant enough to report to network management. The threshold value is expressed in a percentage based on 65536 (which is 100 percent). For example, to set the threshold at 1 percent, the value is 655 (the system default). The lower that you set the percentage, the more likely it is for SMT to report a problem.

Enabling and Disabling LLC Service

The Logical Link Control (LLC) service allows LLC frames to be sent and received on the MAC. LLC frames are all data frames that are transmitted on the network. If there is something wrong on your network, turn off data (user) traffic for a MAC by disabling LLC service. Although you have disabled data traffic from the MAC, the MAC still participates in neighbor notification and is visible to network management.

FDDI Ports

You can display port statistics and configure the following port parameters:

- lerAlarm
- lerCutoff
- port labels

Setting lerAlarm

The lerAlarm attribute is the link error rate (LER) value at which a link connection generates an alarm. If the LER value is greater than the alarm setting, then SMT sends a Status Report Frame (SRF) to the network manager software indicating a problem with a port.

Effects and Consequences

When you set the lerAlarm attribute, consider the following:

- The lerAlarm value is expressed as the absolute value of the exponent (such as 1×10^{-10}).
- A healthy network has an LER exponent between 1×10^{-10} and 1×10^{-15}.
- Set the lerAlarm value below these values so that you only receive alarms if your network is in poor health. The SMT Standard recommended value is 8.

Setting lerCutoff

The lerCutoff attribute is the link error rate estimate at which a link connection is disabled. When the lerCutoff value is reached, the PHY that detected a problem is disabled.

Effects and Consequences

When you set the lerCutoff attribute, consider the following:

- The lerCutoff value is expressed as an exponent (such as 1×10^{-10}).
- A healthy network has an LER exponent between 1×10^{-10} and 1×10^{-15}.
- Set the lerCutoff below these values so that a port is only removed only as a last resort. The SMT Standard recommended value is 7.
- The lerCutoff value must be lower than the lerAlarm value so that the network manager software is alerted to a problem before the PHY (port) is actually removed from the network.

Setting Port Labels

Port labels serve as useful reference points and as an accurate means of identifying your ports for management. Label your FDDI ports for easy identification of the devices attached to them (for example, workstation, server, FDDI backbone).

Station Mode (DAS and SAS)

You can modify the FDDI station mode that is assigned to a specific port number to either DAS (Dual Attachment Station) or SAS (Single Attachment Station) S port or M port. For the new station mode to take effect, you must reboot your system.

Single Attachment Station (SAS)

If you configure the FDDI ports as single-attached stations, each port is selectable as a bridge port. You can select your SAS ports to be either S or M ports.

Dual Attachment Stations

If you configure the FDDI ports as dual-attached stations, you must specify the lowest-numbered (anchor) port in the DAS pair. The other port becomes unselectable.

Effect and Consequence

When you set the station mode, consider the following:

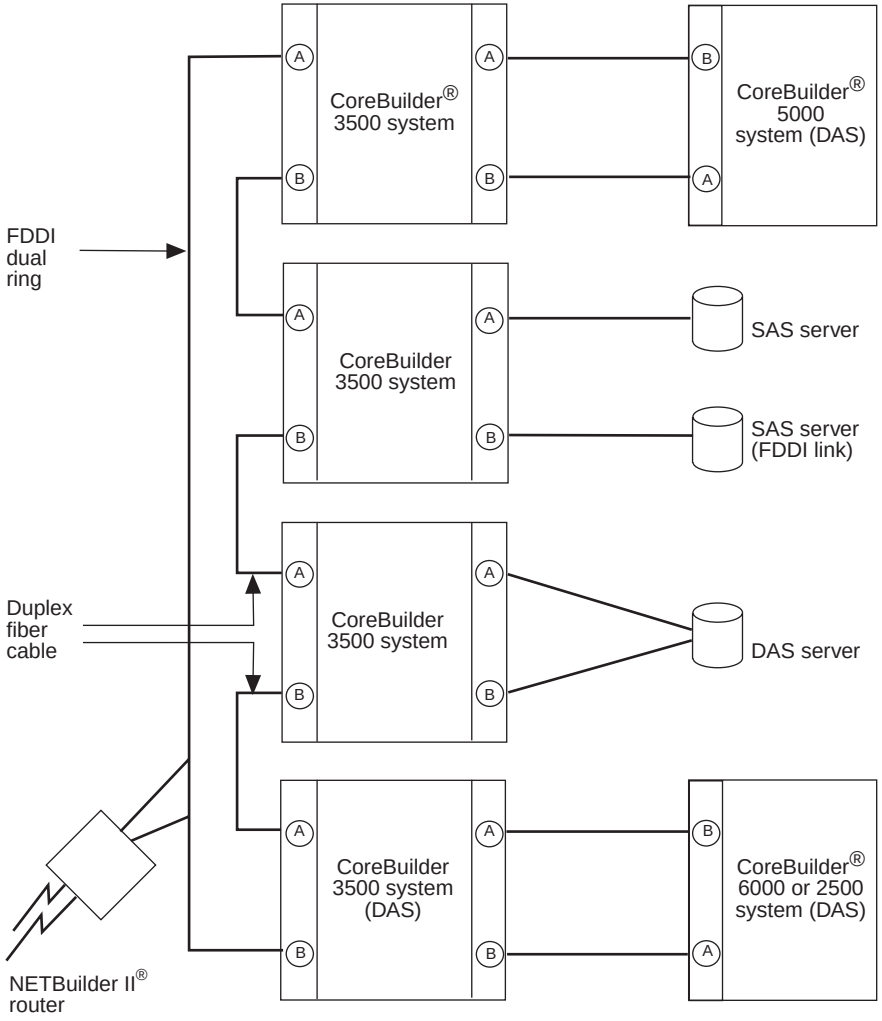
- When you modify the station mode, any FDDI ports that are associated with a VLAN or a trunk are removed from the VLAN or trunk.

Sample FDDI Configurations

You can install your system into many possible FDDI configurations. Figure 18 shows systems attached to an FDDI dual ring. The connection to the dual ring is made by the A and B ports on the system. DASs, excluding concentrators, may be attached to the dual ring, as shown.

CAUTION: *3Com strongly recommends that you connect equipment that can be turned on and off, such as workstations, only through concentrators. Connect intermediate systems that are seldom turned off, such as bridges and routers, to the FDDI dual ring only if they are equipped with an optical bypass switch. These precautions protect the integrity of the dual ring.*

Figure 18 Sample FDDI Configuration



Standards, Protocols, and Related Reading

This section describes how to obtain more technical information about FDDI.

Requests For Comments (RFCs)

Documents called Requests for Comments (RFCs) contain information about FDDI. Some of the RFCs that pertain to the discussions in this chapter are:

- **RFC 1130** — A Proposed Standard for the Transmission of IP Datagrams over FDDI Networks
- **RFC 1390** — Transmission of IP and ARP over FDDI Networks
- **RFC 1512** — FDDI Management Information Base

You can obtain copies of RFCs from the Internet Engineering Task Force (IETF) Web site:

`http://www.ietf.org`

Standards Organizations

Standards organizations ensure interoperability, create reports, and recommend solutions for communications technology. The most important standards groups are:

- International Telecommunications Union (ITU)
- Electronic Industry Association (EIA)
- American National Standards Institute (ANSI)
- International Standards Organization (ISO)
- Institute of Electrical and Electronic Engineers (IEEE)
- Internet Engineering Task Force (IETF)
- National Institute of Standards and Technology (NIST)

Related Reading

For more information about FDDI, be sure to refer to the following books:

- *Understanding FDDI: A 100Mbps Solution for Today's Corporate LANs*. Andrew Mills, Prentice Hall, 1995.
- *FDDI: A High Speed Network*. Amit Shah, G. Ramakrishnan, Akrishan Ram (Contributor), Prentice Hall, 1993

7 BRIDGE-WIDE AND BRIDGE PORT PARAMETERS

This chapter provides an overview of bridging concepts and the Spanning Tree Protocol and describes the bridging options and guidelines for your system.

The chapter covers these topics:

- Bridging Overview
- Key Bridging Concepts
- How the Spanning Tree Protocol Works
- Key Guidelines for Implementation
- STP Bridge and Port Parameters
- Frame Processing
- MAC Address Table
- IP Fragmentation
- IPX SNAP Translation
- Broadcast and Multicast Limit for Bridge Ports
- GARP VLAN Registration Protocol (GVRP)
- Standards, Protocols, and Related Reading

You can manage most bridge-wide and bridge port commands in either of these ways:

- *From the* `bridge` *menu of the Administration Console. See the* Command Reference Guide.
- *From the Bridge folder of the Web Management software. See the* Web Management User Guide.

Bridging Overview

A bridge interconnects two or more LANs and allows them to communicate as if they were one LAN. Bridges make forwarding decisions based on the information that the frames contain, and forward the frames toward the destination. Bridges operate at the Layer 2 data link layer of the Open Systems Interconnection (OSI) reference model. Because bridges operate at this layer, they are not required to examine the upper-layer information.

You system supports transparent bridging, a form of bridging that attaches two or more LANs, listens promiscuously to every packet that is transmitted, and stores each received packet until the packet can be transmitted on to other LANs.

Your system complies with the requirements that are outlined in the *IEEE 802.1D Media Access Control (MAC) Bridges* base standard. A compliant bridge must, at minimum:

- Learn source addresses from packets that stations on attached LANs transmitted.
- Age addresses of stations (on attached LANs) that have not transmitted a packet for a prolonged period.
- Store and forward packets from one LAN to another.
- Use the Spanning Tree Protocol (STP) for loop detection.

Benefits

Bridges provide the following benefits:

- Bridges extend the effective length of a LAN, allowing you to attach distant stations that could not otherwise be connected.
- Bridges can provide a level of separation that prevents some potential damaging errors or undesirable packets from spreading or multiplying on the network.
- Because bridges only forward a percentage of total traffic received, they diminish the traffic that devices on connected segments experience and increase available bandwidth.
- Bridges allow a larger number of devices to communicate than a single LAN can support.

Features

Your system supports several features that are closely related to the bridging process and are therefore categorized under `bridge` on the system interface.

The following bridging topics are covered in this chapter:

- **Spanning Tree Protocol (STP)** — You can configure bridge-wide and bridge port settings to calculate a network topology that reflects a single, loop-free path between any two devices.
- **Multicast and broadcast limits** — You can assign per-port multicast threshold values to limit the per-second forwarding rate of incoming broadcast and multicast traffic from the segment that is attached to that port.
- **GARP VLAN Registration Protocol (GVRP)** — You can enable your system to transmit and receive VLAN information using GVRP. GVRP is also addressed in Chapter 9.
- **IP fragmentation** — When Fiber Distributed Data Interface (FDDI) stations transmit IP packets that are too large for standard Ethernet to handle, IP fragmentation allows your system to reformat large packets into smaller sizes that can be bridged to Ethernet networks.
- **IPX SNAP translation —** IPX SNAP translation allows any 802.3_RAW IPX packets that are forwarded from Ethernet to FDDI to be translated to FDDI_SNAP (instead of FDDI_RAW), and vice versa.

The following bridging topics are covered in other chapters:

- **Virtual LANs (VLANs)** — A VLAN is a logical grouping methodology that allows dispersed users to communicate as if they were physically connected to the same LAN (broadcast domain). For more information about VLANs, including discussion of GVRP, see Chapter 9.
- **Trunking** — You can configure your system to aggregate multiple bridge port links into a single point-to-point trunk. Trunking allows you to increase bandwidth and redundancy without replacing cabling. For more information about trunking, see Chapter 8.

Key Bridging Concepts

Before you configure bridge-wide or bridge port parameters, review the following key concepts.

Learning Addresses

Bridges *learn* addresses so that they can determine which packets to forward from one bridge port to another. A bridge learns addresses by processing the network traffic that it receives. For a bridge to learn the address of a station on the network (a *source address*), that station must transmit a packet. Addresses that are learned are called *dynamic addresses*.

Each bridge maintains a table, called the *address table*, which lists each learned address and associates it with a port. (The address table also lists manually configured addresses called *static addresses*.)

The system can store up to 32 K addresses in its address table.

Aging Addresses

A dynamic address remains in the bridge's address table as long as the station to which it relates regularly transmits packets through the bridge. If the station does not transmit within a specified period of time, the dynamic address is *aged out* (deleted) from the address table.

Address aging ensures that, if a station moves to a different segment on the network, packets are no longer be forwarded to the station's former location. Address aging is necessary because a bridge can learn only a finite number of addresses.

Forwarding, Filtering, and Flooding Packets

A bridge filters, floods, or forwards packets by comparing:

- The packet's destination address to the source addresses in the bridge's address table.
- The destination bridge port (if known) to the port on which the packet was received.

The bridge compares the destination address to the addresses in the address table and does one of the following:

- *If the destination address is known* to the bridge, the bridge identifies the port on which the destination address is located.
  - If the destination bridge port is *different* from the bridge port on which the packet was received, the bridge forwards the packet to the destination bridge port.
  - If the destination bridge port is the *same* as the port on which the packet was received, the bridge filters (discards) the packet.
- *If the destination address is not known* to the bridge, the bridge forwards the packet to all active bridge ports other than the bridge port on which the packet was received. This process is called *flooding*.

Spanning Tree Protocol

A bridge maintains connectivity between LANs with assistance from the Spanning Tree Protocol (STP), which is specified in the IEEE 802.1D MAC Bridges standard.

When a bridge attaches to any single LAN with more than one path, this results in a *loop* in the network topology. Because the bridge receives the same packet from multiple ports within a short period of time, a loop can cause a bridge to continually question where the source of a given packet is located. As a result, the bridge forwards and multiplies the same packet continually, which clogs up the LAN bandwidth and eventually affects the bridge's processing capability.

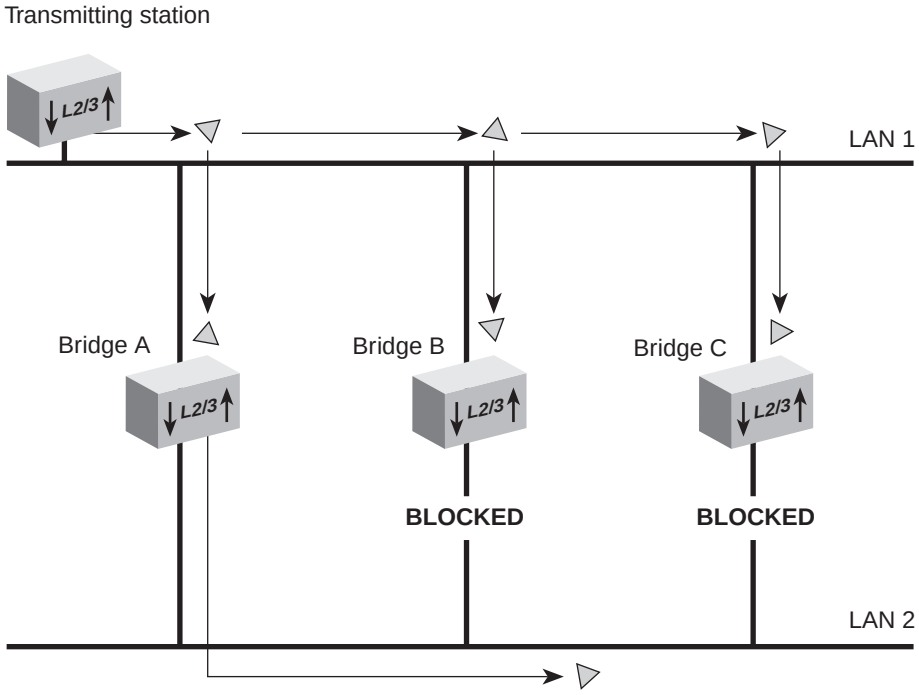
A backup or redundant path remains a valuable concept nevertheless. STP balances both concerns by allowing redundant paths to exist but keeps them inactive until they are needed.

STP uses an algorithm which compares the values in a few different parameters to determine all possible paths and then map out a loopless network topology which ensures that only one active path exists between every pair of LANs. STP keeps one bridge port active and puts redundant bridge ports in the *blocking* state. A port in the blocking state neither forwards nor receives data packets. See Figure 19.

After STP logically eliminates the redundant paths, the network configuration stabilizes. Thereafter, if one or more of the bridges or communication paths in the stable topology fail, STP recognizes the changed configuration and, within a few seconds, activates redundant links to ensure network connectivity is maintained.

For more detailed information about Spanning Tree, see "How the Spanning Tree Protocol Works" later in this chapter.

Figure 19 STP Blocks Redundant Links



How the Spanning Tree Protocol Works

Using the Spanning Tree Protocol (STP), bridges transmit messages to each other that allow them to calculate the Spanning Tree topology. These messages are special packets called *Configuration Bridge Protocol Data Units* (CBPDUs), or configuration messages.

CBPDUs at Work

CBPDUs do not propagate through the bridge as regular data packets do. Instead, each bridge acts as an end station, receiving and interpreting CBPDUs.

Bridge Hierarchy

The CBPDUs help bridges establish a hierarchy (or a *calling order*) among themselves for the purposes of creating a loopless network.

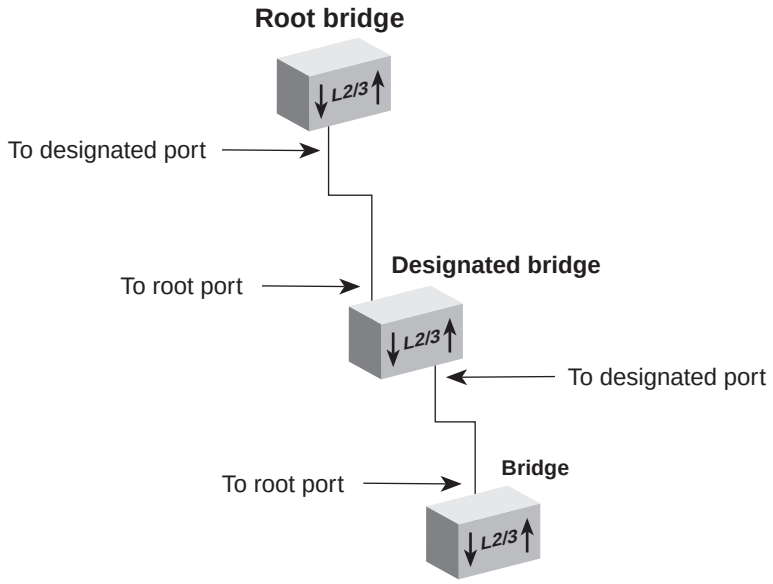
Based on the information in the CBPDUs, the bridges elect a *root bridge*, which is at the top level of the hierarchy. The bridges then choose the best path on which to transmit information to the root bridge.

The bridges that are chosen as the best path, called *designated bridges*, form the second level of the hierarchy:

- A designated bridge relays network transmissions to the root bridge through its *root port*. Any port that transmits to the root bridge is a root port.
- The designated bridges also have *designated ports* — the ports that are attached to the LANs from which the bridge is receiving information.

Figure 20 shows the hierarchy of the STP bridges and their ports.

Figure 20 Hierarchy of the Root Bridge and the Designated Bridge



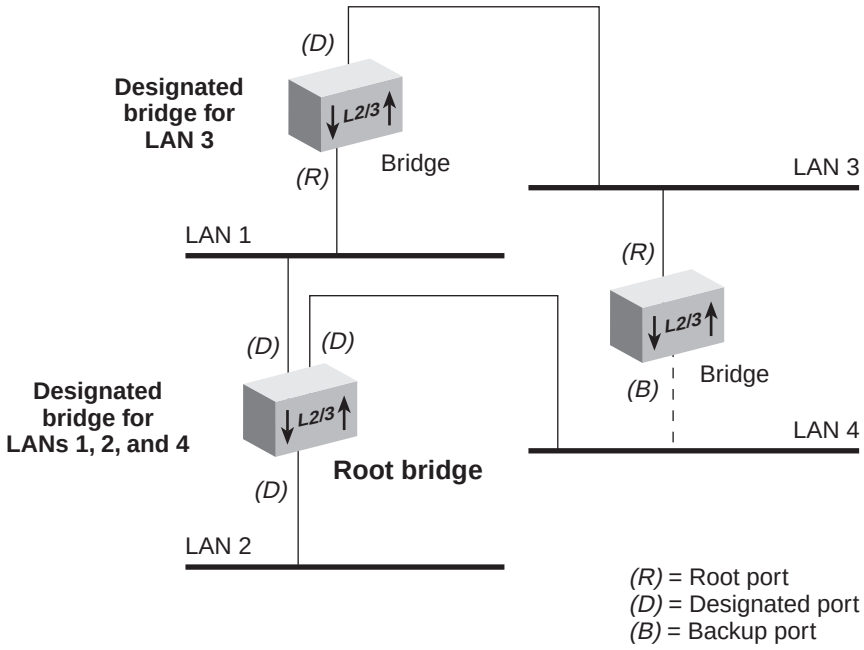
Actions That Result from CBPDU Information

From the information that the CBPDUs provide:

- Bridges elect a single bridge to be the *root bridge*. The root bridge has the lowest bridge ID among all the bridges on the extended network.
- Bridges calculate the best path between themselves and the root bridge.
- Bridges elect as the *designated bridge* on each LAN the bridge with the *least cost path* to the root bridge. The designated bridge forwards packets between that LAN and the path to the root bridge. For this reason, the root bridge is *always* the designated bridge for its attached LANs. The port through which the designated bridge is attached to the LAN is elected the *designated port.*
- Bridges choose a *root port* that gives the best path from themselves to the root bridge.
- Bridges select ports to include in the STP topology. The ports that are selected include the root port plus any designated ports. Data traffic is forwarded to and from ports that have been selected in the STP topology.

Figure 21 shows a bridged network with its STP elements.

Figure 21 STP Root and Designated Bridges and Ports



Contents of CBPDUs

Bridges use information in CBPDU to calculate a STP topology. The content of a CBPDU includes:

- **Root ID** — The identification number of the root bridge.
- **Cost** — The cost of the least-cost path to the root from the transmitting bridge. One of the determining factors in cost is the speed of the bridge's network interface; that is, the faster the speed, the lower the cost.
- **Transmitting bridge ID** — The identification of the bridge that transmits the CBPDU, which includes the bridge address and the bridge priority.
- **Port identifier** — Includes the port priority as well as the number of the port from which the transmitting bridge sent the CBPDU.

  The port identifier is used in the STP calculation only if the root IDs, transmitting bridge IDs, and costs (when compared) are equal. In other words, the port identifier is a tiebreaker in which the lowest port identifier takes priority. This identifier is used primarily for selecting the preferred port when two ports of a bridge are attached to the same LAN or when two routes are available from the bridge to the root bridge.

Comparing CBPDUs

Here are three examples that show how the bridge determines the best CBPDU. In every case, the root ID is the most important determining factor. If the root ID fields are equal, then the cost is compared. The last determining factor is the transmitting bridge ID. If the CBPDUs all have the same root ID, cost, and transmitting bridge ID, then the port identifier is used as a tiebreaker.

Example 1. Root ID is lower for Message 1. The bridge saves Message 1.

Message 1			Message 2		
root ID	**cost**	**transmitter**	**root ID**	**cost**	**transmitter**
12	15	35	31	12	32

Example 2. Root ID is the same for Message 1 and Message 2, but cost is lower in Message 1. The bridge saves Message 1.

Message 1			Message 2		
root ID	**cost**	**transmitter**	**root ID**	**cost**	**transmitter**
29	15	80	29	18	38

Example 3. Root ID and cost are the same for Message 1 and Message 2, but the transmitting bridge ID is lower in Message 1. The bridge saves Message 1.

Message 1			Message 2		
root ID	**cost**	**transmitter**	**root ID**	**cost**	**transmitter**
35	80	39	35	80	40

How a Single Bridge Interprets CBPDUs

The following case describes how *a single bridge* interprets CBPDUs and contributes to the Spanning Tree configuration.

1. When Spanning Tree is first started on a network, the bridge acts as if it is the root bridge and transmits a CBPDU from each of its ports with the following information:
   - Its own bridge ID as the root ID (for example, `85`)
   - Zero (0) as the cost (because, for the moment, it is the root bridge)
   - Its own bridge ID as the transmitting ID (for example, `85`)

   Thus, its CBPDU looks like this: `85.0.85`.
2. The bridge receives CBPDUs on each of its ports from all other bridges and saves the *best* CBPDU from each port.

   The bridge determines the best CBPDU by comparing the information in each message that arrives at a particular port to the message that is currently stored at that port. In general, the lower the value of the CBPDU, the *better* it is. When the bridge comes across a better CBPDU than it has stored, it replaces the old message with the new one.

3 From the messages that are received, the bridge identifies the root bridge.

   For example, if the bridge receives a CPBDU with the contents 52.0.52, then it assumes that the bridge with ID 52 is the root (because 52 is smaller than 85).

4 Because the bridge now knows the root bridge, it can determine its distance to the root and elect a root port.

   It examines CBPDUs from all ports to see which port has received a CBPDU with the smallest cost to the root. This port becomes the root port.

5 Now that the bridge knows the contents of its own CBPDU, it can compare this updated CBPDU with the ones that its other ports received:

   - If the bridge's message is better than the ones received on any of its ports, then the bridge assumes that it is the designated bridge for the attached LANs.
   - If the bridge receives a better CBPDU on a port than the message it would transmit, it no longer transmits CBPDUs on that LAN. When the algorithm stabilizes, only the designated bridge transmits CBPDUs on that LAN.

How Multiple Bridges Interpret CBPDUs

The previous section addressed how a single bridge reviews CBPDUs and makes decisions. The following examples illustrate how STP determines the topology for an entire network.

Figure 22 and Figure 23 shows the same network topology — six bridges that connect six LANs. The topology is designed with redundant links for backup purposes, which create loops in the extended network. Figure 22 shows the network at the start of the STP topology calculation. Figure 23 shows the network after the STP topology has stabilized.

Figure 22 Starting the Spanning Tree Calculation

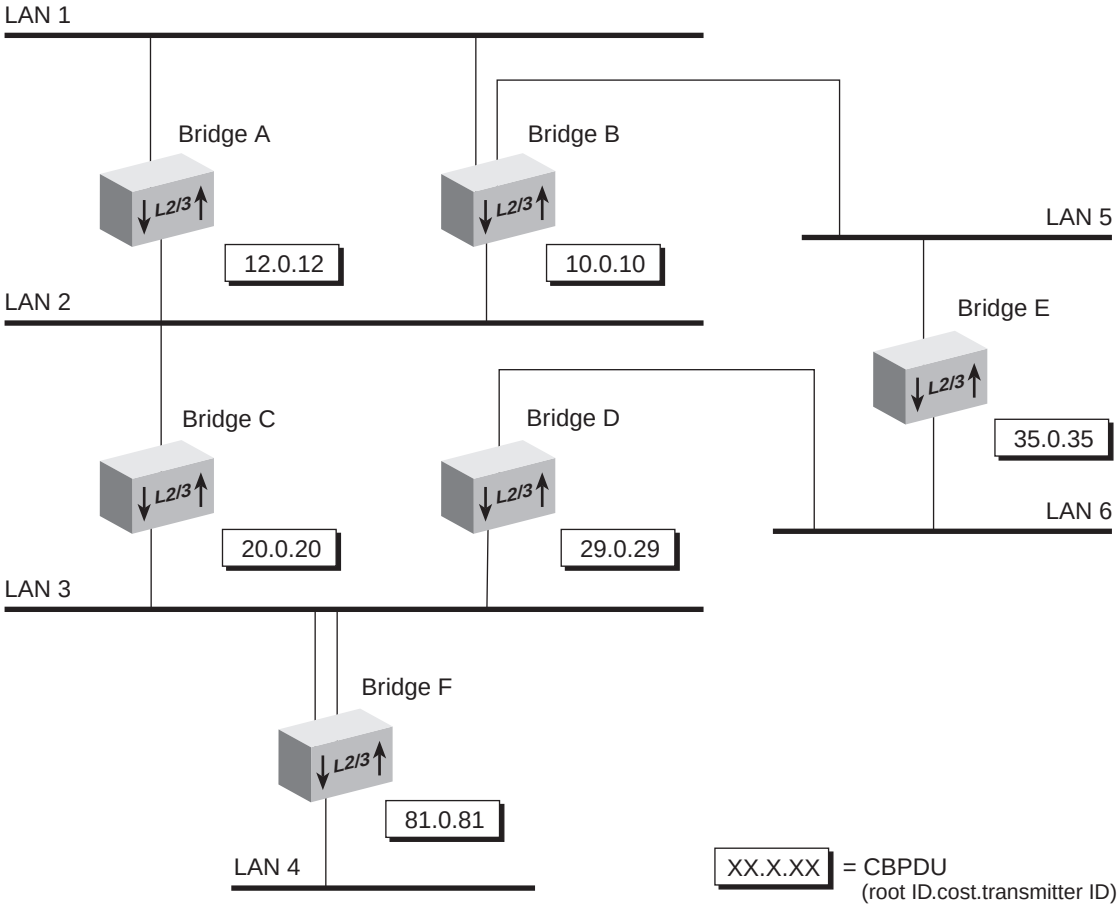
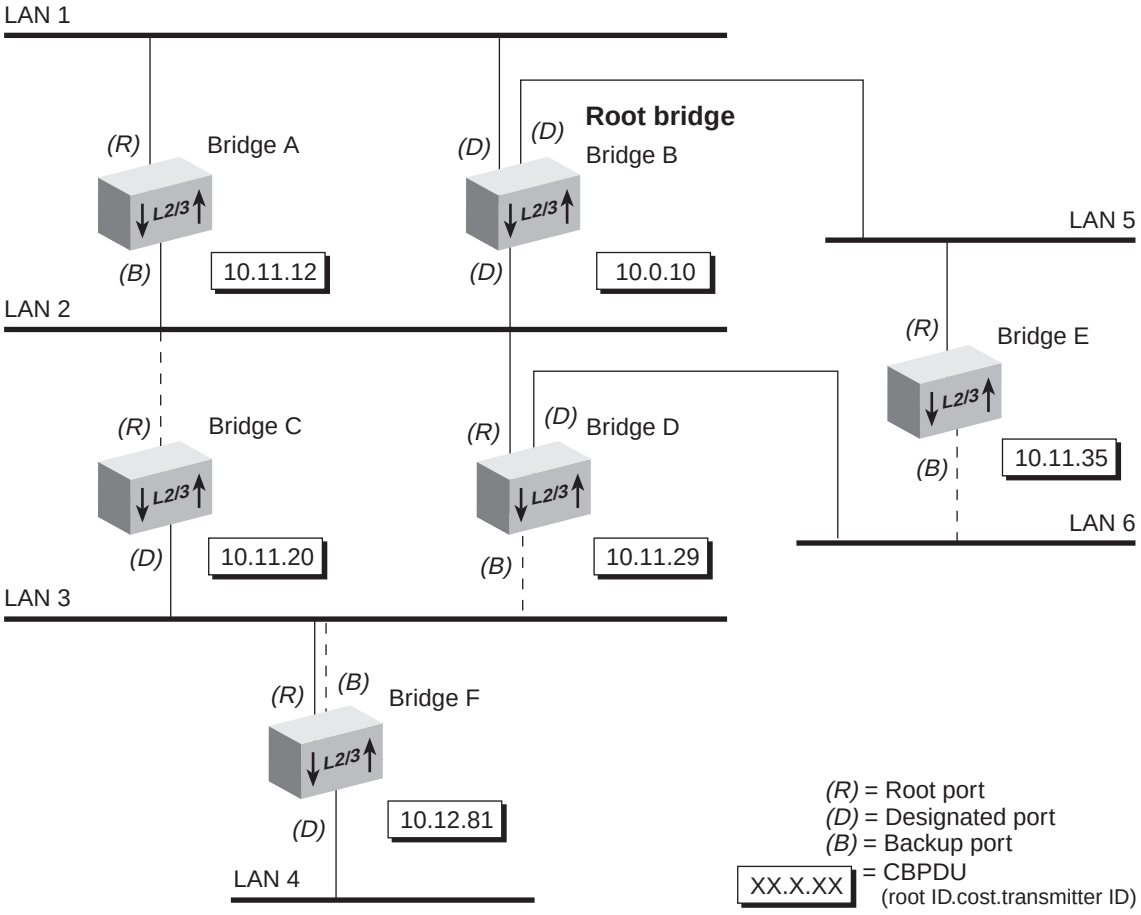


Figure 23 Spanning Tree Topology Calculated



Determining the Root Bridge

The root ID portion of the CBPDU determines which bridge actually becomes the root bridge. In Figure 22, notice how each bridge assumes itself to be the root and transmits a CBPDU that contains its own bridge ID as both the *root ID* and the *transmitting bridge ID*, and zero as the *cost*. In Figure 23, because Bridge B has the lowest root ID of all the bridges, it becomes the root and all other bridges change their root ID to Bridge B's ID (10).

Determining the Root Ports

Next, each bridge (except for the root bridge) must select a root port. To select a root port, each bridge determines the most cost-effective path for packets to travel from each of its ports to the root bridge. The cost depends on:

- The port path cost.
- The root path cost of the designated bridge for the LAN to which this port is attached.

If the bridge has more than one port attachment, the port with the lowest cost becomes the root port, and the other ports become either designated or backup ports. If bridges have redundant links to the same LAN, then the port with the lowest port identifier becomes the root port.

In Figure 23, Bridge F has two links to LAN 3 (through port 1 and port 2). Because the lowest port identifier for Bridge F is port 1, it becomes the root port, and port 2 becomes a backup port to LAN 3.

Determining the Designated Bridge and Designated Ports

For a LAN attached to a single bridge, that bridge is the LAN's designated bridge. For a LAN that is attached to more than one bridge, a designated bridge must be selected from among the attached bridges.

The root bridge functions as the designated bridge for all of its directly attached LANs.

For example, Bridge B, the root bridge in Figure 23, is also the designated bridge for LANs 1, 2, and 5.

A designated bridge must be determined for LANs 3, 4, and 6:

- Because Bridges C, D, and F are all attached to LAN 3, one of them must be the designated bridge for that LAN:
  - The algorithm first compares the root ID of these bridges, which is the same for all.
  - The cost is then compared. Bridge C and Bridge D both have a cost of 11. Bridge F, with a cost of 12, is eliminated as the designated bridge.
  - The transmitting bridge ID is compared between Bridge C and Bridge D. Because Bridge C's ID (20) is smaller than Bridge D's (29), Bridge C becomes the designated bridge for LAN 3.
- The designated bridge for LAN 6 is either Bridge D or Bridge E. Because Bridge D's transmitting bridge ID (29) is lower than Bridge E's (35), Bridge D becomes the designated bridge for that LAN.
- The designated bridge for LAN 4 is Bridge F, the only bridge that is attached to that LAN.

The port that attaches the designated bridge to the LAN determines the designated port. If more than one port is attached to the same LAN, then the port identifier determines the designated port.

Spanning Tree Port States

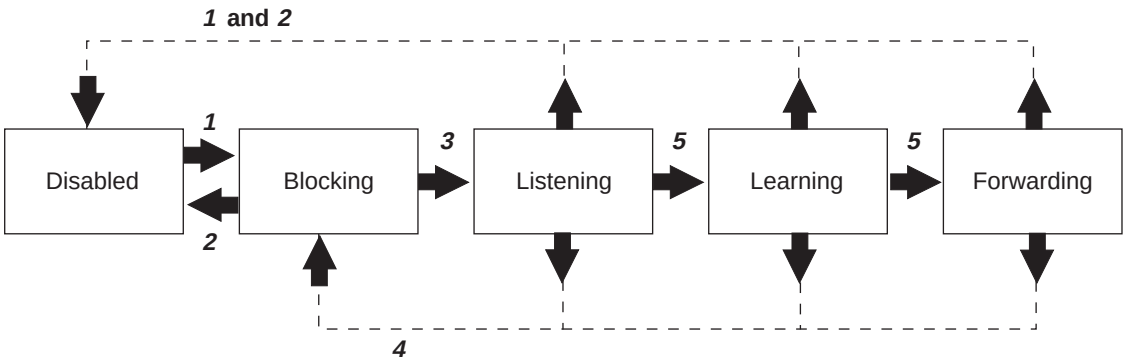
Because STP determines the network configuration or adjusts it, depending on events that occur, it places bridge ports in one of the following states at all times: listening, learning, forwarding, blocking, or disabled. Table 12 describes these states.

Table 12 Spanning Tree Protocol Port States

Port State	Description
Listening	When STP is configuring, all ports are placed in the listening state. Each port remains in this state until the root bridge is elected. While in the listening state, the bridge continues to run STP and to transmit CBPDUs on the port; however, the bridge discards data packets that are received on that port and does not transmit data packets from that port. The listening state should be long enough for a bridge to hear from all other bridges on the network. After being in the listening state, the bridge ports that are to proceed to the forwarding state go into the learning state. All other bridge ports go into the blocking state.
Learning	The learning state is similar to the listening state except that data packets are received on that port for the purpose of learning which stations are attached to that port. After spending the specified time in this state without receiving information to change the port back to the blocking state, the bridge changes the port to the forwarding state. The time that the port spends in each of the listening and learning states is determined by the value of the *forward delay* parameter.
Forwarding	After the port enters the forwarding state, the bridge performs standard bridging functions.
Blocking	When a port is put in the blocking state, the bridge continues to receive CBPDUs on that port (monitoring for network reconfigurations), but it does not transmit them. In addition, the bridge does not receive data packets from the port, learn the locations of station addresses from it, or forward packets onto it.
Disabled	A port is disabled when the STP has been disabled on the port or when the port has failed. In the disabled state, the port does not participate in the Spanning Tree algorithm. The port continues to forward frames only if STP is disabled for the entire bridge and the link is up.

Figure 24 illustrates the factors that cause a port to change from one state to another. The arrows indicate the direction of movement between states. The numbers correspond to the factors that affect the transition.

Figure 24 Factors in Spanning Tree Port State Transitions



1	Port enabled by either network administrator or initialization
2	Port disabled by either network administrator or failure
3	Spanning Tree algorithm selects port as designated or root
4	Spanning Tree algorithm does not select port as designated or root
5	Forwarding timer (forward delay) expires

As shown in Figure 24, for a port in the blocking state to transition to the listening state, STP must select that port as a designated or root port. After the port enters the listening state, forward delay must expire before the port can transition to the learning state. Then another forward delay period must expire (listening state) before the port can transition to the forwarding state. If you disable a port in the listening, learning, or forwarding state or if port initialization fails, then that port becomes disabled.

Reconfiguring the Bridged Network Topology

STP reconfigures the bridged network topology when any of the following events occur:

- Bridges are added or removed.
- The root bridge fails.
- You change any of the bridging parameters that influence the topology decision.

Resulting Actions

Whenever a designated bridge detects a topology change, it sends a Topology Change Notification Bridge Protocol Data Unit (BPDU) through its root port. This information is eventually relayed to the root bridge.

The root bridge then sets the Topology Change Flag in its CBPDU so that the information is broadcast to all bridges. It transmits this CBPDU for a fixed amount of time to ensure that all bridges are informed of the topology change.

If a port changes from the blocking state to the forwarding state as a result of the topology change, STP sends the topology information to all the ports before that port starts forwarding data. This delay prevents temporary data loops.

When a network reconfiguration occurs, a bridge flushes all dynamic addresses from its address table. This action ensures that the bridge learns the correct addresses and paths and continues to forward packets to the correct LANs.

Key Guidelines for Implementation

Consider the following guidelines when you configure bridge-wide and bridge port parameters on your system:

- When you disable bridge-wide STP, the bridge cannot participate in the algorithms for loop detection.
- Table 13 describes the forwarding behavior of a port based on its bridge and port STP states:

Table 13 Port Forwarding Behavior Depends on Bridge and Port STP States

Bridge STP State	Port STP State	Port Participates in STP?	Port Forwards Frames?
Disabled	Disabled	No	Yes, if link state is up.
	Enabled	No	Yes, if link state is up.
	Removed	No	Yes, if link state is up.
Enabled	Disabled	No	No
	Enabled	Yes	Determined by STP provided that the port link state is up.
	Removed	No	Yes, if link state is up.

- When STP is removed from the port but is enabled for the bridge, the port is invisible to STP but can forward frames. Removing the port from STP is useful if you have an edge switch device that is connected to end stations (such as PCs) that are frequently turned on and off.
- The port numbering shown for your ports is always sequential. See Chapter 4 for more information about port numbering.
- When you are prompted to select ports, specify the ? option to see a matrix of information about your bridge ports, including a Selection column, a Port column, and a Label column.
  - *Without trunking,* the Selection and Port columns contain the same port numbers, which indicates that you can select each port.
  - *With trunking,* the Selection column indicates that you can select the anchor port (lowest-numbered port) in the trunk, and the Port column shows each port that is associated with the trunk. The Label column contains the trunk name, if you have assigned one.

- If you want to specify a multicast limit for a trunk, be sure to apply it to the trunk's anchor port (lowest-numbered port) only. However, be aware that the multicast limit applies to *each link* in the trunk (that is, it is not an aggregate).
- You can enable STP with trunks. You may find it useful to configure a backup trunk that STP places in the blocking state. See Chapter 8 for more information about trunking.
- If you have specified `allClosed` as the VLAN mode and you want to administer bridge port address options, you must specify the correct VLAN interface index because each VLAN in `allClosed` mode has a unique address table.
- The system includes an "ignore STP mode" option that affects VLAN configurations. See Chapter 9 for more information or see the *Command Reference Guide*.
- GVRP is useful only when there are other switches or NICs in the network that support GVRP.
- You can define up to 32 bridge ports on the system. One consideration is that if you configure two or more ports of any technology type to form a trunk (a single logical bridge port), the system counts all ports in the trunk toward the bridge port limit.

STP Bridge and Port Parameters

On a bridge-wide basis, you can enable or disable the Spanning Tree Protocol (STP) and set STP bridge parameters. On a bridge-port basis, you can enable, disable, or remove STP and set STP bridge port parameters.

Administering Bridge-wide STP Parameters

You can set the following STP bridge-wide parameters:

- **STP state on a bridge** — When STP is disabled on the system, the bridge does not participate in the Spanning Tree algorithm and other STP settings have no effect on bridge operation or network topology calculations. If other devices on the network are running STP, then these packets are bridged.
- **Bridge priority** — The *bridge priority* influences the choice of the root bridge and the designated bridge. The *lower* the bridge's priority number, the *more likely* it is that the bridge is chosen as the root bridge or a designated bridge. The bridge priority value (`0x0-0xffff`) is appended as the most significant portion of a bridge identifier (for example: `8000 00803e003dc0`). It is a 2-octet value.
- **Bridge maximum age** — The *bridge maximum age* determines when the stored configuration message information is judged to be too old and is discarded from the bridge's memory. If the value is too small, then STP may reconfigure the topology too often, causing temporary loss of connectivity in the network. If the value is too large, the network may take longer than necessary to adjust to a new STP configuration after a topology change such as the restarting of a bridge. A conservative value assumes a delay variance of 2 seconds per hop. The recommended value is 20 seconds.

  The value that you set for bridge maximum age is only used if the system is selected as the root bridge. Otherwise, the system uses the value that is assigned to it by the root bridge.
- **Bridge hello time** — *Hello time* is the period between the configuration messages that a root bridge generates. If the probability of losing configuration messages is high, shorten the time to make the protocol more robust. Alternatively, to lower the overhead of the algorithm, lengthen the time. The recommended value is 2 seconds.

  The value that you set for bridge hello time is only used if the system is selected as the root bridge. Otherwise, the system uses the value that is assigned to it by the root bridge.

- **Bridge forward delay** — The *forward delay* value specifies the amount of time that a bridge spends in each of the listening and the learning states. This value temporarily prevents a bridge from starting to forward data packets to and from a link until news of a topology change has spread to all parts of a bridged network. The delay gives enough time to turn off to all links that need to be turned off in the new topology before new links are turned on.

  Setting the value too low can result in temporary loops while the Spanning Tree algorithm reconfigures the topology. Setting the value too high can lead to a longer wait while the STP reconfigures the topology. The recommended value is 15 seconds.

  The value that you set for bridge forward delay is only used if the system is selected as the root bridge. Otherwise, the system uses the value that is assigned to it by the root bridge.

- **STP group address** — The STP group address is a single address to which a bridge listens when it receives STP information. Each bridge on the network sends STP packets to the group address. Every bridge on the network receives STP packets that were sent to the group address, regardless of which bridge sent the packets.

  You may run separate STP domains in your network by configuring different STP group addresses. A bridge only acts on STP frames that are sent to the group address for which it is configured. Frames with a different group address are ignored.

  Because there is no industry standard about group address, bridges from different vendors may respond to different group addresses. If STP does not seem to be working in a mixed-vendor environment, verify that all devices are configured with the same group address.

Administering STP Parameters on Bridge Ports

You can enable, disable, or remove the Spanning Tree Protocol for one or more ports on the system. This setting affects the operation of a port only if the STP is enabled for the bridge. You can also set the following STP port parameters:

- **Port path cost** — The STP algorithm adds the path cost to the root cost field in a configuration message that is received on this port. The system uses this value to determine the path cost to the root through this port. You can set this value individually on each port. The range is 1 through 65535.

  A higher path cost value makes the LAN that is reached through the port more likely to be low in the Spanning Tree topology. The lower the LAN is in the topology, the less through traffic it carries. For this reason, assign a high path cost to a LAN that has a lower bandwidth or to one on which you want to minimize traffic.

- **Port priority** — The STP port priority influences the choice of port when the bridge has two ports connected to the same LAN, which creates a loop. The port with the lowest port priority is selected by STP. Port priority is a 1-octet value. The range for the port priority is 0x0 through 0xff hexadecimal. The default is `0x80`.

Frame Processing

All frames that are received on a physical interface and not explicitly directed to the system or discarded are delivered to the corresponding bridge port. The bridge port either forwards each frame to another bridge port or discards it.

The system can discard an incoming frame for the following reasons:

- The destination station is on the same segment as the source station.
- The receive bridge port is blocked.
- There is a problem with the frame.

  The physical interface does not deliver frames with errors to the bridge port. Thus, the `rxFrames` fields in the Ethernet statistics display and bridge statistics display often report different values — that is, the latter value is lower because it does not count frames in error.
- A user-defined packet filter indicated not to receive the frame.

A frame that is forwarded from a physical interface to a bridge port is then transmitted to a physical interface unless it is discarded. The system can discard a frame at this point for the following reasons:

- The transmit bridge port is blocked.
- The frame is too large for the corresponding physical interface.
- A user-defined packet filter indicated not to forward the frame.

MAC Address Table

The system includes several options for managing MAC addresses on bridge ports. The system recognizes two different kinds of addresses:

- **Static MAC addresses** — Addresses that you manually add to the bridge address table using menu options. These addresses never age; you must add and remove them manually.
- **Dynamic MAC addresses** — Addresses that the bridge learns by receiving and processing packets and ages. In the bridge address table, each dynamic address is associated with a specific port and is assigned an age so that it can be cleared from the table if the station is inactive.

Your system can store up to 32 K addresses.

Aging Time

The bridge aging time is the maximum period (in seconds) that dynamically learned forwarding information (addresses) is held in the bridge address table before it is aged out.

Use this parameter to configure the system to age addresses in a timely manner, without increasing packet flooding beyond acceptable levels.

Address Threshold

The address threshold is the value at which the system reports the total number of addresses that are known. Specifically, when this threshold is reached, the system generates the SNMP trap *addressThresholdEvent*.

The range of values that you can enter for this parameter is between 1 and 1 plus the maximum address table size (32 K). Setting the address threshold to one greater than the address table size prevents the system from generating events because the limit can never be reached.

Important Considerations

- All dynamic addresses are flushed from the bridge address table whenever you cycle power to the system or reboot the system. All dynamic addresses are also flushed when STP reconfigures the topology. Both dynamic and static addresses are flushed when you reset nonvolatile data.
- If you have multiple ports associated with a trunk, the addresses that are defined for the anchor port apply to all ports in the trunk.
- You can remove individual MAC addresses from selected ports. Typically, this action is only applied to static addresses because the system can quickly relearn dynamic addresses that you remove.

- A statically configured address is never aged and it cannot be learned dynamically on a different port until it is removed from the port on which it is configured.
- The number of static MAC addresses that you can configure depends on the availability of system resources.
- If a station whose address is statically configured on one port is moved to a different port, the system discards all received packets as a security measure and increments a statistical counter. (From the `bridge display` of the Administration Console, see the `rxSecurityDiscs` field. From the Bridge Display option on the Web Management interface, see the Received Security Discards column.)

IP Fragmentation

Standard FDDI allows larger maximum packet sizes than standard Ethernet. FDDI stations that transmit IP packet sizes larger than approximately 1500 bytes wish cannot communicate with stations on an Ethernet LAN. If the system receives such packets and they are destined for one or more Ethernet LANs, it filters them — except when IP fragmentation is enabled.

When you enable IP fragmentation, the system breaks up large FDDI packets into smaller packets before bridging them to Ethernet.

IPX SNAP Translation

IPX SNAP Translation allows an alternative method of translating IPX packets from Ethernet to FDDI and vice-versa.

- When IPX SNAP translation is enabled, any 802.3_RAW IPX packets that are forwarded from Ethernet to FDDI are translated to FDDI_SNAP. Likewise, SNAP IPX packets that are forwarded from FDDI to Ethernet are translated to 802.3_RAW packets.
- When IPX SNAP translation is disabled, the system uses standard IEEE 802.1H bridging to translate 802.3_RAW packets to FDDI_RAW packets.

Broadcast and Multicast Limit for Bridge Ports

You can assign a rate limit to any bridge port in the system to control the per-second forwarding rate of incoming multicast and broadcast packets. If the limit is reached, all remaining multicast and broadcast packets that are received in that second of time are dropped. This feature is useful for suppressing potential multicast or broadcast storms.

Important Considerations

When you set a limit, consider the following:

- A value of zero means that there is no limit set on the port. The system default is zero on all ports.
- You specify the limit in K frames per second (approximately 1000 frames per second). To determine an appropriate limit, measure the normal amount of broadcast or multicast traffic on your network.
- If you have IP multicast application traffic on your network, be sure that any limits that you configure do not constrain these traffic flows.
- If you want to specify a limit for a trunk, you only need to specify the trunk's anchor port (lowest-numbered port) when you configure the limit for the entire trunk. However, be aware that the multicast limit operates on *each link* in the trunk.
- There are similar options available through the Quality of Service menu. For more information, see Chapter 17.

GARP VLAN Registration Protocol (GVRP)

To activate GVRP on the system, you enable the GARP VLAN Registration Protocol (GVRP) first on the bridge and then on individual bridge ports.

On a port-by-port basis, GVRP allows the system to automatically learn the presence of and updates to 802.1Q VLANs. GVRP simplifies the management of IEEE 802.1Q VLAN configurations in large networks by making aspects of VLAN configuration dynamic.

GVRP maintains a database of VLAN member ports as the bridge learns about them. Specifically, GVRP tracks which ports are added to and removed from each VLAN and communicates this information to other GVRP-aware bridges. The bridges then determine active topologies for the network and for each VLAN using STP to prevent network loops.

GVRP operates only on ports that are in the STP forwarding state. If GVRP is enabled, a port that changes to the STP forwarding state automatically begins to participate in GVRP. A port that changes to an STP state other than forwarding no longer participates in GVRP. For more information about GVRP and VLANs, see Chapter 9.

Important Considerations

To use GVRP, consider the following:

- GVRP updates are not sent out to any blocked STP ports. GVRP operates only on ports that are in the STP forwarding state.
- GVRP is disabled by default on the bridge and on all bridge ports.
- Enabling GVRP determines whether the VLAN origin for a port-based VLAN is dynamic (GVRP enabled) or static (GVRP disabled).
- To maximize the effectiveness of GVRP, it should be supported in as many end stations and network devices as possible.
- Based on updates from GVRP-enabled devices, GVRP allows the system to dynamically create a port-based VLAN (unspecified protocol) with a specific VLAN ID and a specific port.
- On a port-by-port basis, GVRP allows the system to learn about GVRP updates to an existing port-based, protocol-based, or network-based VLAN with that VLAN ID and IEEE 802.1Q tagging.
- VLANs that are created dynamically with GVRP exist only as long as a GVRP-enabled device is sending updates — if the devices no longer send updates, or if GVRP is disabled, or if the system is rebooted, all dynamic VLANs are removed.

- GVRP manages the active topology, not nontopological data such as VLAN protocols. If a local bridge needs to classify and analyze packets by VLAN protocols, you must manually configure protocol-based VLANs and simply rely on GVRP to send VLAN ID updates. But if the local bridge needs to know only how to reach a given VLAN, then GVRP provides all necessary information.
- The VLAN topologies that GVRP learns are treated differently from VLANs that are statically configured. Although static updates are saved in nonvolatile RAM, GVRP's dynamic updates are not. When GVRP is disabled, the system deletes all VLAN interfaces that were learned through GVRP and leaves unchanged all VLANs that were configured through the Administration Console or through the Web management software.

Standards, Protocols, and Related Reading

For more information about bridging, STP, and GVRP consult the following standards:

- **IEEE 802.1D** — This standard specifies the requirements to which your system, as a transparent bridge, complies.
- **IEEE 802.1Q** — This standard defines GVRP, tagging, and the dynamic registration of VLANs.

To obtain copies of these standards, register for an on-line subscription at the Institute of Electrical and Electronics Engineers (IEEE) Web site:

`http://www.ieee.org`

8 TRUNKING

This chapter provides guidelines, limitations, and other important information about how to implement the trunking function for CoreBuilder® 3500 systems. This chapter covers the following topics:

- Trunking Overview
- Key Concepts
- Key Guidelines for Implementation
- Defining Trunks
- Modifying Trunks
- Removing Trunks
- Standards, Protocols, and Related Reading

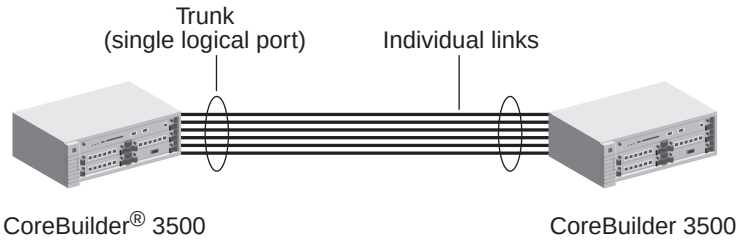
You can manage trunking in either of these ways:

- *From the* `bridge trunk` *menu of the Administration Console. See the* Command Reference Guide.
- *From the Define Wizard in the Bridge Trunk folder of the Web Management software. See the* Web Management User Guide.

Trunking Overview

A *trunk* (also known as an *aggregated link*) works at Layer 2 and allows you to combine multiple Fast Ethernet, Gigabit Ethernet, or FDDI ports into a single high-speed link between two switches (see Figure 25).

Figure 25 Example of a Trunk



The system treats trunked ports in the same way that it treats individual ports. Also, all higher-level network functions — including Spanning Tree algorithms, VLANs, and Simple Network Management Protocol (SNMP) management — do not distinguish a trunk from any other network port.

Features

You can configure the following trunking features:

- **Define** — You specify ports and characteristics associated with the trunk.
- **Modify** — You modify a trunk's characteristics or add or remove a port from the trunk.
- **Remove** — You remove a trunk definition from the system.

Benefits

Trunking can help you meet your network capacity and availability needs. With trunks, you can cost-effectively increase the bandwidth between switches or between servers and switches as your network requires. With trunking, you combine multiple Fast Ethernet, Gigabit Ethernet, or Fiber Distributed Data Interface (FDDI) ports into a single high-speed channel.

If Gigabit Ethernet is not available, you can use trunked Fast Ethernet to increase network capacity. After Gigabit Ethernet is in place and the time comes to scale links beyond 1000 Mbps, you can use trunking to create multigigabit connections.

Trunks also enhance network availability, because the Trunk Control Message Protocol (TCMP) detects and handles physical configuration errors in the point-to-point configuration. The system automatically distributes traffic across the ports that are associated with the trunk. If any of the trunk's ports go down or up, the system automatically redistributes traffic across the new arrangement of operational ports.

Key Concepts

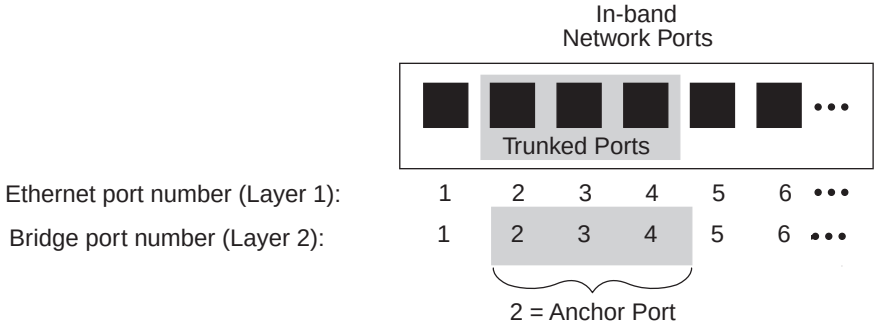
Before you configure trunking on your system, become familiar with the key concepts in this section.

Port Numbering in a Trunk

When you combine ports on a trunk, the system logically groups the physical ports that you specify into a single bridge port, identified by a single bridge port number in bridge statistics. For example, Figure 26 shows that Ethernet ports 2, 3, and 4 are represented by bridge port 2 after trunking.

The lowest numbered port in the trunk, called the *anchor port*, represents the entire trunk. After trunking, you can select bridge port 2 when you specify bridge port or virtual LAN (VLAN) information, but you cannot select bridge ports 3 or 4 since they are part of the trunk.

Figure 26 Bridge Port Numbering After Trunking



Regardless of whether you define trunking, the physical port numbering on your system remains the same.

It is important to understand the relationships between Ethernet, bridge, and VLAN port-related information:

- **Ethernet port information** — Each physical port is always listed individually, regardless of whether it is part of a trunk.
- **Bridge port information** — This information uses the concept of bridge ports. When you perform bridge port operations, you specify the trunk's anchor port, not the other ports in the trunk, as the representative bridge port. In the bridge port displays, each selectable bridge port has a port field that contains multiple port numbers if the bridge port represents a trunk (for example, `3,5` or `6-8`).
- **VLAN information** — When you define VLANs (as described in Chapter 9), you must specify the bridge ports that you want to be part of the VLAN. If you have a trunk, you specify its anchor port as the bridge port. The VLAN that you create then includes all of the physical ports in the trunk.

Trunk Control Message Protocol (TCMP)

The Trunk Control Message Protocol (TCMP) performs the following functions:

- Detects and corrects trunks that violate trunk configuration rules
- Ensures orderly activation and deactivation of trunk ports

The system runs a separate TCMP agent for each trunk. If TCMP detects an invalid configuration, the protocol restricts the trunk to the largest subset of ports that is a valid configuration.

Enabling TCMP is optional, but recommended. If TCMP is disabled, the network still functions, but without automatic trunk validation and reconfiguration. By default, TCMP is enabled.

Each TCMP agent:

- Periodically transmits a TCMP helloMessage through every trunk port.
- Continuously listens for helloMessages from other trunk ports.
- Builds a list of ports that TCMP has detected.
- Uses this list to activate or deactivate trunk ports to maintain valid trunk configurations.

TCMP uses three trunk port states to control port activation and deactivation:

- **notInUse** — A trunk port in this state has not been *selected* to participate in the trunk.
- **selected** — TCMP has *selected* the trunk port to participate in the trunk, but the port has not yet become *active*.
- **inUse** — A trunk port is fully *active* on the trunk.

Key Guidelines for Implementation

Consider the following important factors when you implement and configure trunks:

General Guidelines

- Create trunks before you define VLANs.
- The system supports four point-to-point trunks, each built from up to eight ports. All channels in a trunk must be *parallel* and must connect:
  - Correctly configured ports
  - Identical types of ports (with no two ports on a trunk connected to the same network)
  - Identical types of network nodes (switches or servers)
- You cannot mix FDDI, Fast Ethernet, and Gigabit Ethernet links in a trunk. All links to be trunked must be homogeneous.
- When multiple links are trunked, it can be difficult to manage and troubleshoot individual port-to-port connections if a connectivity problem occurs. This issue may not be of concern in a server farm room. But if you use trunking extensively between wiring closets and data centers, the large number of connections involved and their distributed nature may make their management and troubleshooting difficult. 3Com recommends that you apply trunking only *within* data center and campus interconnect areas.

- 3Com recommends that you use trunks to increase network availability in the following scenarios:
  - Switch-to-switch connections in the data center and campus interconnect areas
  - Switch-to-server connections in the data center and campus interconnect areas
  - Downlinks from the data center to the campus interconnect
- The trunking feature in 3Com switches is currently a proprietary implementation. No *de facto* standards currently exist.

Trunk Capacity Guidelines

- The device-to-device burst-transmission rate across a trunk is limited to the speed of just *one* of the port-to-port links within the trunk. For example, the maximum burst rate over a 400 Mbps pipeline with four trunked Fast Ethernet links is 100 Mbps. This limitation preserves frame ordering between devices, usually by moving all traffic between two specific MAC addresses across *only one port-to-port link*. Therefore, trunking provides no direct benefit for some one-way applications, such as server-to-server backups. This limit exists for most vendor implementations.
- The total throughput of a trunk is typically less than the bandwidth obtained by adding the theoretical capacity of its individual links. For example, four 1000 Mbps links do not yield a 4000 Mbps trunk. This is true with all vendor implementations.

- A trunked Fast Ethernet pipeline may seem to offer comparable bandwidth to a single Gigabit Ethernet link, and trunked Fast Ethernet may seem like a good way to buy some time before you upgrade connections to Gigabit Ethernet. Table 14 shows that given a choice, trunking Fast Ethernet may not be a cost-effective strategy.

  If you cannot upgrade to Gigabit Ethernet, then trunking Fast Ethernet in switch-to-switch or switch-to-server links can help you fine-tune or expand network capacity. After Gigabit Ethernet is in place, you can use trunking to further expand switch-to-switch or server-to-switch links.

Table 14 Comparing Gigabit Ethernet with Trunked Fast Ethernet

Comparison Point	Gigabit Ethernet	Trunked Fast Ethernet
Max burst rate	1000 Mbps	100 Mbps
Max aggregate rate	1000 Mbps	600 Mbps (over 10 links)
	(2000 Mbps full duplex)	(1200 Mbps full duplex)

Defining Trunks

To define a trunk, you specify the ports that you want to be in the trunk.

Important Considerations

- If you have already defined other trunks on your system, you cannot select ports that are part of an existing trunk.
- Devices that you use in a trunking configuration must have the hardware to support the trunking algorithm.
- You can define more than one trunk at a time, which saves having to reboot the system after each trunk definition.
- When you define a trunk, you specify ports and characteristics associated with the trunk (including Gigabit Ethernet flow control). You can specify them all in one `define` operation.
- When you create the trunk, the entire trunk assumes the current port characteristics, such as the FDDI station mode [dual attach station (DAS) or single attach station (SAS)].
- Trunk names can be no longer than 32 characters.
- 3Com recommends that the TCMP state be `enabled`. But devices can operate without TCMP. When TCMP is not in effect on a point-to-point link, its configuration validation is simply absent.
- If your system has more than one media type (for example, FDDI, Fast Ethernet, and Gigabit Ethernet), you are prompted for a media type before you are prompted for the trunk information.
- Trunk names become the port labels when you display information on the trunks.
- All ports in the trunk are set to the selected operating mode (half-duplex or full-duplex).
- Each Gigabit Ethernet module that you install takes up one of the switch's four trunk resources (but does not itself constitute a trunk). If you have two or more Gigabit Ethernet modules, you can trunk them together to free up switch trunk resources. For example, if you install three Gigabit Ethernet modules, the switch allows only one additional trunk. But if you trunk the Gigabit Ethernet modules, the switch supports three additional trunks after you reboot.

  If you add a Gigabit Ethernet module to a switch that has four trunks already defined, the module does not power up, and you receive an error message.

- When you create a VLAN that includes ports that are part of a trunk, specify the *anchor* port (lowest-numbered port) that is associated with the trunk. For example, if ports 1 through 3 are associated with a trunk, specifying port 1 defines the VLAN to include all of the physical ports in the trunk. If you have not defined trunks, simply specify one or more port numbers, or specify `all` to assign all ports to the VLAN interface.
- When you create a trunk that includes ports that are part of a VLAN, those ports are removed from the VLAN. You must modify the VLAN and add the new bridge port to the appropriate VLAN. This situation does not apply to the default VLAN (all ports are part of the default VLAN, including the trunk's anchor port).
- If you upgrade from Version 1.1 and exceed four trunk channels, the Gigabit Ethernet port is not initialized and an error message is posted to the system log. When this situation occurs:
  - The Gigabit Ethernet port MIB returns a `config error` state.
  - The Gigabit Ethernet port is disabled.
  - The console displays `Configuration incompatible please check release notes`.
  - The system error LED lights.
- Doing an `nvData reset` operation erases all previous trunk information.

Modifying Trunks

You can modify a trunk in two ways:

- You can modify a trunk's characteristics (for example, the operating mode or the TCMP state).
- You can add or remove a port from the trunk.

Important Considerations

- You must keep at least one port that you defined in the original trunk. To completely redefine a trunk configuration, remove the trunk and define a new one.
- You cannot modify, add, or remove ports that are part of different trunks from the one you that you are modifying.
- To avoid configuration errors, do not modify FDDI station mode port-pairs when any of the ports in the pair are members of a trunk.
- If you have more than one media type on your system (for example, Fast Ethernet and Gigabit Ethernet), you are prompted for a media type before you are prompted for the trunk information.
- Any changes that you make to the trunk's characteristics take effect immediately and do not interrupt trunk operations. If you add or remove a port, however, you must reboot the system to implement the change.
- In an FDDI trunk:
  - You cannot modify FDDI station mode port pairs when any of the ports in the pair are in a trunk.
  - When you modify the station mode, any FDDI ports that are associated with VLANs or a trunk are removed from the VLAN or trunk.
- If you change an FDDI port pair from SAS to DAS, select the pair using only the lower of the two port numbers, as you do with a trunk anchor port.

- You cannot change some port characteristics within a trunk. For example, in an FDDI trunk, you cannot change a trunked DAS port to a SAS port.

  Here is an example of how to change the FDDI station mode of a trunk:

a Remove the desired trunk.

b Reboot and then change the station mode.

c Reboot and redefine the trunk (and any affected VLANs).

d Reboot.

To avoid configuration errors, do not modify FDDI-station mode port-pairs when any of the ports in the pair are members of a trunk.

Removing Trunks

You can remove one, several, or all trunks using a single `remove` command. This saves having to reboot the system after each trunk remove.

Important Considerations

- If you remove a Gigabit Ethernet module that has trunks defined, NVRAM is not cleaned up, but the trunk ports are available for use by a replacement module of the same type.
- Because each Gigabit Ethernet module uses an internal trunk resource towards the system limit of four, keep in mind how many trunk resources may be used when you remove a trunk. For example, if your system has a trunk with two Gigabit Ethernet ports (which consolidates two trunk resources into one) plus three other trunks, and you then try to untrunk the two Gigabit Ethernet ports, you will exceed the trunk resource limit. The untrunked Gigabit Ethernet ports try to take over two separate trunk resources (for an illegal total of 5), and the system sends a warning message like the following:

  ```
  Unable to remove trunk(s).
  Internal trunk resource limit would be exceeded.
  ```

Standards, Protocols, and Related Reading

The system supports these Ethernet standards:

- **IEEE 802.3** — 10BASE-T Ethernet over unshielded twisted pair (UTP)
- **IEEE 802.3u** — 100BASE-T Fast Ethernet over UTP or fiber
- **IEEE 802.3z** — 1000BASE-SX Gigabit Ethernet over multimode fiber and 1000BASE-LX Gigabit Ethernet over multimode or singlemode fiber

3Com trunking technology interoperates with similar technology from other vendors, including Sun Microsystems and Cisco Systems.

9 VIRTUAL LANS

This chapter provides guidelines and other key information about how to use virtual LANs (VLANs) on your system.

This chapter covers the following topics:

- VLAN Overview
- Key Concepts
- Key Guidelines for Implementation
- VLAN allOpen or allClosed Mode
- Ignore STP Mode
- Port-based VLANs
  - The Default VLAN
  - Static Port-based VLANs
  - Dynamic Port-based VLANs Using GVRP
- Protocol-based VLANs
- Network-based IP VLANs
- Rules of VLAN Operation
- Modifying and Removing VLANs
- Monitoring VLAN Statistics

You can manage VLANs in either of these ways:

- *From the* `bridge vlan` *menu of the Administration Console. See the* Command Reference Guide.
- *From the Bridge VLAN folder of the Web Management software. See the* Web Management User Guide.

VLAN Overview

A *virtual LAN (VLAN)* is a logical grouping that allows end users to communicate as if they were physically connected to a single LAN, independent of the physical configuration of the network. A VLAN is generally considered equivalent to a Layer 2 broadcast domain or a Layer 3 network.

Your system's point of attachment to a given VLAN is called a *VLAN interface*. A VLAN interface exists entirely within a single switch; you control the configuration of the VLAN interfaces on the switch. A VLAN and a VLAN interface are analogous to an IP subnet and an IP interface on a router.

Need for VLANs

If a bridge port in a LAN switch receives a frame with a broadcast, multicast, or unknown destination address, it forwards the data to all bridge ports in the VLAN that is associated with the frame, except the port on which it was received. This process is referred to as bridge *flooding*. As networks grow and the amount and types of traffic increase, bridge flooding may create unnecessary traffic problems that can clog the LAN.

To help control the flow of traffic through a switch and meet the demands of growing networks, vendors have responded by:

- Using customized packet filtering to further control which packets are forwarded through the bridge. These filters can be complex to configure.
- Using more and more routers as broadcast firewalls to divide the network into broadcast domains. As the number of legacy routers increase, latency begins to degrade network performance, administration overhead increases, and operating costs rise.
- Using the Spanning Tree algorithm in switches to control the flow of traffic among LANs (for redundant links). These mechanisms work best only in certain types of LAN topologies.

VLANs provide a high-performance and easy-to-implement alternative to routers for broadcast containment. Using switches with VLANs:

- Each network segment can contain as few as one user (approaching private port LAN switching), while broadcast domains can be as large as 1,000 users or even more.
- VLANs can help network administrators track workstation movements to new locations without manual reconfiguration of IP addresses.
- VLANs can be used to isolate unicast traffic to a single broadcast domain, thereby providing a form of network security.

Benefits You can use VLANs to:

- Reduce the cost of equipment moves, upgrades, and other changes and simplify network administration.
- Create virtual workgroups in which members of the same department or section appear to share the same LAN, with most of the network traffic staying in the same VLAN broadcast domain.
- Help avoid flooding and minimize broadcast and multicast traffic.
- Reduce the need for routing to achieve higher network performance, ease of administration, and reduced costs.
- Control communication among broadcast domains.

Features Your system supports the following VLAN features:

- **Settable modes** — For the entire system, you can establish a less-restrictive VLAN environment with allOpen mode or a more secure VLAN environment with allClosed mode. Using allClosed mode also enables you to use another VLAN feature called Ignore STP mode. The chosen VLAN mode dictates the requirements for the port-based, protocol-based, and network-based VLANs. See "Terminology" for more information about the VLAN modes and Ignore STP Mode.
- **Configurable types of VLANs** — The system allows you to configure different types of VLANs for controlling the flow of traffic through a network:
  - **Port-based VLAN** — Determines VLAN membership using a group of ports. By default, your system provides a special port-based VLAN that contains all ports without tagging. This special VLAN is called the *default VLAN*. It always uses the VLAN ID of 1, the name `Default`, and the protocol type unspecified. See "The Default VLAN" later in this chapter for more information.

    The system also supports both *static* and *dynamic* port-based VLAN configuration if you choose to set it up that way. See "Static Port-based VLANs" and "Dynamic Port-based VLANs Using GVRP" later in this chapter for more information.
  - **Protocol-based VLAN** — Determines VLAN membership using a group of ports that share one or more protocol types. In addition to the user-defined protocol-based VLANs, the system supports a special type of protocol-based VLAN called a *router port IP VLAN*. This type of VLAN, which the system generates when you define an IP interface as a router port IP interface, requires allClosed mode. See "VLANs Created by Router Port IP Interfaces" later in this chapter for more information.
  - **Network-based VLAN** — Determines IP VLAN membership for a group of ports that are configured for IP and a specific network address.

VLAN hierarchy The VLAN type classification is hierarchical: a protocol-based VLAN is a special type of port-based VLAN, and a network-based VLAN is a special type of an IP protocol-based VLAN. This hierarchy allows you to use a combination of VLAN types to group users and traffic types.

You can either configure network-based IP VLANs (IP VLANs with unique Layer 3 IP addresses) or you can define a single VLAN with the protocol type IP and then define multiple IP routing interfaces for that single IP VLAN. See Chapter 11 for more information about defining VLAN-based routing interfaces.

- **Per-port IEEE 802.1Q tagging** — Selecting IEEE 802.1 tagging on a per-port basis dictates that frames be encapsulated and tagged as specified in the IEEE 802.1Q standard. See "Port-based VLANs","Protocol-based VLANs", and "Network-based IP VLANs" later in this chapter for specific information on tagging for the types of VLANs.

Key Concepts

Before you configure VLANs, review the following key concepts.

Related Standards and Protocols

The following standards and protocols apply to the VLANs that you can configure on your system:

- **IEEE 802.1Q** — A proposed standard for VLANs, it is aimed at:
  - Defining an architecture to logically partition bridged LANs and provide services to defined user groups, independent of physical location.
  - Allowing interoperability among multivendor equipment.

  IEEE 802.1Q defines the bridging rules for VLANs, that is, ingress and egress rules, as defined in "Key Concepts" (and described in detail in "Rules of VLAN Operation" later in this chapter).

  The standard also specifies a tag format that embeds explicit VLAN membership information in a 12-bit VLAN ID (VID) that provides 4094 possible VLANs. (Standard IEEE 802.1p uses this same frame format, but also takes advantage of an additional 3 bits for specifying the priority levels used for class of service differentiation.)
- **Generic Attribute Registration Protocol (GARP)** — This protocol is defined in IEEE 802.1p, which is a supplement to the IEEE 802.1D standard. GARP is a Layer 2 transport mechanism that allows switches and end systems to propagate information across the switching domain.
- **GARP VLAN Registration Protocol (GVRP)** — This protocol, which is defined in IEEE 802.1Q, defines dynamic registration of VLANs that use IEEE 802.1Q tagging (the VLAN ID).

VLAN IDs Each VLAN is identified by its VLAN ID (VID). For VLANs that you create, the system keeps track of its used VLAN ID numbers to help you select the next available VLAN ID. Outgoing data frames are tagged per IEEE 802.1Q (which specifies the VID) if tagging is enabled on the transmit port for that VLAN. Tagged IEEE 802.1Q data frames that are received on the system are assigned to the VLAN that corresponds to both the VID contained in the tag and the protocol type.

Be aware of these additional guidelines:

- The default VLAN always uses the reserved VID of 1.
- Before you assign a VID, review the information in Table 15.

Table 15 Assigning ID Numbers to VLANs

VLAN ID Number	Description
VID 1	Reserved for the default VLAN assigned by IEEE and 3Com Corporation
VID 4095	Reserved
VID 2–4094	Numbers that you assign when you create VLANs

- If you rely on dynamic configuration to create a port-based VLAN based on GVRP updates, the VID is the unique IEEE 802.1Q VID.

When you define a router port IP interface, the system automatically creates a router port IP VLAN and assigns it the next available VID. See Chapter 11 for information on router port IP interfaces.

Terminology The following terms apply to VLANs:

- **Default VLAN** — The predefined port-based VLAN interface on your system that always uses VID 1, the protocol type unspecified, and the name Default. The default VLAN also initially includes all of the bridge ports without any tagging, but you can modify the bridge ports and tag status of the default VLAN. If you maintain the default VLAN and you install a new module, the system adds all ports that are associated with the new module to the default VLAN. See "The Default VLAN" for more detailed information.
- **VLAN origin** — Whether the VLAN was created in one of the following ways:
  - **Statically** — The VLAN display shows an origin of static if you define the VLAN.
  - **Dynamically** — The VLAN display shows an origin of GVRP if the system learned the VLAN dynamically through GVRP.
  - **Router** — The VLAN display shows an origin of router if you have defined a router port IP interface on a single bridge port. When you define a router port IP interface, you must place the system in allClosed mode. This removes any allOpen VLANs and re-creates the default VLAN. See Chapter 11 for more information on defining router port IP interfaces.
- **VLAN mode** — A system-wide mode that determines whether data with a unicast MAC address can be forwarded between configured VLANs (allOpen). In allClosed mode, each VLAN has its own address table and data cannot be forwarded between VLANs (although data can still be *routed* between VLANs). The default VLAN mode is allOpen. See "VLAN allOpen or allClosed Mode" for more information.
- **Ignore STP mode** — A per-VLAN mode that determines whether the system ignores the blocking Spanning Tree Protocol (STP) mode for the ports of a designated VLAN. (One instance of STP runs on the system, but you can disable it for each VLAN.) Ignore STP mode is only available in allClosed mode; it is disabled by default. It allows the user to select (for each VLAN) which VLANs ignore STP blocked ports. This mode is typically used for VLANs that have router interfaces that choose to ignore the STP state. It allows routing (or bridging) over a port that is blocked by STP. See "Ignore STP Mode" later in this chapter for more information.

- **Protocol suite** — The protocol family that is associated with a protocol-based VLAN. Protocol-based VLANs can be associated with one or more protocol suites. The protocol suite is unspecified for the default VLAN and all port-based VLANs.
- **Layer 3 address** — The network or subnetwork address that is associated with a network-based IP VLAN.
- **Tagging type** — On a per-port basis, whether there is explicit VLAN membership information (the IEEE 802.1Q header and the VLAN ID or VID) in each frame. You can specify no tagging or IEEE 802.1Q tagging.
- **Port membership** — The bridge ports that you assign to be part of the VLAN.

If you have created trunks, you must specify the anchor port (the lowest-numbered port) port in the trunk when you define the VLAN interface. All bridge ports are initially part of the default VLAN.

- **VLAN name** — The name that you assign to the VLAN. It can contain up to 32 ASCII characters. If the name includes spaces, enclose the name in quotation marks. The default VLAN uses the name Default.
- **Dynamic VLAN configuration** — Using the GARP VLAN Registration Protocol (GVRP), this configuration enables dynamic VLAN configuration of port-based VLANs and dynamic updates of IEEE 802.1Q tagged port-based VLANs.
- **Ingress and egress rules** — *Ingress rules* determine the VLAN to which an incoming frame belongs. If it cannot be assigned to any VLAN, it is assigned to the null VLAN, which contains no ports and has no associated address table in allClosed mode. *Egress rules* determine whether the frame is forwarded, flooded, or filtered, as well as the tag status of the transmitted frame. For more information, see "Rules of VLAN Operation" later in this chapter.

Key Guidelines for Implementation

This section provides a series of guidelines to consider when you use VLANs. The guidelines are organized as follows:

- Network-based VLANs vs. multiple interfaces per VLAN
- VLANs created by router port IP interfaces
- Number of VLANs
- General guidelines

Network-based VLANs vs. Multiple Interfaces per VLAN

You can either configure network-based IP VLANs (IP VLANs with unique Layer 3 IP addresses) or you can define a single VLAN with the protocol type IP and then define multiple IP routing interfaces for that single protocol-based VLAN (an IP VLAN).

If you decide to convert an existing network-based VLAN to a protocol-based VLAN that has multiple interfaces associated with it, use the following procedure:

1. Remove one or more network-based VLANs.
2. Define an IP VLAN or a VLAN that supports IP as one of its protocols.
3. Define multiple IP interfaces (with different IP addresses) to use that IP VLAN.

You can define up to 32 IP interfaces on the system, including IP routing interfaces for static VLANs, router port IP VLANs, or any combination of static VLANs and router port IP VLANs.

If you define multiple interfaces for an IP VLAN, you cannot subsequently modify that IP VLAN to supply Layer 3 address information. If only one routing interface is defined for the IP VLAN, then you can supply Layer 3 address information as long as it matches the Layer 3 information that is specified for the routing interface.

If you use network-based VLANs, you are limited to defining only *one* IP routing interface for that VLAN. When you define an IP routing interface with the interface type *vlan*, the system does not allow you to select a network-based IP VLAN that already has a routing interface defined for it. For more information on IP routing interfaces, see Chapter 11.

VLANs Created by Router Port IP Interfaces

By default, your system uses a routing over bridging model, in which any frame is bridged before it is potentially routed. If you want to define IP routing interfaces that use a routing versus bridging model, however, you can bypass your static VLAN configuration and instead go directly to defining an IP interface on a single router port (a router port IP interface). That process is described in this section.

If you define a router port IP interface, note the following information:

- Defining an IP interface for a router port requires the interface type port. Defining an IP interface for a configured IP VLAN requires you to specify the interface type vlan.
- The IP interface definition procedure for a router port requires that you place the system in allClosed mode. The allClosed mode prevents MAC addresses from being shared between the router port IP VLAN and any other VLANs and enables the router port to ignore Spanning Tree states on the port.
- Once you define the router port IP interface and change the VLAN mode to allClosed, the following events occur:
  - The system deletes all other VLANs and redefines the default VLAN. You must redefine any VLANs that you had configured, keeping in mind that unicast traffic will no longer be forwarded between VLANs. You must define routing interfaces to allow forwarding between VLANs. Also, you cannot specify the bridge port owned by the router port IP interface in any VLAN that you configure or modify.
  - The system creates a special protocol-based VLAN called a router port IP VLAN and assigns to it the next available VID. The VLAN displays identify the origin of a router port IP VLAN as router, as well as the port that is owned by the router port IP interface. You cannot modify or remove a router port IP VLAN, nor can you change its Ignore STP mode (which is always enabled).
- To disable bridging entirely for the router port, remove that port from the default VLAN.

For more information on defining a router port IP interface, see Chapter 11.

Number of VLANs

Your system supports a maximum of 64 VLANs based on a physical limit of 125 VLAN table entries. To determine the number of VLANs of any type that you can have on the system, use the following equation:

Number of VLANs supported =
(125 divided by the number of protocol suites) minus 3

Important Considerations

- When you use the VLAN equation to calculate the number of VLANs that you can have on your system, keep in mind that the equation provides an estimate. Your system may allow additional or fewer VLANs, depending on your configuration, use of protocol suites, and chosen tag style. If, for example, you are using the Release 3.0 VLAN tag style of all ports, this formula generally yields a maximum number of VLANs. If you use the Release 1.2 tag style of taggedVlanPorts, then this formula generally yields a minimum number of VLANs.
- The number of allowable VLANs includes the default VLAN.

Determining the Number of Protocol Suites

To perform the calculation, first determine the total number of protocol suites used on your system. Use the following guidelines:

- IP counts as one protocol suite for IP VLANs.
- AppleTalk counts as one protocol suite for AppleTalk VLANs.
- Generic IPX, which uses all four IPX types, counts as four protocol suites. (Each IPX type alone counts as one.)
- DECnet counts as one protocol suite for DECnet VLANs.
- The unspecified type of protocol suite counts as one for the default VLAN or port-based VLANs. (Even if you have *only* the unspecified protocol suite on the system, the limit is still 64 VLANs.)
- If you are using GVRP (for dynamic port-based VLANs), use the type unspecified in the VLAN formula

Remember to include the unspecified type for the default VLAN, even if you have removed the default VLAN and do not have another VLAN defined with the unspecified protocol type.

In addition to the limit on the number of VLANs, you are limited to 15 different protocols that can be implemented by the protocol suites on the system. See Table 19 later in this chapter for a list of supported protocol suites and the number of protocols within each suite.

VLAN Equation Examples

Example 1 You have 7 protocol suites on the system (IP, AppleTalk, unspecified for the default VLAN, and generic IPX, which counts as 4 protocol suites):

(125 / 7) – 3 = 14

In this configuration, the system supports a minimum of 14 VLANs. Per Table 19, these 7 protocol suites use 10 protocols: 3 IP, 2 AppleTalk, 1 unspecified, and 4 generic IPX.

Example 2 You have 5 protocol suites on the system (IP, unspecified, AppleTalk, IPX 802.2 Sub-Network Access Protocol [SNAP], and IPX 802.3 Raw):

(125 / 5) – 3 = 22

In this configuration, the system supports a minimum of 22 VLANs. Per Table 19, these 5 protocol suites use 7 protocols: 3 IP, 1 unspecified, 2 AppleTalk, 1 IPX 802.2 SNAP, and 0 IPX 802.3 Raw, because it does not use an Ethernet protocol type.

If you are upgrading your system from Release 1.2 and the VLAN resource limit is reached during a power up with a serial port console connection, use the Administration Console option `bridge vlan vlanAwareMode` *to change the VLAN aware mode to taggedVlanPorts. See "VLAN Aware Mode" later in this chapter for more information.*

General Guidelines

- The VLAN mode of allOpen or allClosed applies to *all* VLANs associated with the system (static, dynamic, or router port). Configure the VLAN mode *before* you define any static VLANs. (As part of the configuration procedures for a router port IP interface, you must place the system in allClosed mode; see Chapter 11.)

If you change the VLAN mode after you have defined VLANs, the system deletes all configured VLANs and redefines the default VLAN. See "Modifying the VLAN Mode" later in this chapter.

- If you configure the system for allClosed mode, you can enable Ignore STP mode on any VLAN. You can also disable STP on a any port for either allOpen or allClosed mode by using a bridge port option. (Use `bridge port stpState` on the Administration Console.) See Chapter 7 for bridging information. Also see "Ignore STP Mode" later in this chapter.
- To take advantage of GVRP for dynamic configuration or dynamic updates of port-based VLANs, verify that GVRP is enabled as both a bridge-wide and a bridge-port parameter. See Chapter 9 for information about bridging parameters. See "Dynamic Port-based VLANs Using GVRP" for information about GVRP.
- You can configure overlapping VLANs if they have some distinguishing characteristic. For example, a bridge port can be shared by multiple VLANs as long as the shared port has a distinguishing characteristic for the shared port, such as protocol type or tagging type. In allClosed mode, you must tag overlapped ports of any network-based VLANs. See "Network-based IP VLANs" later in this chapter.
- Per-port tagging requirements depend on whether the hosts connected to the port are configured for IEEE 802.1Q tagging. Per-port tagging is also required to differentiate between overlapped ports of the same protocol type and between overlapped IP Layer 3 VLANs in allClosed mode.
- Consider maintaining the system's default VLAN. The default VLAN preserves the flooding of unspecified traffic, since it initially contains all of the system's bridge ports, with unspecified protocol information and no tagging.

- To establish routing between static VLANs and configure a VLAN interface to support one or more routing protocols, configure the VLAN for the protocols *before* you configure a routing interface. For protocols other than IP, the system does not define the routing interface for a protocol if a VLAN for that protocol does not exist.

  If you define an IP interface and specify vlan as the interface type, the system does not define the IP routing interface unless you have an IP VLAN configured. See the appropriate routing chapter for an overview of your routing options and guidelines. See Chapter 11 for information on defining either an IP router interface (for a static IP VLAN) or a router port IP interface.

- If you plan to use trunks, define the appropriate trunks *before* you define your VLANs. (If you define a VLAN with certain ports and subsequently configure some of those ports to be part of a trunk, the system removes those ports from the VLAN and places them in the default VLAN.) See "Trunking and the Default VLAN" for more information. When you define a VLAN that includes trunk ports, you must specify the trunk's anchor port (lowest-numbered port). For trunking information, see Chapter 8.

- When the system receives a frame, the frame is assigned to a VLAN using the ingress rules. See "Ingress Rules" later in this chapter. When the system transmits the frame, it determines the tag status (none or IEEE 802.1Q tagging) by referring to the tag status of the transmit port in the frame's assigned VLAN. In allOpen mode, if a frame is transmitted on a port that does not belong to the assigned VLAN, the frame is transmitted untagged.

VLAN allOpen or allClosed Mode

You can select allOpen or allClosed as the VLAN mode for your entire system. The default is allOpen.

3Com's use of the term "allOpen" is equivalent to the IEEE Standard 802.1Q term "Shared VLAN Learning" (SVL). The term "allClosed" is equivalent to the IEEE 802.1Q term "Independent VLAN Learning" (IVL). 3Com imposes the restriction that you must choose one VLAN mode for the entire system. More complex logic for assigning SVL and IVL to individual ports is described in the IEEE 802.1Q standard.

Important Considerations

- In general, select your VLAN mode before you define your VLANs (VLANs with an origin of static).
- As part of the configuration procedures for a router port IP interface, you must place the system in allClosed mode. Once you define a router port IP interface (and the system creates the router port VLAN), you cannot change the VLAN mode until you delete the router port IP interface.
- Select the VLAN mode as follows:
  - **allOpen** — Use this less restrictive mode if you have no security issues about the forwarding of data between VLANs. The allOpen mode is the default VLAN mode for all VLANs that you create. It permits data with a unicast MAC address to be forwarded between VLANs. For example, data received on IP VLAN 2 with a destination of IP VLAN 3 is forwarded there.

    The allOpen mode implies that the system uses a single bridge address table for all of the VLANs on the system (the default configuration).
  - **allClosed** — Use this more restrictive mode if you are concerned about security between VLANs. Data cannot be forwarded between VLANs (although data can still be routed between VLANs). The allClosed mode implies that each VLAN that you create has its own address table. Router port IP interfaces require allClosed mode.

- If you are using allClosed mode and STP on the system (with multiple routes to a destination), you can also specify a mode called *Ignore STP mode* to disable STP blocking for a specified static VLAN. (Although each VLAN has its own address table, there can be only one instance of STP on the system.) See "Ignore STP Mode" for information on this mode. To disable STP blocking on any port with allOpen or allClosed VLANs, use the `bridge port stpState` option on the Administration Console. See Chapter 7 for bridging information.
- Your choice of the VLAN mode affects how you manipulate bridge port addresses (via the Console or the Web). For example:
  - If you select allClosed mode, you *must* specify a VLAN interface index to identify the appropriate bridge address table.
  - If you select allOpen mode (the default), the entire system has only one address table, so you can manipulate the bridge port addresses without specifying a VLAN interface index.

Modifying the VLAN Mode

To change your VLAN mode, perform these procedures:

1. Delete all routing interfaces (including router port IP interfaces) that you have configured on the system. You cannot change the mode if you have router interfaces defined on the system.
2. Using your configuration tool (for example, the Administration Console or the Web Management applications), modify the VLAN mode to specify the new VLAN mode.

   When you change the mode, the system deletes all of your existing configured VLANs and reverts to the default VLAN.
3. Reconfigure your VLANs and redefine your routing interfaces.

   For the specific commands for these procedures, see the *Command Reference Guide*.

Mode Requirements

Table 16 shows the requirements for defining static VLANs in allOpen and allClosed mode.

Table 16 Mode Requirements for Static VLANs

Type of Static VLAN	Requirements
Port-based	For *nonoverlapped* port-based VLANs: ■ Protocol type: unspecified ■ Separate member ports. That is, each port-based VLAN owns a different set of ports. For *overlapped* port-based VLANs: ■ Protocol type: unspecified ■ IEEE 802.1Q tagging for shared ports. That is, the shared ports can employ a tagging mode of none in only one VLAN; shared ports in all other VLANs must use IEEE 802.1Q tagging.
Protocol-based	For *nonoverlapped* protocol-based VLANs: ■ Either the protocol type or the member ports are unique per VLAN For *overlapped* protocol-based VLANs (multiple VLANs of the same protocol type that share ports): ■ IEEE 802.1Q tagging for shared ports. That is, the shared ports can employ a tagging mode of none in only one of the same protocol type VLANs; shared ports in all other VLANs of the same protocol type must use IEEE 802.1Q tagging.
Network-based (IP VLAN only)	■ A Layer 3 address that is unique per network-based VLAN ■ For *allOpen mode*, no tagging restrictions on the shared ports ■ For *allClosed mode*, IEEE 802.1Q tagging for shared ports. That is, the shared ports can employ a tagging mode of none in only one of the network-based VLANs; shared ports in all other network-based VLANs must use IEEE 802.1Q tagging.

Ignore STP Mode

When you use allClosed VLAN mode on your system, you can enable the system to ignore the Spanning Tree Protocol (STP) mode on a per-VLAN basis, that is, to ignore STP blocked ports for static protocol-based VLANs associated with routing interfaces. (When STP detects multiple paths to a destination, it blocks all but one of the paths.)

If you have configured router port IP interfaces on your system (so that the system generates router port VLANs owned by the router IP interfaces), ignore STP mode is automatically enabled and you cannot disable it.

Important Considerations

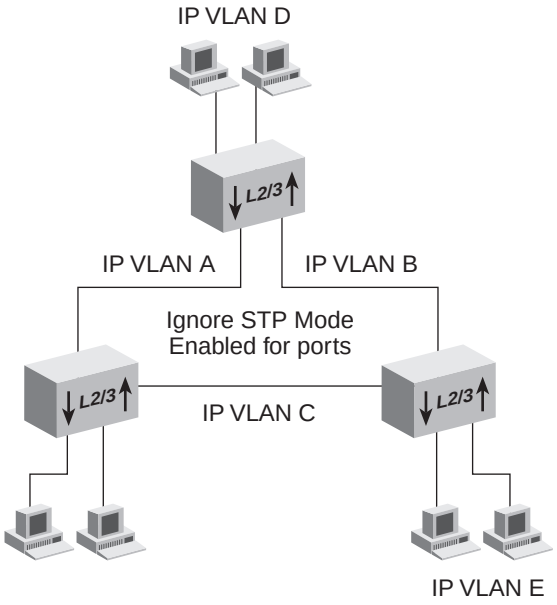
- Ignore STP mode is disabled by default for static VLANs.
- You can use this mode *only* when the system is in allClosed mode.
- Ignore STP mode is useful when you have redundant router connections between systems that have STP enabled. In this situation, if you want to create multiple VLANs and use one VLAN for routing, you can configure your system to ignore the STP blocking mode for that VLAN. This setting avoids disruptions to routing connectivity based on the STP state.
- To disable STP blocking on a *per-port* basis with allOpen or allClosed VLANs, use the bridging option (`bridge port stpState` on the Administration Console). See Chapter 7 for bridging information.

Ignore STP mode affects bridging as well as routing. If you have STP enabled on the system and you have redundant bridged paths between systems with different VLANs, STP blocks one of the paths unless you enable Ignore STP mode. See Figure 28 later in this chapter for an example of redundant paths between systems that have different port-based VLANs.

Example of Ignore STP Mode

Figure 27 shows two paths available if a workstation associated with IP VLAN E wants to communicate with a server associated with IP VLAN D. STP blocks the routed as well as bridged traffic for the one path unless you enable Ignore STP Mode for the routed IP VLANs. With the blocking removed for IP routed traffic, the best path is used.

Figure 27 Using Ignore STP Mode



VLAN Aware Mode

VLAN aware mode accommodates the difference in VLAN resource usage as well as tagged-frame ingress rules between Release 1.2 and Release 3.0 of the system software. For more information on ingress rules, see "Rules of VLAN Operation" later in this chapter. (The Release 1.2 ingress rules in allOpen mode mandated that incoming tagged frames assigned to one of the configured VLANs if the VID of the frame matched that of the VLAN *and* if a port in that VLAN were tagged.)

The VLAN aware mode, which you set with the Administration Console option `bridge vlan vlanAwareMode`, reflects the difference in VLAN resource usage and modes of tagging as follows:

- At Release 1.2, all bridge ports were *not* VLAN aware (tagging aware) unless they were assigned to a VLAN that has one or more tagged ports.
- At Release 2.0 and later, all bridge ports become VLAN aware after a software update or after an NV data reset and do not have to be explicitly tagged in order to forward tagged frames.

This difference in resource usage and modes of tagging has the following impact: After you upgrade the system from 1.2 to 3.0, the release uses VLAN resources differently than it did at Release 1.2 and may cause a change in the total number of allowable VLANs.

VLAN aware mode is currently supported only through the Administration Console, not through Web Management or SNMP.

Initial installation of Release 3.0 provides a default VLAN aware mode of allPorts, which is consistent with the Release 3.0 ingress rules and resource allocation.

If you upgrade your system from Release 1.2 to a later release and the VLAN resource limit is reached during a power up with a serial port console connection, the system displays an error message similar to the following one to identify the index of the VLAN that it was unable to create:

```
Could not create VLAN xx - Internal resource threshold
exceeded
```

In this situation, the system removes all bridge ports from the VLAN that it could not restore from nonvolatile (NV) data, although it does maintain the previously stored NV data. To restore your VLANs after you see the resource error message, use the `bridge vlan vlanAwareMode` option and then set the VLAN aware mode to taggedVlanPorts. If VLANs are already defined, the system prompts you to reboot to put the new mode into effect.

If you do not see the VLAN internal resource error message, maintain the default VLAN aware mode of allPorts. In this case, the system can accommodate the number of Release 1.2 VLANs, but it now uses different ingress rules for tagged frames.

The Administration Console options `bridge vlan summary` and `bridge vlan detail` display the current VLAN aware mode after the VLAN mode (allOpen or allClosed).

Port-based VLANs

Port-based VLANs logically group together one or more bridge ports on the system and use the generic protocol type unspecified. Each arbitrary collection of bridge ports is designated as a *VLAN interface.* This VLAN interface belongs to a given VLAN. Flooding of all frames that are received on bridge ports in a VLAN interface is constrained to that VLAN interface.

Your system supports the following types of port-based VLANs:

- The default VLAN, a special VLAN predefined on the system
- Static port-based VLANs that you create
- Dynamic port-based VLANs created using GVRP

An alternative to port-based VLANs is packet filtering using port groups, as described in Chapter 10.

The Default VLAN

The system predefines a port-based VLAN to initially include all of the system's bridge ports without any tagging. For example, if you have four 10/100 Ethernet modules (24 bridge ports) installed on your system, the default VLAN initially contains all 24 ports.

The default VLAN always uses the VID of 1, the name Default, and the protocol type unspecified. No other VLAN than the default VLAN can use a VID of 1.

The default VLAN is the flood domain in either of these cases:

- The system receives data for a protocol that is not supported by any VLAN in the system.
- The system receives data for a protocol that is supported by defined VLANs, but these VLANs do not contain the port receiving the data.

See "Rules of VLAN Operation" later in this chapter.

Modifying the Default VLAN

The default VLAN is always associated with the VID of 1, the unspecified protocol type, and the name `Default`. Initially, the default VLAN is also associated with all ports and no tagging. Keeping the default VLAN intact ensures that the system accommodates the addition of a module by automatically adding the new module's bridge ports to the default VLAN. If necessary, the system also renumbers its ports when you add the module.

If necessary, you can modify (or remove) the default VLAN on the system. For example, you may want to modify the default VLAN to remove certain ports. Such a change does not prevent the system from adding a new module's bridge ports to the default VLAN.

However, the following changes *do* prevent the system from adding a new module's bridge ports to the default VLAN:

- If you modify the default VLAN to remove all ports
- If you remove the default VLAN completely. Even if you subsequently redefine the default VLAN, the system will not add bridge ports to the newly defined default VLAN.
- If you modify the default VLAN to tag a port

To ensure that data can be forwarded, associate a bridge port with a VLAN. This association is mandatory in allClosed mode. If you remove the default VLAN (and you do not have other VLANs defined for the system), your ports may not forward data until you create a VLAN for them.

Trunking and the Default VLAN

Another benefit of maintaining the default VLAN (with any number of ports) involves trunking. 3Com strongly recommends that you define your trunks *before* you define your VLANs.

Trunking with the default VLAN intact

Trunking actions affect the default VLAN in the following ways:

- If you have only the default VLAN with all ports and you define a trunk (or subsequently remove a trunk), the ports listed in the VLAN summary for the default VLAN do not change. In this case, maintaining the default VLAN with all ports ensures that trunks can come and go without causing any VLAN changes.
- If you have the default VLAN as well as additional VLANs and you subsequently define a trunk for ports in one of the other VLANs, the system removes those ports from that other VLAN and places them in the default VLAN. The same action occurs when you remove an existing trunk from a VLAN that you created after the trunk. For example:

Ports Before Action	Trunking Action	Ports After Action
default VLAN: ports 1-4 ipvlan1: ports 5-11	Define a trunk with ports 7,8	default VLAN: ports 1-4, 7-8 ipvlan1: ports 5-6, 9-11

- If you have the default VLAN as well as other VLANs and you subsequently modify an existing trunk that has ports in one of the VLANs, any port removed from the trunk is removed from the VLAN and placed in the default VLAN. For example:

Ports Before Action	Trunking Action	Ports After Action
default VLAN: ports 1-4 ipvlan1: ports 5-11 (ports 5-8 are trunk ports)	Modify existing trunk to have ports 6-8 (remove port 5, the anchor port)	default VLAN: ports 1-5 ipvlan1: ports 6-11 (port 6 is new anchor port)

Trunking with the default VLAN removed

If you remove the default VLAN, the system has nowhere to return ports altered by trunking, as discussed in these examples:

- If you have VLANs (but no default VLAN) and you then define a trunk for ports in one of the VLANs, those ports are removed from that VLAN and are not assigned to any other VLAN. If you later remove the trunk, these ports are not reassigned to the VLAN; they no longer have a VLAN associated with them. For example:

Ports Before Action	Trunking Action	Ports After Action
ipvlan1: ports 1-11	Define trunk with ports 5-8	ipvlan1: ports 1-4, 9-11

- If you have VLANs (but no default VLAN) and you modify an existing trunk that has ports in one VLAN, any port that is removed from the trunk is removed from the VLAN and no longer has a VLAN. For example:

Ports Before Action	Trunking Action	Ports After Action
ipvlan1: ports 1-11 (ports 5-8 are trunk ports)	Modify existing trunk to have ports 6-8 (remove port 5, the anchor port)	ipvlan1: ports 1-4, 6-11 (port 6 is new anchor port)

See Chapter 8 for information on using trunks.

Static Port-based VLANs

You can explicitly configure port-based VLAN interfaces instead of relying on GVRP to dynamically create port-based VLAN interfaces.

Important Considerations

When you create this type of VLAN interface, review these guidelines:

- When you select the bridge ports that you want to be part of the VLAN, the bridge ports that you specify as part of the VLAN are the same as your physical ports, unless you have created trunks or unless you have DAS ports defined on an FDDI module.
- If you define trunks, a single bridge port called the *anchor port* (the lowest-numbered port in the trunk) represents all ports that are part of the trunk. Only the anchor bridge port for the trunk, not the other bridge ports in the trunk, is selectable when you are creating VLANs. For more information, see Chapter 8.

- If you define FDDI DAS ports, select the lowest-numbered port in the DAS pair when you define the ports in the VLAN. The higher-numbered port in the DAS pair is not selectable. See Chapter 6.
- Decide whether you want the ports that you are specifying for the VLAN interface to be shared by any other VLAN interface on the system. Shared ports produce *overlapped* VLANs; ports that are not shared produce *nonoverlapped* VLANs.
- The per-port tagging options are IEEE 802.1Q tagging or no tagging. The IEEE 802.1Q tagging option embeds explicit VLAN membership information in each frame.
- Overlapped VLANs require tagging; that is, two port-based VLAN interfaces may contain the same bridge port if one of the VLAN interfaces defines the shared port to use IEEE 802.1Q tagging. This rule is true for either allOpen or allClosed mode. For example, a shared bridge port is set to tagging none for one VLAN and IEEE 802.1Q tagging for the other VLAN, or IEEE 802.1Q tagging for each VLAN.
- Port-based VLANs use the protocol type unspecified.
- To define a port-based VLAN interface, specify this information:
  - A VID in the range 2 through 4094, or accept the next available VID.
  - The bridge ports that are part of the VLAN. If you have trunk ports, specify the anchor port for the trunk. For FDDI DAS ports, specify the lowest-numbered port in the DAS pair.
  - The protocol type unspecified.
  - Tag status (none or IEEE 802.1Q).
  - The unique name of the VLAN interface.

Example 1: Nonoverlapped VLANs

Figure 28 shows two systems that have nonoverlapping port-based VLANs and no port tagging. Ports 1 through 4 on Device1 make up the VLAN called unspecA, while ports 5 through 8 make up unspecB. All frames that are received on a port are assigned to the VLAN that is associated with that port. For instance, all frames that are received on port 2 in unspecA are assigned to unspecA, regardless of the data contained in the frames. After an incoming frame is assigned to a VLAN, the frame is forwarded, filtered, or flooded within its VLAN, based on the standard bridging rules.

This situation causes different behavior for allOpen versus allClosed VLANs. For example, for allClosed VLANs, if a frame is received on a port in unspecA with a destination address that is known in the address table of unspecB, the frame is flooded throughout unspecA because it has an unknown address for unspecA. For allOpen VLANs, there is one address table; therefore; the frame is forwarded to the port that corresponds to the known destination address. However, if the transmit port is not a member port of unspecA, the frame is transmitted untagged, regardless of that port's tag status on unspecB.

In Figure 28, if STP is enabled, STP blocks one of the paths unless you enable Ignore STP mode. See "Ignore STP Mode" earlier in this chapter for more information.

Figure 28 Port-based VLANs Without Overlapped Ports

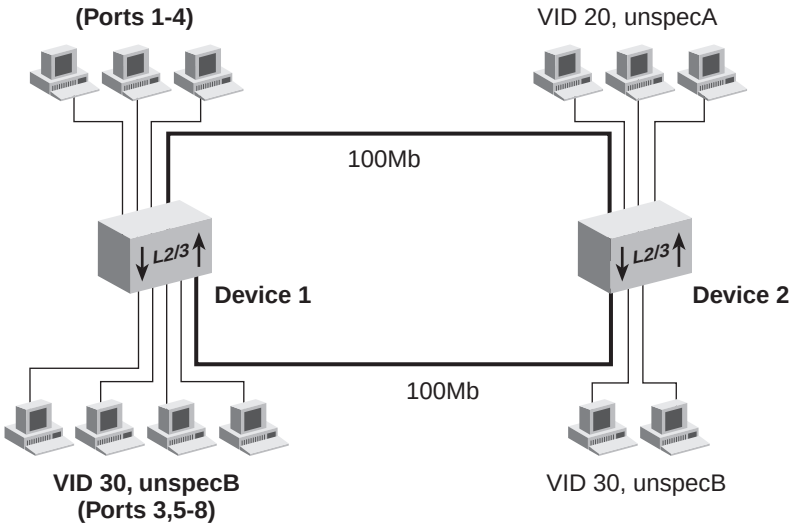


Table 17 shows the information that can be used to configure these VLANs *without* overlapped ports on Device 1 (the device on the left):

Table 17 Port-based VLAN Definitions Without Overlapped Ports for Device 1

unspecA	**unspecB**
VLAN Index 2	VLAN Index 3
VID 10	VID 15
Bridge ports 1-4	Bridge ports *5-8*
Protocol type unspecified	Protocol type unspecified
Per-port tagging: ■ Ports 1-4 — *none*	Per-port tagging: ■ Ports 5-8 — *none*
VLAN name unspecA	VLAN name unspecB

Example 2: Overlapped VLANs

Figure 29 shows port-based VLANs that *overlap* on bridge port *3*.

Figure 29 Port-based VLANs with Overlapped Ports

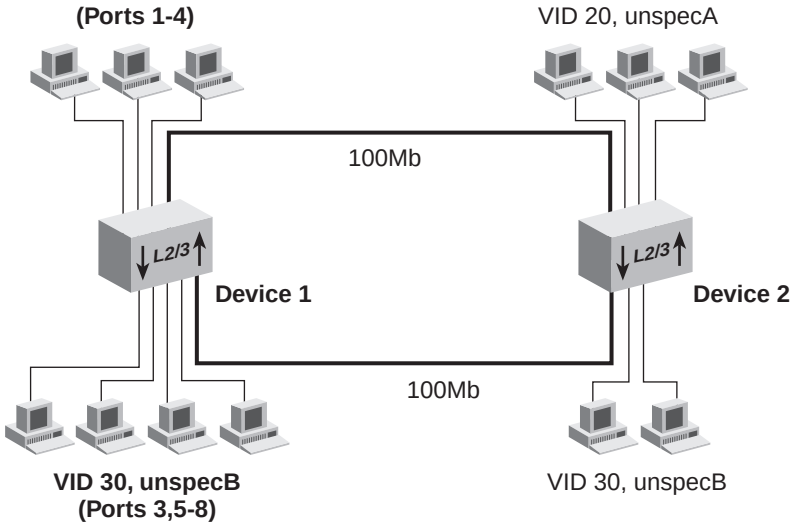


Table 18 shows the information that you use to configure these VLANs *with* overlapped ports on Device 1:

Table 18 Port-based VLAN Definitions with Overlapped Ports for Device 1

unspecA	**unspecB**
VLAN Index 2	VLAN Index 3
VID 20	VID 30
Bridge ports *1-4*	Bridge ports *3, 5-8*
Protocol type unspecified	Protocol type unspecified
Per-port tagging: Ports 1-4 — *none*	Per-port tagging: ■ Port 3 — *IEEE 802.1Q* ■ Port 5 — *IEEE 802.1Q* ■ Ports 6-8 — *none*
VLAN name unspecA	VLAN name unspecB

If you plan for your VLAN to include trunk ports, specify the anchor port (lowest-numbered port) associated with the trunk. For example, if ports 5 through 8 in unspecB were associated with a trunk, you specify only bridge port 5 to define the VLAN to include all of the physical ports in the trunk (ports 5 through 8). The IEEE 802.1 Q tagging applies to all ports in the trunk.

Dynamic Port-based VLANs Using GVRP

GARP VLAN Registration Protocol (GVRP) can help you simplify the management of VLAN configurations in your larger networks.

GVRP allows the system to:

- Dynamically create a port-based VLAN (unspecified protocol) with a specific VID and a specific port, based on updates from GVRP-enabled devices.
- Learn, on a port-by-port basis, about GVRP updates to an existing port-based VLAN with that VID and IEEE 802.1Q tagging.
- Send dynamic GVRP updates about its existing port-based VLANs.

GVRP enables your system to advertise its manually configured IEEE 802.1Q VLANs to other devices supporting GVRP. Because the VLANs are advertised, GVRP-aware devices in the core of the network need no manual configuration to pass IEEE 802.1Q frames to the proper destination. The method of VLAN advertisement used by all GVRP-capable switches involves protocol data units (PDUs), similar to the method used by STP. GVRP-capable devices send their updates to a well-known multicast address to which all GVRP-capable devices listen for information changes.

Enabling GVRP lets the system dynamically adjust active network topologies in response to configuration changes in one or more VLANs. GVRP then advertises VLAN changes on each bridge to all other GVRP bridges in the network.

Important Considerations

To use GVRP, consider the following:

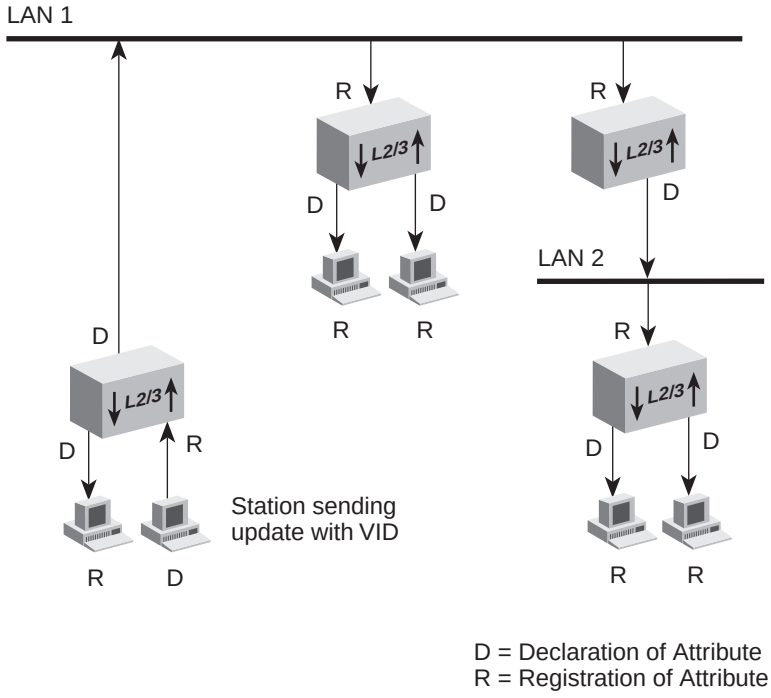
- To take advantage of dynamic IEEE 802.1Q VLAN configuration, enable GVRP as an entire bridge state and then as an individual bridge port state for the appropriate ports. See Chapter 7. By default, GVRP is disabled as both a bridge state and a bridge port state. If GVRP is enabled, the VLAN origin for a port-based VLAN is dynamic (with GVRP). When GVRP is disabled, the VLAN origin is either static (traditional static VLAN without GVRP) or router (router port).
- In a GVRP environment, devices must be GVRP-enabled (that is, support GVRP). These devices may be end stations with 3Com's Dynamic*Access*® software or other switches that explicitly enable GVRP.
- VLANs created dynamically with GVRP exist only as long as a GVRP-enabled device is sending updates. If the devices no longer send updates, or GVRP is disabled, or the system is rebooted, all dynamic VLANs are removed.
- GVRP updates are not sent out on any blocked STP ports. GVRP operates only on ports that are in the STP forwarding state. If GVRP is enabled, a port that changes to the STP forwarding state begins to participate in GVRP. A port that changes to an STP state other than forwarding no longer participates in GVRP.

- The VLAN topologies that GVRP learns are treated differently from VLANs that are statically configured. GVRP's dynamic updates are not saved in NVRAM, while static updates are saved in NVRAM. When GVRP is disabled, the system deletes *all* VLAN interfaces that were learned through GVRP and leaves unchanged all VLANs that were configured through the Administration Console, SNMP, or the Web Management software.
- GVRP manages the active topology, not nontopological data such as VLAN protocols. If you need to classify and analyze packets by VLAN protocols, you manually configure protocol-based VLANs. But if the system needs to know only how to reach a given VLAN, then GVRP provides all necessary information.
- A GVRP-created VLAN is useful in situations where only Layer 2 switching needs to be performed for that VLAN. (Routing between a GVRP-created VLAN and another VLAN can be performed with an external router.) Because GVRP-created VLANs are assigned the unspecified protocol type, router interfaces cannot be assigned to them. Therefore, all communication within a GVRP-created VLAN is constrained to that VLAN in allClosed mode; in allOpen mode, only unicast frames with a known destination address can be transmitted to another VLAN.

Example: GVRP

Figure 30 shows how a GVRP update (with the VID) sent from one end station is propagated throughout the network.

Figure 30 Sample Configuration Using GVRP



Protocol-based VLANs

Protocol-based VLANs enable you to use protocol type and bridge ports as the distinguishing characteristics for your VLANs. When you select a protocol such as IP, you do so based on the guidelines in this section.

Important Considerations

Before you create this type of VLAN interface, review these guidelines:

- If you plan to use the VLAN for *bridging* purposes, select one or more protocols per VLAN. Select them one protocol at a time.
- If you plan to use the VLAN for *routing*, select one or more protocols per VLAN, one protocol at a time, and subsequently define a routing interface for each routable protocol that is associated with the VLAN.
- The system supports routing for three protocols: IP, IPX, and AppleTalk.
- To define a protocol-based VLAN interface, specify this information:
  - The VID of your choice (except 1 or any VID already assigned), or accept the next available VID.
  - The bridge ports that are part of the VLAN interface. (If you have trunk ports, specify the anchor port for the trunk.)
  - The protocol for the specified ports in the VLAN.
  - Tag status (none or IEEEE 802.1Q). IEEE 802.1Q tagging must be selected for ports that overlap on both port and protocol (for example, if two IPX VLANs overlap on port 3).
  - The name you want to assign to this VLAN interface.
- If you use IP as the protocol and also specify a Layer 3 address, the protocol-based VLAN becomes a *network-based VLAN*.

You can either configure network-based IP VLANs (IP VLANs with unique Layer 3 IP addresses) or you can define a single protocol-based VLAN with the protocol type IP and then define multiple IP routing interfaces for that VLAN. For more information on network-based VLANs, see "Network-based IP VLANs" later in this chapter. For more information about IP interfaces, see in Chapter 11.

Selecting a Protocol Suite

The protocol suite describes which protocol entities can comprise a protocol-based VLAN. For example, the system's VLANs support the IP protocol suite, which has three protocol entities (IP, ARP, and RARP).

Table 19 lists the protocol suites that the system supports, as well as the number of protocols that are associated with each protocol suite.

Table 19 Supported Protocol Suites for VLAN Configuration

Protocol Suite	Protocol Entities	Number of Protocol Suites (PVIDs)	Number of Protocols in Suite
IP	IP, ARP, RARP (Ethernet Version 2, SNAP PID)	1	3
Novell IPX	IPX supports these IPX types:	4	4
	■ IPX - type II (Ethernet Version 2)	1	1
	■ IPX - 802.2 LLC (DSAP/SSAP value 0xE0 hex)	1	0*
	■ IPX - 802.3 Raw (DSAP/SSAP value 0xFF hex)	1	0*
	■ IPX - 802.2-SNAP (DSAP/SSAP value 0xAA hex)	1	1
AppleTalk	DDP, AARP (Ethernet Version 2, SNAP PID)	1	2
Xerox XNS	XNS IDP, XNS Address Translation, XNS Compatibility (Ethernet Version 2, SNAP PID)	1	3
DECnet	DEC MOP, DEC MOP Remote Console, DEC DecNet Phase IV, DEC LAT, DEC LAVC (Ethernet Version 2, SNAP PID)	1	5
SNA	SNA Services over Ethernet (Ethernet Version 2 and DSAP/SSAP values 0x04 and 0x05 hexadecimal)	2	1
Banyan VINES	Banyan (Ethernet Version 2, DSAP/SSAP value 0xBC hexadecimal, SNAP PID)	1	1
X25	X.25 Layer 3 (Ethernet Version 2)	1	1
NetBIOS	NetBIOS (DSAP/SSAP value 0xF0 hexadecimal)	1	0*
Default	Default (all protocol types)	1	1
(unspecified)	No protocol types		

* This protocol does not use an Ethernet protocol type.

The system imposes two important limits regarding the number of VLANs and the number of protocols:

- **Number of VLANs supported on the system** — To determine the minimum number of VLANs that the system can support, use the equation described in "Number of VLANs" earlier in this chapter. The system supports a maximum of 64 VLANs.
- **Maximum number of protocols** — Use the value 15 as the maximum number of protocols that can be implemented on the system. A protocol suite that is used in more than one VLAN is counted only once toward the maximum number of protocols. For example, the DECnet protocol suite uses 5 of the available 15 protocols, regardless of the number of VLANs that use DECnet.

Example: Protocol-based VLANs for Bridging

Figure 31 is an example of a VLAN bridging configuration that contains three protocol-based VLANs (two IP and one IPX) that overlap on an FDDI link (port 1 in each VLAN). (You can configure the link to be part of a trunk, as described in Chapter 8.) The end stations and servers are on 100Mbps ports, with traffic segregated by protocol. They are aggregated over the FDDI link.

Figure 31 Example of a Bridging Protocol-based VLAN Configuration

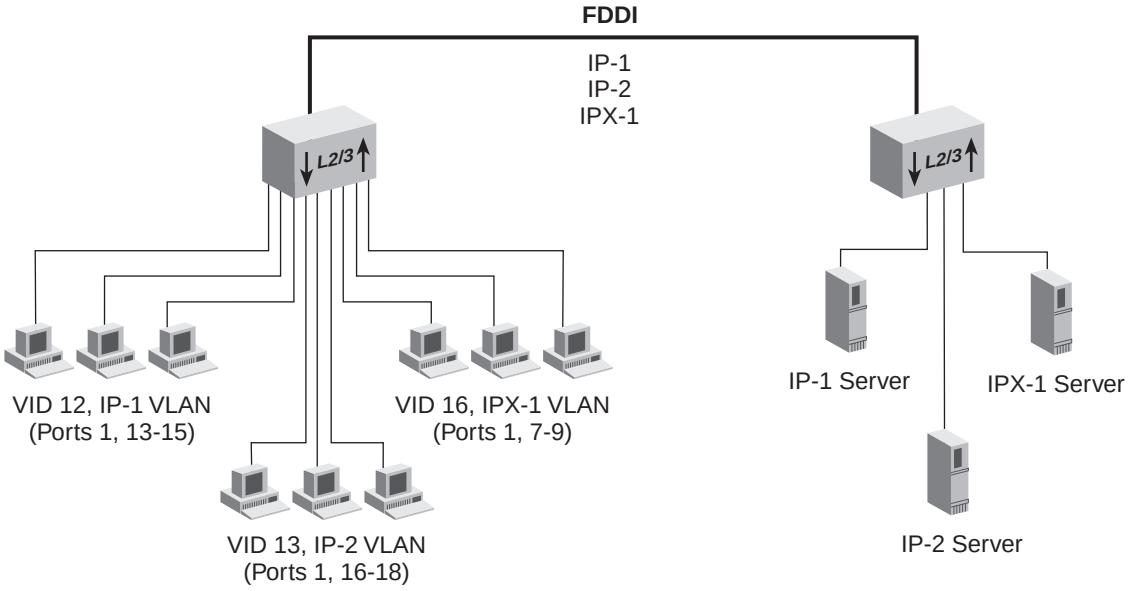


Table 20 shows the information that can be used to configure these VLANs on Device 1 (the device on the left):

Table 20 Sample Protocol-based VLAN Definitions

IP-1 VLAN	**IP-2 VLAN**	**IPX-1 VLAN**
VLAN Index 2	VLAN Index 3	VLAN Index 4
VID 12	VID 13	VID 16
Bridge ports *1, 13-15*	Bridge ports *1, 16-18*	Bridge Ports *1, 7-9*
Protocol type *IP*	Protocol type *IP*	Protocol type *IPX-802.3*
No Layer 3 address	No Layer 3 address	No Layer 3 address
Per-port tagging: ■ Port 1 — *IEEE 802.1Q* ■ Ports 13-15 — *none*	Per-port tagging: ■ Port 1 — *IEEE 802.1Q* ■ Ports 16-18 — *none*	Per-port tagging: ■ Port 1 — *none* ■ Ports 7-9 — *none*
VLAN name *IP-1*	VLAN name *IP-2*	VLAN name *IPX-1*

Establishing Routing Between VLANs

Your system supports routing using IP, IPX, and AppleTalk VLANs. If VLANs are configured for other routable network layer protocols, the VLANs can communicate between those protocols only through an external router.

The system's routing over bridging model allows you to configure routing protocol interfaces based on a static VLAN defined for one or more protocols. You must first define a VLAN to support one or more protocols and then assign a routing interface for each protocol associated with the VLAN. (You can also opt to use a routing versus bridging model by defining a router port IP interface, as defined in Chapter 11.)

Because the system supports router port IP interfaces as well as IP router interfaces for static VLANs, you must specify the interface type vlan when you define an IP interface for a static VLAN.

Important Considerations

To create an IP interface that can route through a static VLAN, you must:

1. Create a protocol-based IP VLAN for a group of bridge ports. If the VLAN overlaps with another VLAN at all, define it in accordance with the requirements of your VLAN mode.

   This IP VLAN does not need to contain Layer 3 information. An IP VLAN with Layer 3 information is a network-based VLAN. See "Network-based IP VLANs" later in this chapter.

2. Configure an IP routing interface with a network address and subnet mask, and specify the interface type vlan.

3. Select the IP VLAN index that you want to "bind" to that IP interface.

   If Layer 3 information is provided in the IP VLAN for which you are configuring an IP routing interface, the subnet portion of both addresses must be compatible. For example:

   - IP VLAN subnet 157.103.54.0 with subnet mask of 255.255.255.0
   - IP host interface address 157.103.54.254 with subnet mask of 255.255.255.0

   Layer 2 (bridging) communication is still possible within an IP VLAN (or router interface) for the group of ports within that IP VLAN:

   - For allClosed VLANs, IP data destined for a different IP subnetwork uses the IP routing interface to reach that different subnetwork even if the destination subnetwork is on a shared port.
   - For allOpen VLANs, using the destination MAC address in the frame causes the frame to be bridged; otherwise, it is routed in the same manner as allClosed VLANs.

4. Enable IP routing.

You perform similar steps to create IPX and AppleTalk routing interfaces. For more information, see the routing chapters in this guide (for routing protocols such as IP, OSPF, IPX, and AppleTalk).

Example: Protocol-based VLANs for Routing

Figure 32 shows a VLAN configuration that contains three IP VLANs without overlapped ports.

Figure 32 Sample VLAN Routing Configuration

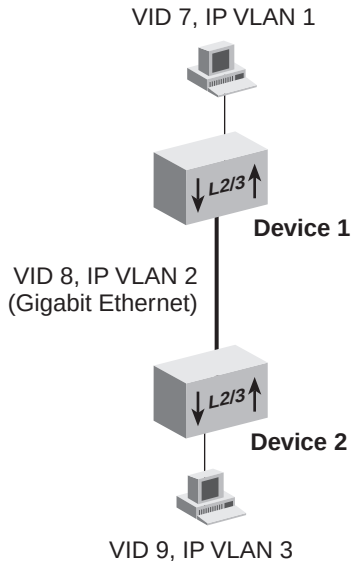


Table 21 shows the information that is used to configure these routing VLANs:

Table 21 Sample Protocol-based VLAN Definitions for Routing

IP VLAN1 (Device 1)	IP VLAN2 (Devices 1 and 2)	IP VLAN3 (Device 2)
VLAN Index 2	VLAN Index 3	VLAN Index 4
VID 7	VID 8	VID 9
Bridge port 6	Bridge port 13 on device 1 Bridge port 1 on device 2	Bridge port 8
Protocol type *IP*	Protocol type *IP*	Protocol type *IP*
No Layer 3 address	No Layer 3 address	No Layer 3 address
Per-port tagging: Port 6 — *none*	Per-port tagging: Ports 1 and 13 — *none*	Per-port tagging: Port 8 — *none*
VLAN name *"IP VLAN 1"*	VLAN name *"IP VLAN 2"*	VLAN name *"IP VLAN 3"*

Network-based IP VLANs

For IP VLANs only, you can configure network-layer subnet addresses. With this additional Layer 3 information, you can create multiple independent IP VLANs with the same bridge ports. Untagged frames are assigned to a network-based VLAN according to both the protocol (IP) and the Layer 3 information in the IP header. Assigning Layer 3 address information to IP VLANs is one way that network administrators can manage their IP routing interfaces by subnetwork.

Because network-based IP VLANs accommodate multiple routing interfaces over the same set of ports without tagging, this option can be useful in allOpen mode. In allClosed mode, overlapped network-based IP VLANs must be IEEE 802.1Q tagged, which means that the system does not use the Layer 3 information.

Important Considerations

When you create a network-based VLAN interface, review these guidelines:

- You can either configure network-based IP VLANs (IP VLANs with unique Layer 3 IP addresses) or you can define a single protocol-based VLAN with the protocol type IP and then define multiple IP routing interfaces for that VLAN.
- The network information is used only when multiple network-based VLANs are defined on a particular port. In situations where there is only one network-based VLAN defined on a port, the VLAN is treated as an ordinary IP protocol-based VLAN, and network-based information is ignored.
- When they are overlapped, network-based VLAN interfaces take precedence over protocol-based and port-based VLAN interfaces.

- You can define only *one* IP routing interface for a network-based VLAN. When you define an IP routing interface with the interface type vlan, the system does not allow you to select a network-based IP VLAN that already has a routing interface defined for it. For more information about IP routing interfaces, see Chapter 11.
- If you define multiple interfaces for an IP VLAN (instead of defining a network-based VLAN), you cannot subsequently modify that IP VLAN to supply Layer 3 address information. If only one routing interface is defined for the IP VLAN, then you can supply Layer 3 address information as long as it matches the Layer 3 information specified for the routing interface.
- In allClosed VLAN mode, you must supply IEEE 802.1Q tagging for any overlapped ports. Therefore, this feature has no added benefit. When IEEE 802.1Q tagging is implemented, implicit VLAN membership information such as the protocol or Layer 3 IP network address is not used; the frame is assigned to the VLAN based solely on the tag VID and the receive port.
- In allOpen mode, you are not required to supply the IEEE 802.1Q tagging. To ensure line-speed throughput for overlapped network-based IP VLANs in allOpen mode, however, you should still supply the IEEE 802.1Q tagging.

Example of Network-based VLANs

Figure 33 shows two network-based IP VLAN interfaces. The IPVLAN2 interface includes trunk ports and defines the protocol type IP, a Layer 3 address, a subnet mask, and IEEE 802.1Q tagging on bridge ports 6 and 7 (the anchor port for the trunk that uses ports 7 and 8). The IPVLAN3 interface defines IP and a different Layer 3 address; it uses exactly the same ports as IP VLAN2, with IEEE 802.1Q tagging on bridge ports 6 and 7.

Figure 33 Network-based VLANs with Overlapped Ports

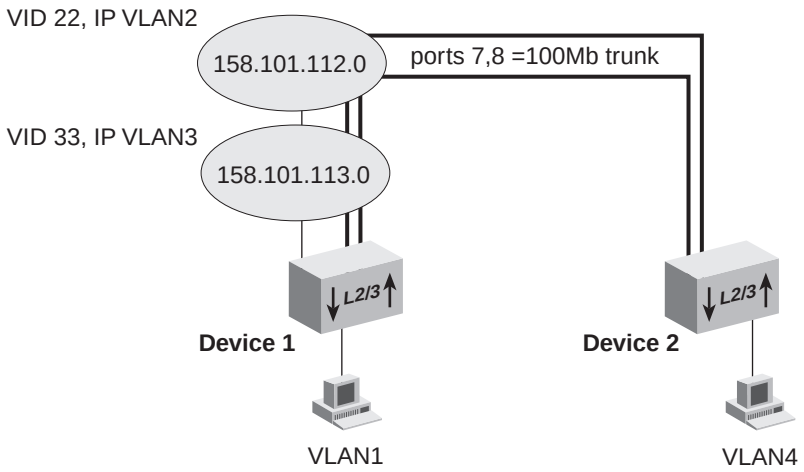


Table 22 shows the information that can be used to configure the two overlapped IP VLANs on Device 1:

Table 22 Network-based IP VLAN Definitions with Overlapped Ports

IP VLAN2	IP VLAN3
VLAN Index 2	VLAN Index 3
VID 22	VID 33
Bridge ports *6, 7* (7 is anchor port for a trunk that uses ports 7 and 8)	Bridge ports *6,7* (7 is anchor port for a trunk that uses ports 7 and 8)
Protocol type *IP*	Protocol type *IP*
158.101.112.0 Layer 3 address 255.255.255.0 mask	158.101.113.0 Layer 3 address 255.255.255.0 mask
Per-port tagging: ■ Port 6 — *IEEE 802.1Q* ■ Anchor port 7 — *IEEE 802.1Q*	Per-port tagging: ■ Port 6 — *IEEE 802.1Q* ■ Anchor port 7 — *IEEE 802.1Q*
VLAN name *IPVLAN2*	VLAN name *IPVLAN3*

Rules of VLAN Operation

After you select a VLAN mode for the system and create VLAN interfaces with VLAN characteristics such as IEEE 802.1Q or no tagging, port membership, protocol type, and Layer-3 (network) address information, the system determines the details of VLAN operation by observing two main types of rules:

- **Ingress rules** — Assign an incoming frame to a specific VLAN.
- **Egress rules** — Use standard bridging rules to determine whether the frame is forwarded, flooded, or filtered. These rules also determine the tag status of the transmitted frame.

These rules are classified in the IEEE 802.1Q standard. In addition, the system relies on some system-specific rules.

Ingress Rules

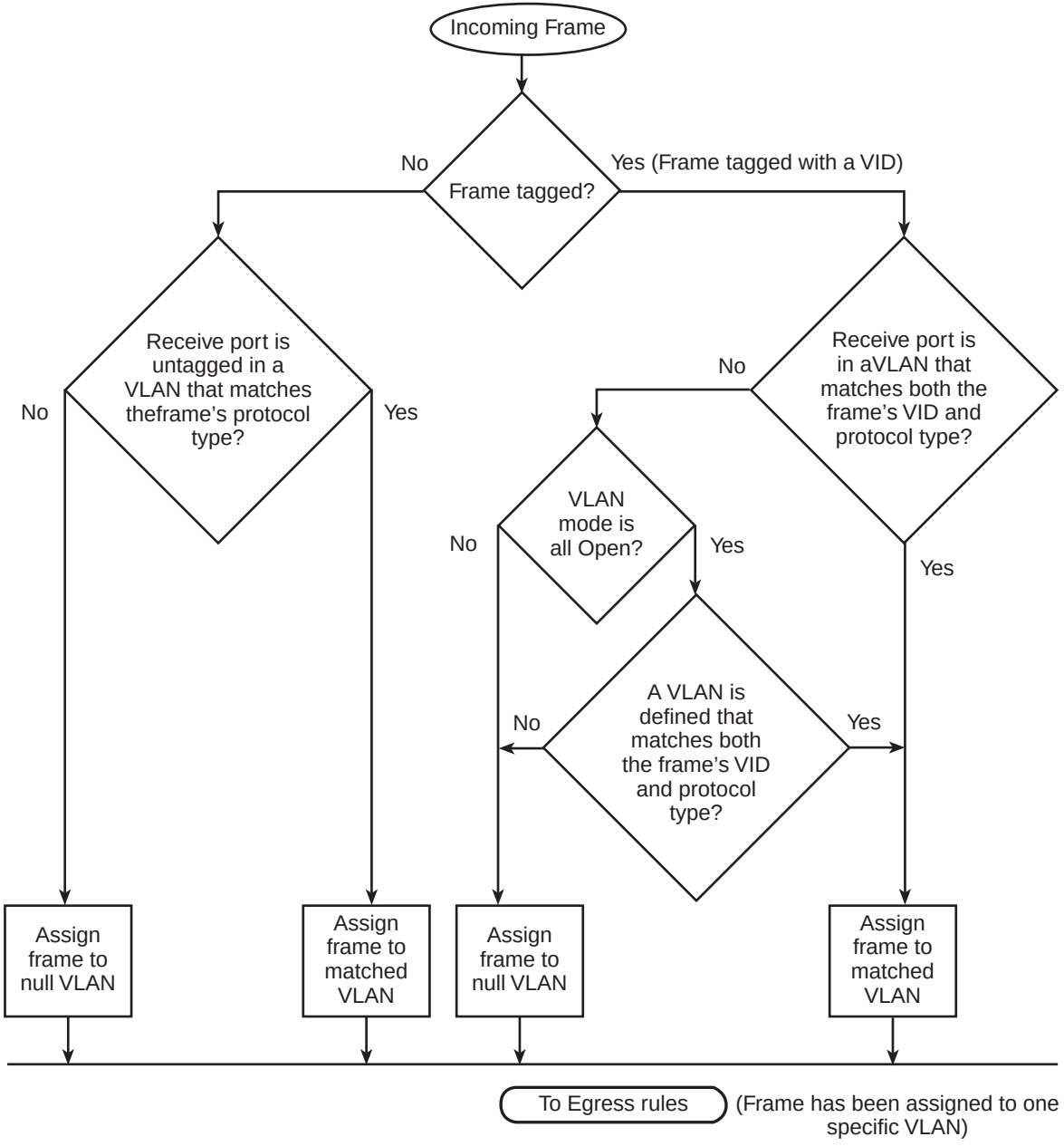
These rules determine the VLAN to which an *incoming* frame belongs. The frame is assigned to the VLAN that has the most specific match. The system uses this protocol match hierarchy to find the most specific match:

1. IEEE 802.1Q tag VID value, if the frame is tagged
2. A specific protocol match (for example, IP, IPX, or AppleTalk)
3. Either the default VLAN (an untagged, unspecified protocol type VLAN with all ports and a VID of 1) or any VLAN that has the unspecified protocol type
4. The *null VLAN*, a special VLAN that the system uses if the frame cannot be assigned to any VLAN. This VLAN has no ports and has no address table (in allClosed mode).

The CoreBuilder 3500 Release 3.0 ingress rules are classified according to the tag status of the frame and the VLAN mode (allOpen for open VLANs or allClosed for closed VLANs). For the ingress rules, the system considers a priority tagged frame to be an untagged frame.

The flow chart in Figure 34 shows the VLAN ingress rules for the system at Release 3.0.

Figure 34 Flow Chart for 3.0 Ingress Rules



The ingress rules for tagged frames vary for the various system releases. Table 23 summarizes the differences.

Table 23 Ingress Rules for IEEE 802.1Q Tagged Frames Based on VLAN Mode and Software Release Number

VLAN Mode	Release 1.2	Release 2.0	Release 3.0	Action Without Required Match
allOpen	The tagged frame is assigned to one of the configured VLANs if: ■ The VID of the frame matches that of a VLAN *and* ■ A port in that VLAN is tagged	The tagged frame is assigned to one of the configured VLANs if: ■ The VID of the frame matches that of a VLAN	The tagged frame is assigned to one of the configured VLANs if: ■ The VID of the frame matches that of a VLAN *and* ■ The protocol type of the frame matches that of the same VLAN	The frame is assigned to the null VLAN. It can still be forwarded (untagged) if the destination address of the frame is associated with another port in the bridge address table.
allClosed	The tagged frame is assigned to one of the configured VLANs if: ■ The receive port is in a VLAN with a VID that matches that of the frame *and* ■ A port in that VLAN is tagged	The tagged frame is assigned to one of the configured VLANs if: ■ The receive port is in a VLAN with a VID that matches that of the frame	The tagged frame is assigned to one of the configured VLANs if: ■ The receive port is in a VLAN with a VID matching that of the frame *and* ■ The protocol type of the frame matches that of the same VLAN	The frame is assigned to the null VLAN and dropped.

Egress Rules These rules determine whether the *outgoing* frame is forwarded, filtered (dropped), or flooded; they also determine the frame's tag status. Although the same standard bridging rules apply to both open and closed VLANs, they result in different behavior depending on the allOpen mode (one address table for the system) versus allClosed mode (one address table for each VLAN).

Standard Bridging Rules for Outgoing Frames

The frame is handled according to these bridging rules:

- If the frame's destination address matches an address that was previously learned on the receive port, it is *filtered* (dropped).
- If the frame's destination address matches an address that was learned on a port other than the receive port, it is *forwarded* to that port if the receive port and transmit port are in the same VLAN or the system is in allOpen mode.
- If a frame with an unknown, multicast, or broadcast destination address is received, then it is *flooded* (that is, forwarded to all ports on the VLAN that is associated with the frame, except the port on which it was received). Those frames assigned to the null VLAN are not flooded to any ports because no ports are associated with the null VLAN. See "Examples of Flooding and Forwarding Decisions" later in this chapter.
- If the frame's destination address matches a MAC address of one of the bridge's ports, it is further processed, not forwarded immediately. This type of frame is either a management/configuration frame (such as a RIP update, SNMP get/set PDU, Administration Console Telnet packet, or a Web Management Interface http packet), or it is a routed packet. If it is a routed packet, the system performs the routing functions described in the appropriate routing chapter (for example, IP, OSPF, IPX, or AppleTalk).

For example, if a frame is associated with VLAN A and has a destination address associated with VLAN B, the frame is flooded over VLAN A in allClosed mode but forwarded untagged in allOpen mode.

Tag Status Rules

After the VLAN and the transmit ports are determined for the frame, the Tag Status rules determine whether the frame is transmitted with an IEEE 802.1Q tag. Priority tagged frames for QoS use the same frame format as IEEE 802.1Q tagging but with a VID of 0. Priority tagged frames received by the system are transmitted as either untagged frames (that is, no priority tagging) or IEEE 802.1Q tagged frames.

- For each port on which the frame is to be transmitted.
- If that port is tagged for the VLAN associated with the frame, transmit the frame as a tagged frame.
- If that port is *not* tagged for the VLAN that is associated with the frame, transmit the frame as an untagged frame.

If the transmit port is not a member of the assigned VLAN, the frame is transmitted untagged. For VLANs in allOpen mode, this result may occur in either of these situations:

- *If the frame is assigned to the null VLAN. (The frame can still be forwarded if the address was statically entered in the address table or dynamically learned on another VLAN.)*
- *If the frame is assigned to a specific VLAN but the transmit port is* not *part of this VLAN.*

Examples of Flooding and Forwarding Decisions

This section provides several examples of flooding and forwarding decisions.

Example 1: Flooding Decisions for Protocol-based VLANs

Table 24 shows how flooding decisions are made according to three VLANs that are set up by protocol (assuming a 12-port configuration). In this example, ports and frames are untagged and the destination address is unknown, multicast, or broadcast.

Table 24 Protocol-based VLANs and Flooding Decisions

Index	VID	VLAN Name	Ports
1	1	Default	1 – 12
2	2	IP1	1 – 8
3	3	IPX1	9 – 11

Untagged data received on this port	Is flooded on this VLAN	Because
IP - port 1	IP1, VID 2	IP data received matches IP1 on the source (receive) port.
IPX - port 11	IPX1, VID 3	IPX data received matches IPX1 on the source port.
XNS - port 1	Default, VID 1	XNS data received matches no protocol VLAN, so the Default VLAN is used.

Example 2: VLAN Exception Flooding

If an untagged frame arrives on an untagged bridge port that belongs to a VLAN that matches the protocol type of the incoming frame, the frame is assigned to the matching VLAN. The default VLAN (if it exists) provides the match and defines the flooding domain for the data when other VLANs that match the frame's protocol type are defined in the system but not on the receive port. This case is called *VLAN exception flooding.*

Table 25 shows how the VLAN exception flooding decision is made (assuming a 12-port configuration).

Table 25 VLAN Exception Flooding

Index	VID	VLAN Name	Ports
1	1	Default	1 – 12
2	2	IP1	1 – 8

Untagged data received on this port	Is flooded on this VLAN	Because
XNS - port 1	Default, VID 1	XNS data on port 1 matches the unspecified protocol of the default VLAN on port 1.
IP - port 2	IP1, VID 2	IP data received matches IP1 for source ports 1 – 8.
IP - port 12	Default, VID 1	IP data on port 12 matches the unspecified protocol of the default VLAN on port 12.

Rules for Network-based (Layer 3) VLANs

Whenever an IP VLAN is defined with Layer 3 information, another VLAN, called the *All IP Subnets* VLAN, is defined over the same ports. Information about this VLAN is not available to the network administrator. Also, this VLAN has no VID associated with it and has no IEEE 802.1Q tagging on any of the ports. Incoming IP frames are assigned to this VLAN if they cannot be assigned to any of the network-based IP VLANs.

You can either configure network-based IP VLANs (IP VLANs with unique Layer 3 IP addresses) or you can define a single protocol-based VLAN with the protocol type IP and then define multiple IP routing interfaces for that VLAN.

The following IP protocols are applicable to network-based VLANs:

- IP (hexadecimal 0800 or 0x0800)
- ARP (0x0806)
- RARP (0x8035)

The frames associated with these protocols have different ingress rules for assignment to the appropriate network-based VLAN:

- **IP frames** — These frames are assigned to the network-based IP VLAN if the IP source address is consistent with the VLAN subnetwork and the IP destination address is one of the following:
  - 0.0.0.0
  - 255.255.255.255
  - A Class D (multicast) address

  Otherwise, assign the frame to the network-based IP VLAN if the IP destination address is consistent with the VLAN subnetwork. Otherwise, assign the frame to the *All IP Subnets* VLAN.
- **ARP frames** — These frames are assigned to the network-based IP VLAN if the IP destination address is consistent with the VLAN subnetwork and the IP source address is 0.0.0.0. Otherwise, assign the frame to the network-based IP VLAN if the IP source address is consistent with the VLAN subnetwork. Otherwise, assign the frame to the *All IP Subnets* VLAN.
- **RARP frames** — These frames are assigned to the *All IP Subnets* (multicast) VLAN.

Example 3: Decisions for One Network-Based VLAN

Table 26 shows the information for one network-based IP VLAN and how forwarding and flooding decisions are made for this VLAN.

Table 26 One Network-based VLAN and Forwarding and Flooding Decisions

Index	VID	VLAN Name	Ports	IP Subnet
2	2	IP_100	1 (untagged)	158.101.100.0
			2 – 6 (tagged)	mask: 255.255.255.0

Frame received on Port 1	Action
■ IP Frame (Protocol 0x0800) ■ IP destination address (DA) 158.101.103.1 ■ MAC DA is known on port 6	Frame is assigned to the IP_100 VLAN and transmitted on port 6 tagged.
■ RARP Response Frame (Protocol 0x8035) ■ IP DA = 158.101.103.2 ■ MAC DA is unknown	Frame is assigned to the IP_100 VLAN and transmitted on port 6 tagged.

Example 4: Forwarding and Flooding for Network-Based VLANs

Table 27 shows the information for network-based IP VLANs and how forwarding and flooding decisions are made according to these VLANs. In the following example, the system is in allOpen mode and the incoming frame is untagged.

Table 27 Network-based VLANs and Forwarding and Flooding Decisions

Index	VID	VLAN Name	Ports	IP Subnet
2	2	IP_100	1 (untagged)	158.101.100.0
			2 – 5 (tagged)	mask: 255.255.255.0
			6 (untagged)	
3	3	IP_101	1 (untagged)	158.101.101.0
			2 – 6 (tagged)	mask: 255.255.255.0
4	4	IP_102	1 (untagged)	158.101.102.0
			2 – 6 (tagged)	mask: 255.255.255.0

Untagged frame received on Port 1	**Frame is**
■ Frame (Protocol 0x0800) ■ IP destination address (DA) 158.101.100.1 ■ MAC DA is known on port 6	■ Assigned to the IP_100 VLAN ■ Transmitted on port 6 untagged
■ IP Frame (Protocol 0x0800) ■ IP DA = 158.101.101.1 ■ MAC DA is known on port 6	■ Assigned to the IP_101 VLAN ■ Transmitted on port 6, tagged with a VID of 3
■ IP Frame (Protocol 0x0800) ■ IP DA = 158.101.102.1 ■ MAC DA is known on port 6	■ Assigned to the IP_102 VLAN ■ Transmitted on port 6, tagged with a VID of 4
■ Frame (Protocol 0x0800) ■ IP DA = 158.101.103.1 ■ MAC DA is known on port 6	■ Assigned to the *All IP Subnets* VLAN ■ Transmitted on port 6 untagged
■ ARP Request Frame (Protocol 0x0806) ■ IP SA = 158.101.100.2, ■ Broadcast MAC DA	■ Assigned to the IP_100 VLAN ■ Flooded over that VLAN (ports 2 – 5 tagged, port 6 untagged)
■ ARP Request Frame (Protocol 0x0806) ■ IP SA = 158.101.103.2 ■ Broadcast MAC DA	■ Assigned to the *All IP Subnets* VLAN ■ Flooded over that VLAN (ports 2 – 6 untagged)
■ RARP Response Frame (Protocol 0x8035) ■ IP DA = 158.101.102.2 ■ MAC DA is known on port 6	■ Assigned to the *All IP Subnets* VLAN ■ Transmitted on port 6 untagged
■ IP Frame (Protocol 0x0800) ■ IP DA = 255.255.255.255 ■ IP SA = 158.101.101.2, MAC DA is known on port 6	■ Assigned to the IP_101 VLAN ■ Transmitted on port 6 tagged with a VID of 3.

Modifying and Removing VLANs

You can modify or remove any VLANs on your system. Review the following guidelines before you modify or remove VLANs:

- When you modify VLAN information for a VLAN interface other than the Default VLAN on your system, you have the option to change VLAN characteristics such as the VID, member bridge ports, protocol type, and form of explicit tagging. You can modify the bridge ports and port tagging type that is associated with the Default VLAN, but you cannot change its protocol type, name, or VID.
- When you modify or remove a VLAN interface, you must specify a VLAN interface index to identify the VLAN interface. The Default VLAN always uses the VLAN interface index of 1.
- You cannot delete a VLAN for which you have defined a routing interface.
- If you add ports to a specific VLAN, you are permitting additional traffic through that port. If you remove ports from a specific VLAN and the default VLAN is intact, those ports come under jurisdiction of the Default VLAN (unspecified protocol type, and no explicit or implicit tagging).
- Verify that each bridge port is associated with at least one VLAN in order to handle traffic.
- If you modify the default VLAN to remove certain ports, verify that those ports are included in another VLAN. If the VLAN is in allClosed mode, those ports are not able to pass data if they are not part of another VLAN. See "Modifying the Default VLAN" earlier in this chapter for more information about the Default VLAN.
- If you remove the Default VLAN (and you have no other VLANs defined for the system), your ports may not be able to forward data until you create a VLAN for them (for example, if you are using allClosed mode).
- If you remove the Default VLAN, the system can no longer recognize any ports on a newly installed module, even if you delete the Default VLAN and then redefine it on the system.
- If you delete the default VLAN, you must use the reserved VID of 1 if you redefine it.

Monitoring VLAN Statistics

When you display VLAN statistics, the system-generated statistics are valid only under these conditions:

- When the VLANs are defined for the same protocol type (or the type unspecified) *and* do not have any overlapping ports (for example, an IP VLAN1 with ports 1 – 6 and IP VLAN2 with ports 7 – 12).
- If the VLANs are explicitly defined for different protocol types but may have overlapping ports (for example, an IP VLAN and an IPX VLAN that both use ports 2 – 4).

10 PACKET FILTERING

This chapter describes what packet filters are, how to create them, and how to use system utilities to apply them to ports of your CoreBuilder® 3500 system. The chapter covers these topics:

- Packet Filtering Overview
- Key Concepts
- Important Considerations
- Managing Packet Filters
- Tools for Writing Filters
- Downloading Custom Packet Filters
- The Packet Filtering Language
- Common Syntax Errors
- Custom Packet Filter Examples
- Limits to Filter Size
- Using Port Groups in Custom Packet Filters
- Port Group Management and Control Functions
- Long Custom Filter Example

You can control and manage packet filters in either of these ways:

- *From the* `bridge packetFilter` *menu of the Administration Console. See the* Command Reference Guide*.*
- *From the Filter Builder application in the Web Management software. The Filter Builder Help system serves as its documentation.*

Packet Filtering Overview

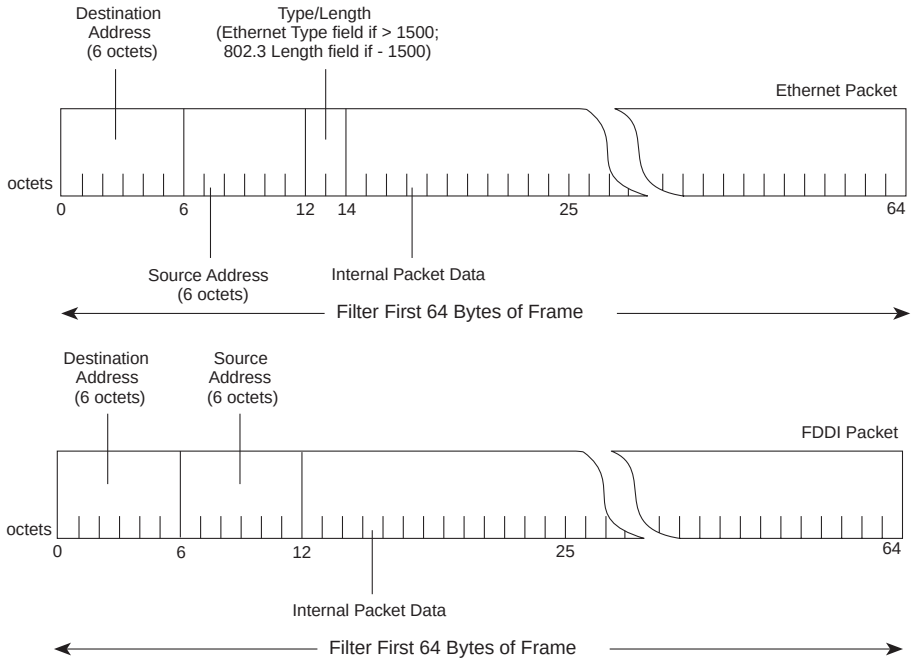
The packet filtering feature allows a switch to make a permit-or-deny decision for each packet based on the packet contents. Use packet filters to control traffic on your network segments to:

- Improve LAN performance.
- Implement LAN security controls.
- Shape traffic flow to emulate virtual LAN (VLAN) behavior. See Chapter 9.

What Can You Filter?

Before you create a packet filter, you must decide which part of the packet you want to use for your filtering decisions. You can filter on any data in the first 64 bytes of the *frame*. You can filter Ethernet, Fast Ethernet, Fiber Distributed Data Interface (FDDI), or Gigabit Ethernet frames by the destination address, source address, type, length, or any attribute within the first 64 bytes. Keep in mind that the offsets may differ between FDDI and Ethernet, so the same filter may not work on all interfaces. Ethernet and FDDI packet fields are shown in Figure 35.

Figure 35 Ethernet and FDDI Packet Fields



You must filter on the *input* packet type. For example, if you write a filter that you intend to assign to the transmit path of an Ethernet port, it will not be sufficient to compose a filter that only filters Ethernet traffic. This is because the filtering function is applied *before* the conversion to Ethernet format. Consider all possible sources of the packets. Might the packet originate as an FDDI packet? If so, then filter on the FDDI format as well as any Ethernet source formats.

When Is a Filter Applied? — Paths

Packets travel on many different *paths* through the switch. You can control to which path a filter is applied.

Input Packet Filtering: Receive Path

Input packet filtering applies to packets immediately upon reaching the switch port, before they reach the switch's internal forwarding processing (*receive path*). Because the packets never enter the switch, the switch itself is protected against an external attack.

Output Packet Filtering: Transmit Path

Output packet filtering applies to packets after they have been through the switch's internal forward processing (*transmit path*).

Internal Packet Filtering: Receive Internal Path

Internal packet filtering applies to packets intended for the switch itself (such as pings, Telnet packets, and so forth) on the *receive internal path*.

Path Assignment

After you create a packet filter, you can assign it to any combination of the `transmit all`, `transmit multicast`, `receive all`, `receive multicast`, and `receive internal` paths of each port. The filter executes a series of operations on the packet's contents and, if the result is 0, it stops (filters) the packet. If the result is not 0, the filter allows the packet to pass.

The packet processing paths are defined in Table 28.

Table 28 Packet Processing Paths

Path	Description
Transmit all (txA)	All frames that are transmitted to the segment that is connected to the port
Transmit multicast (txM)	All multicast (including broadcast) frames that are transmitted to the segment connected to the port
Receive all (rxA)	All frames that are received by the port from the segment that is connected to the port
Receive multicast (rxM)	All multicast (including broadcast) frames that are received by the port from the segment that is connected to the port
Receive Internal (rxI)	All frames received by the port that have a system internal destination, such as ping and Telnet packets

Key Concepts

Before you use packet filters, review the following key concepts and terms:

- **Standard Filters** — Packet filters that are supplied with the CoreBuilder 3500 that the hardware executes at wire speed. You can load them from the Administration Console, or select them from the set of predefined filters with the Filter Builder application. (Filter Builder is part of the Web Management suite of applications. See Table 30 later in this chapter.)

One standard hardware filter is supported: the portGroup (rejdiffportgrp) filter.

- **Custom Filters** — Packet filters that are executed in software. You create custom filters in any of these ways:
  - By writing a filter definition using the filter definition language.
  - By selecting from among the predefined custom filters provided by the Filter Builder application.
  - By using the Filter Builder's wizards to construct a new filter. (See Table 30 later in this chapter.)
- **Predefined filters** — Hardware and software filters that are supplied with the Filter Builder application. Filter Builder provides one standard filter that is executed by the hardware; the others are custom filters that are executed in software. (See Table 30 later in this chapter.)
- **Port Groups** — A collection of ports that you can reference in a packet filter. You create port groups from the Administration Console. You can specify different filtering rules between various port groups.

Standard Packet Filters

The CoreBuilder 3500 hardware supports standard packet filters. Standard filters are implemented in the ASIC hardware to achieve the wirespeed performance. To load them, use the Administration Console's `bridge packetfilter create` command.

At present, one standard hardware filter is supported: the portGroup (rejdiffportgrp) filter.

Standard packet filter support in the hardware is limited to the *receive all* and *transmit all* paths. Hardware filtering on the *receive multicast, transmit multicast,* and *receive internal* paths is not available; therefore, if you assign a standard filter to one of these paths, the system implements the filter in the software, which can affect performance.

Placing a filter on the *receive* path confines the packet to the segment that it originated from if it does not meet the forwarding criteria. Placing a filter on the *transmit* path prohibits a packet from accessing certain segments unless it meets the forwarding criteria. The system discards any packet that does not meet the forwarding criteria on the *transmit* path.

If you want to filter packets destined for the switch itself (for example, ping packets or Telnet packets), you must use the *receive internal* path. They are not filtered on the *receive all* path.

Custom Packet Filters

You create custom packet filters by writing a *packet filter definition*. Software implements custom filters. Consequently, use custom filters only on ports and paths that need them. Processing too many frames in software can affect performance on the ports where custom filters are assigned.

If you are trying to filter a certain type of broadcast or multicast packet assign the filter to either the txM or the rxM paths, allowing only unicast traffic to bypass the filter.

Each packet-processing path on a port may have a unique custom packet filter definition or may share a definition with other ports on the system. Custom packet filter definitions are written in the *packet filter language,* which allows you to construct complex logical expressions.

After you write a packet filter definition, you load it onto a system; the corresponding port assignments are preserved in the nonvolatile memory (NVRAM) of the system, thus ensuring that the packet filter configuration for each system is saved across system reboots and power failures.

Important Considerations

- After you create a packet filter, you must:
  - Assign the filter to the applicable ports
  - Assign the filter to the applicable transmit and receive paths
  - Define port groups, if needed
- If you assign standard (hardware) filters on the receive multicast and transmit multicast paths, they will be executed in software which can slow the switch substantially. See "Standard Packet Filters" earlier in this chapter for details.
- Processing too many frames in software can affect performance on the ports where custom filters are assigned. See "Custom Packet Filters" earlier in this chapter for details.
- Exit a filter as soon as possible. See "Implementing Sequential Tests in a Packet Filter" later in this chapter for details.

Managing Packet Filters

You can control and manage packet filters from the `bridge packetFilter` menu of the Administration Console, as described in the *Command Reference Guide*.

- **Listing packet filters** — You can list the packet filters that are defined for the system. The display includes the filter identification, filter name (if any), and filter assignments. Use the `bridge packetfilter list` command.
- **Displaying packet filters** — When you display the contents of a single packet filter, you select the packet filter using the filter id number that you see when you list the packet filters. The system displays the packet filter instructions. Comments in the original packet filter definition file are not displayed because they are not saved with the packet filter. Use the `bridge packetfilter display` command.
- **Creating packet filters** — You can create the standard (portGroup) hardware filter or your own custom packet filters. Placing a filter on the *receive* path confines the packet to the segment it originated from if it does not meet the forwarding criteria. Placing a filter on the *transmit* path prohibits a packet from accessing certain segments unless it meets the forwarding criteria. The system discards any packet that does not meet the forwarding criteria. Use the `bridge packetfilter create` command.

- **Deleting packet filters** — Deleting a packet filter removes the filter from the system. A filter cannot be deleted if it is assigned. You must unassign the filter from any ports before you can delete the filter. Use the `bridge packetfilter delete` command.
- **Editing, checking, and saving custom packet filters** — You can use the built-in line editor to edit custom packet filters. After you save the custom packet filter, the software examines it for syntax errors. The system software does not allow you to assign the packet filter to a port until the filter is error-free. Use the `bridge packetfilter edit` command.

  You can also edit a packet filter using an ASCII-based text editor such as EMACS, vi, or Notepad.
- **Loading packet filters** — After you create custom packet filters using an external text editor, you must download the filters using the TFTP or FTP file transfer protocol onto the system from the network host on which you created them. When you have loaded it, the packet filter definition is converted into the internal format that is used by the packet filter code in the system. Use the `bridge packetfilter load` command.
- Filters created with the Filter Builder Web Management application can be downloaded directly from Filter Builder. See the example in "Downloading Custom Packet Filters" later in this chapter.
- **Assigning packet filters** — When you assign a packet filter to one or more ports, you must select the ports and a processing path. For descriptions of the available packet processing paths, see Table 28 at the beginning of this chapter. Each path of each port can have only one packet filter assigned to it; however, you can assign a single packet filter to multiple paths and ports. Use the `bridge packetfilter assign` command.
- **Unassigning packet filters from ports** — To unassign a packet filter from one or more ports, the packet filter must have been assigned to at least one port. Use the `bridge packetfilter unassign` command.
- **Defining port groups** — Before you assign packet filters that refer to port groups, create the port groups. See "Defining Port Groups" later in the chapter for more information.

See the *Command Reference Guide* for more information about using these commands and management functions.

Tools for Writing Filters

The following tools can be used to create packet filters.

- ASCII Text Editor
- Built-in Line Editor
- Web Management Filter Builder Tool

ASCII Text Editor

You can create a new custom packet filter using an ASCII-based text editor (such as EMACS, vi, or Notepad). By using an ASCII-based text editor on a networked workstation, you can create multiple copies of the packet filter definition, which you can then store and copy onto one or more systems from the workstation. This method also allows you to archive copies of filter definitions and put them under source code control.

Built-in Line Editor

You can create a new custom packet filter using the line editor that is built into the Administration Console. The built-in text editor provides a minimal set of EMACS-style editing functions that you can use to edit a packet filter definition one line at a time. A single line is limited to no more than 79 characters. The number of lines is limited only by available memory.

Because the built-in editor is deliberately limited in scope, this method is most suited to making small temporary changes to a running filter.

The built-in editor assumes a terminal capability no higher than a glass tty (that is, it does not assume an addressable screen). You can place any ASCII printable character into the editing buffer at the cursor position. If the number of characters in the line buffer exceed the maximum number of characters permitted for the line, the characters that fall outside maximum line length are discarded. The built-in editor initially operates in *insert* mode. Table 29 summarizes the commands that the editor supports.

Table 29 Commands for the Built-In Packet Filter Editor

Command	Keys	Description
List buffer	Ctrl+l	Displays each of the lines in the editing buffer, and then redisplays the line currently being edited.
Next Line	Ctrl+n	Moves cursor to start of next line.
Previous Line	Ctrl+p	Moves cursor to start of previous line.
Start of Line	Ctrl+a	Moves cursor to the start of the line it is in.
End of Line	Ctrl+e	Moves cursor to the end of the line it is in.
Left 1 Character	Ctrl+b	Moves cursor *left* one character within a line.
Right 1 Character	Ctrl+f	Moves cursor *right* one character within a line.
Insert Line	Enter	Inserts a new line. The new line becomes the current line, with the cursor positioned at the start. If the cursor is positioned over the first character on a line when you press [Enter], a blank new line is inserted before the current line. Otherwise, the current line is split at the cursor position, with the current line retaining the characters before the cursor, followed by the new line containing the rest of the characters.
Delete Previous Character	Ctrl+h	Deletes a single character preceding the cursor and shifts the remainder of the line *left* one position.
Delete Current Character	Ctrl+d	Deletes a single character under the cursor and shifts the remainder of the line *left* one position.
Delete Line	Ctrl+k	Deletes the remainder of the line from the current cursor position. If the cursor is positioned over the first character, all of the characters on the line are deleted, but the line is retained. A second Delete Line command removes the line from the edit buffer.
Insert/Overstrike Toggle	Ctrl+o	Toggles between the insert mode and overstrike mode.
Write Changes	Ctrl+w	Writes (saves) the current contents of the edit buffer into the packet filter definition. No syntax verification of the definition is performed at this point other than to verify that the length of the source is within the maximum limits. If the source is too long, the message `Error: Edit buffer exceeds maximum length` is displayed. The contents of the edit buffer are unaffected; however, the packet filter definition contains only those lines that fit entirely within the length limitation.
Exit Editor	ESC	Allows you to leave the editor. You receive a warning if the edit buffer has not been successfully written since the last modification. You can either discard the changes or return to the editor. Note that only those changes made since the last Write Changes command are discarded.

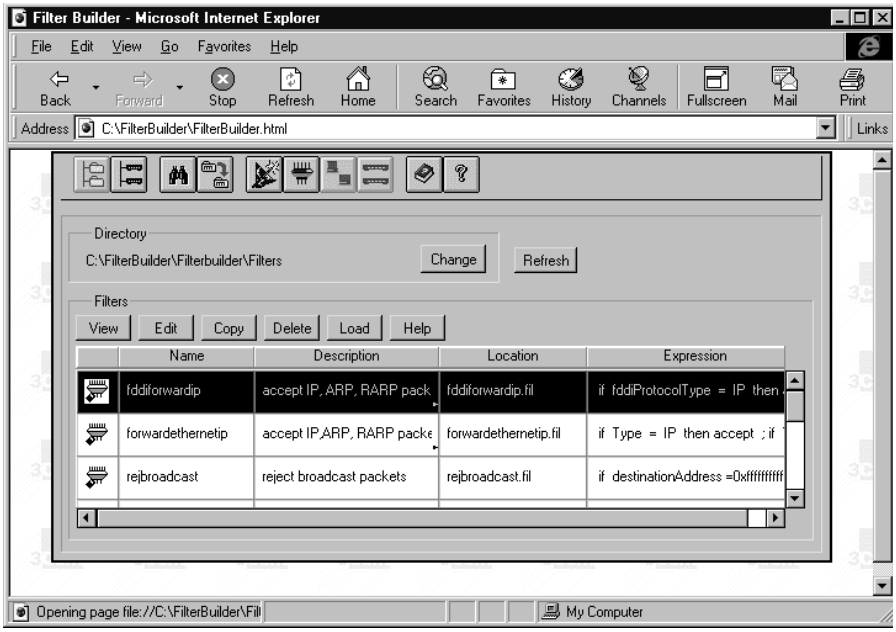
Web Management Filter Builder Tool

Filter Builder is part of the Web Management tool suite. You can use Filter Builder to:

- Download one of the predefined standard hardware or custom software filters to your switch.
- Create your own custom filters and then download them to your switch.

With Filter Builder, you can implement custom packet filters easily and verify that your filters are syntactically correct before you test them on the system. Figure 36 shows the Filter Builder configuration form.

Figure 36 Filter Builder Configuration Form



Filter Builder includes 10 predefined filters, which are displayed on the Filter screen. Table 30 lists the filters by name, what each does, and whether the filter operates in the software or the hardware.

Table 30 Predefined Filter Builder Packet Filters

Filter Name	Type	Filtering Function	Implemented
fddiforwardip	Custom	Forwards FDDI IP, ARP, and RARP packets	Software
forwardethernetip	Custom	Forwards Ethernet IP, ARP, and RARP packets	Software
rejbroadcast	Custom	Rejects broadcast packets	Software
rejdiffaddgrp	Standard	Rejects packets from a specific address group	Software
rejdiffportgrp	Standard	Rejects packets from a specific port group	Hardware
rejethernetappletalk	Custom	Rejects Ethernet AppleTalk packets	Software
rejethernetipx802	Custom	Rejects Ethernet IPX packets	Software
rejfddiat	Custom	Rejects FDDI AppleTalk packets	Software
rejfddiipx802	Custom	Rejects FDDI IPX packets	Software
rejmulticast	Custom	Rejects multicast packets	Software

You can distinguish predefined filters from the custom filters that you create by the icon that pertains to each filter's name in the list on the Filter tab. The icon for predefined filters has a lock in the lower left corner, which indicates that the filter is write protected; you cannot edit or delete it.

Although the predefined filters are write-protected, you can edit a predefined filter indirectly *by copying it, giving it a new name, and then editing it.*

To create a filter, Filter Builder has two interfaces:

- **Filter Wizard** — If you are unfamiliar with the packet filtering or to create a simple filter, use this interface.
- **Create or Edit Filter window** — If you are familiar with the packet filtering or to create a complex filter, use this interface.

For more information on the Filter Builder tool, see the *Web Management User Guide* and the Filter Builder's Help system.

Downloading Custom Packet Filters

You download a packet filter from the system on which it was created to the CoreBuilder 3500 in one of two ways:

- If you are using the Filter Builder Web Management applications, you can download filter through the Filter Builder interface.
- If you have created the filter with a text editor, you can download the filter using TFTP or FTP.

Download with Filter Builder

To download a filter through Filter Builder:

- Your system must be running the TFTP file transfer daemon (server). Filter Builder performs a "put" operation on the file to transfer it to the CoreBuilder 3500.
- TFTP's home directory must be set to the directory where the filter file is stored. On PC systems, by default, Web Management is installed in `/3com` and Filter Builder's filter directory is `/3Com/Filterbuilder/Filters`. If you store your own filters in a different directory, point TFTP to it instead.
- On the CoreBuilder 3500, you must set the file transfer protocol to TFTP with the `system fileTransfer TFTP` option.

To download the filter:

1. Display the directory in which the filter is stored.
2. Select the filter file.
3. Click *Load*.

   The Load Filter panel is displayed. These items are checked: Use TFTP and Include the full file path. Leave them checked.
4. Type the IP address of the CoreBuilder 3500 to which you want to download the filter.
5. Always leave Slot at its default value, which is 1.
6. Click *OK*.

After downloading, the filter resides in volatile memory on the switch. You must assign the filter to a port and path before it will work:

1. Go to the Device Information panel.
2. Type the IP address of the CoreBuilder 3500.
3. Click *Refresh*.

4 When the downloaded filter is displayed, press *Assign*.

5 Always leave Slot as 1.

6 Check the type of port that you want to filter.

7 Type the number of the port(s) that you want to filter.

8 Check the path(s) that you want to filter.

Download an ASCII File

To download a filter created as an ASCII file, without the presence of Filter Builder:

- Your system must be running either the TFTP or the FTP file transfer daemon (server). The CoreBuilder 3500 performs a "get" operation on the file to transfer it from the remote system.
- The filter file must be placed in a directory that either you or the CoreBuilder 3500 has permission to access. TFTP and FTP use different permission mechanisms.
  - TFTP grants all outside systems permission to access files in its defined home directory. See your system's TFTP documentation to find out where the TFTP home is, then either copy the filter file to that directory, or change home to point to the directory the file is in. On most UNIX systems, TFTP's home directory is `/tftpboot.` Next, verify that the filter file has "world read" access.
  - FTP, through the CoreBuilder 3500, prompts you for a username and password valid on the remote system. You are now "logged in" as that user and have permission to access any file which that user can access.

To download the filter:

1 Connect to the CoreBuilder 3500.

2 Use the `system fileTransfer` option to set either FTP or TFTP file transfer.

3 Type: **`bridge packetfilter load`**

4 You are prompted in turn to supply:

- The IP address of the remote system where the file is.
- The full pathname to the file.

At this point, TFTP simply transfers the file. FTP prompts for the:

- Remote system username.
- Remote system password.

Example (FTP):

```
Select menu option: bridge packetFilter load
Host IP address: 160.103.8.112
File pathname {?}: /userfiles/thewriter/rejmulticast.fil
User name {?}: thewriter
Password {?}:
Packet filter 1 stored.
```

5 Verify that the filter has been loaded:

Example:

```
Select menu option: bridge packetFilter list

   Packet Filter 1 - rejMulticast
      No port assignments
```

6 At the module prompt, enter **`bridge packetFilter assign`** to assign filters to port(s). See the *Command Reference Guide* for details.

The Packet Filtering Language

You define packet filters using a *stack-oriented* language, which uses a LIFO (last in, first out) queue when the packet filter is running. The program places values (called *operands*) on the stack and tests them with various logical expressions (called *operators*), such as *and*, *or*, *equal*, and *not equal.* These expressions typically test the values of various fields in the received packet, which include MAC addresses, type fields, IP addresses, or any field within the first 64 bytes of any frame.

Principles for Writing a Custom Filter

Before you write a packet filter, understand these basic principles:

- How the Packet Filter Language Works
- What Can You Filter?
- Implementing Sequential Tests in a Packet Filter

A packet filter program is stored in a preprocessed format to minimize the space that is required by the packet filter definition. Comments are stripped. When assigned to a port, the packet filter is converted from the stored format to a run-time format to optimize the performance of the filter. Each system is limited to a maximum of 16 packet filter programs.

How the Packet Filter Language Works

A program in the packet filter language typically consists of a series of one or more instructions that results in the top of the stack containing a byte value after execution of the last instruction in the program. This top-of-stack byte value determines whether to forward or discard the packet.

In this stack-oriented language, instructions:

- *Push* operands onto the stack
- *Pop* the operands from the stack for comparison purposes
- *Push* the results back onto the stack

Therefore, with the exception of the push instructions, instructions (such as logical operators) locate their operands implicitly and do not require additional operand specifiers in the instruction stream.

Opcodes are the variables that are used to identify the type of operands and operators you are specifying in the packet filter instructions.

Procedure for Writing a Custom Filter

This section describes the process of writing a packet filter. Detailed examples are provided in "Long Custom Filter Example" later in this chapter.

You write the instructions for the packet filter using the following syntax:

```
<opcode>[.<size>] [<operand>...] [# <comment>]
```

The opcode descriptions are in "Packet Filter Opcodes" later in this chapter. Table 31 describes the supported operand sizes later in this chapter. The operand value is determined by what you are testing (for example, an address or a length).

Implicit operands for an instruction must be of the size expected by the instruction. Any mismatch in implicit operand size results in an error `operand size mismatch` *when you load the program into the system.*

When you write a packet filter, be sure that you use comments (preceded by #) to describe each step in the filter. This habit helps you to revise filters and enables others to understand and use the filters you create.

To write a packet filter, follow these basic steps:

1. Assign a unique, descriptive name to the filter using the `Name` opcode.
2. Specify what to test. For example, use the `pushField` opcode to select a field in the packet.
3. Specify what to compare to the value in step 2. For example, use the `pushLiteral` opcode to select a constant value.
4. Apply a logic operation to the values in steps 2 and 3. The operator you use depends on what comparison you want to make.

Table 31 describes the instructions and stacks of a packet filter.

Table 31 Packet Filter Instructions and Stacks — Descriptions and Guidelines

Element		Descriptions and Guidelines
Instructions		Each instruction in a packet filter definition must be on a separate line in the packet filter definition file.

	Instruction format	A typical instruction consists of an *opcode* followed by explicit *operands* and a *comment*. Although comments are optional, it is recommended that you use them throughout the packet filter to make it easier for yourself and others to administer the filters. Several opcodes include an explicit operand size specification.
		The general syntax of an instruction is:
		`<opcode>[.<size>] [<operand>...] [# <comment>]`
		Example:
		`pushliteral.l  0xffffff00  #load the type field mask`
		Use any combination of uppercase and lowercase letters for the opcode and size.
		The contents of a line following the first # outside a quoted string are ignored, so use the # to begin your comments. Comments are not stored in the system; they are useful when the filter is created and saved externally.

	Operand sizes	The following operand sizes are supported: ■ 1 byte = .b ■ 2 bytes = .w ■ 4 bytes = .l ■ 6 bytes = .a (Included primarily for use with 48-bit, IEEE, globally assigned MAC addresses)
	Maximum length	The maximum length for a filter definition is 4096 bytes.
Stack		The packet filter language uses a *stack* to store the operands that will be used by an instruction and the results of the instruction. Operands are popped from the stack as required by the instructions. An instruction using two or more operands takes the first operand from the top of the stack, with subsequent operands taken in order from succeeding levels of the stack. The stack is a maximum of 64 bytes long, with space within the stack allocated in multiples of 4 bytes. Thus you can have a maximum of 16 operands on the stack. The address size operand .a consumes 8 bytes on the stack, decreasing the maximum number of operands on the stack for a 48-bit address.

The Ethernet and FDDI packet fields in Figure 35 are used as *operands* in the packet filter. The two simplest operands are described in Table 32.

Table 32 Two Packet Filter Operands

Operand	Description	Opcode
packet field	A field in the packet that can reside at any offset. The size of the field can be 1, 2, 4, or 6 bytes. Typically, you only specify a 6-byte field when you want the filter to examine a 48-bit address.	pushField
constant	A literal value to which you are comparing a packet field. As with a field, a constant can be 1, 2, 4, or 6 bytes long.	pushLiteral

Packet Filter Opcodes Opcodes are instructions used in packet filter definitions. The available opcodes are described in Table 33.

Table 33 Packet Filtering Opcodes

Opcode	Memory Requirements	Description
name "<name>"	2 + *n* bytes, where *n* is the length of the <name>	Assigns a user-defined <name> to the packet filter. The name may be any sequence of ASCII characters other than quotation marks. The name is limited to 32 characters. You can include only a single name statement in each packet filter program.
pushField.size <offset>	3 bytes	Pushes a field from the target packet onto the stack. Packet data starting at <offset> is copied onto the stack. The most significant byte of the field is the byte at the specified offset. The size field of the instruction determines the number of bytes pushed. The pushField instruction provides direct access to any 1, 2, 4, or 6 byte (.b, .w, .l, or .a) field contained within the first 64 bytes of the target packet. Specify the offset as an octal, decimal, or hexadecimal number. ■ Precede an octal number by a "0". ■ Precede a hexadecimal number by either "0x" or "0X". ■ Use either upper or lower case letters for the hexadecimal digits "a" through "f".
pushLiteral.size <value>	1 (.b) 2 (.w) 4 (.l) 6 (.a) bytes depending on the size of <value> plus 1 byte for a total of 2, 3, 5, or 7 bytes	Pushes a literal constant <value> onto the stack. The most significant byte of the <value> is the first byte of the literal. Bytes are copied directly from the operand onto the stack. The size field of the instruction determines number of bytes pushed. Specify the value as either an octal, decimal, or hexadecimal number. ■ Precede an octal number by a "0". ■ Precede a hexadecimal number by either "0x" or "0X". ■ Use either upper or lower case letters for the hexadecimal digits "a" through "f".

Table 33 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description

pushTop	1 byte	Pushes the current top of the stack onto the stack (that is, it reads the top of the stack and pushes the value onto the stack, which effectively duplicates the item currently on top of the stack). The size of the contents of the stack determines the size of the push.
		
		Use `pushTop` for each additional comparison you intend to make with the current top of the stack. The `pushTop` instruction makes a copy of the field more efficiently than if you use a second `pushField` instruction.
		
		If you are writing a filter that is going to check the same offset more than once, such as checking the Ethernet type field to filter multiple protocols, use the following guidelines. Assume that you want to filter DEC LAT, IP, and ARP traffic on a port. Rather than use multiple `pushField .w 12` commands to look at the 12th offset where the Ethernet type field resides, use multiple `pushTop` commands, as shown here:
		
		Original Filter:
		
		`pushField.w 12`
		`pushLiteral.w 0x6004`
		`eq`
		`reject`
		`pushField.w 12`
		`pushLiteral.w 0x0800`
		`eq`
		`reject`
		`pushField.w 12`
		`pushLiteral.w 0x0806`
		`ne`
		
		Shortened Filter:
		
		`PushField.w 12`
		`pushTop`
		`pushTop`
		`pushLiteral.w 0x6004`
		`eq`
		`reject`
		`pushLiteral.w 0x0800`
		`eq`
		`reject`
		`pushLiteral.w 0x0806`
		`ne`

Table 33 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description
pushSPGM	1 byte	Pushes the source port group mask (SPGM) onto the top of the stack. The SPGM is a bitmap representing the groups to which the source port of a packet belongs. This instruction pushes 4 bytes on to the stack. Each port group mask is represented by a single bit in the SPGM bitmap. Port group masks are assigned to the bitmap in sequence, starting with port group mask 1 as the least significant bit through port group mask 32 as the most significant bit. Use `pushSPGM` to filter by port groups. See "Using Port Groups in Custom Packet Filters" for more information.
pushDPGM	1 byte	Pushes the destination port group mask (DPGM) onto the top of the stack. The DPGM is a bitmap representing the groups to which the destination port of a packet belongs. Pushes 4 bytes on to the stack. Each port group mask is represented by a single bit in the DPGM bitmap. Port group masks are assigned to the bitmap in sequence, starting with port group mask 1 as the least significant bit through port group mask 32 as the most significant bit. Use `pushDPGM` to filter by port groups. See "Using Port Groups in Custom Packet Filters" for more information.
eq (equal)	1 byte	Pops two values from the stack and compares them. If they are equal, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determines the size of the operands.
ne (not equal)	1 byte	Pops two values from the stack and compares them. If they are not equal, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The size of the operands is determined by the contents of the stack.
lt (less than)	1 byte	Pops two values from the stack and performs an unsigned comparison. If the first is less than the second, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determine the size of the operands.

Table 33 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description
le (less than or equal to)	1 byte	Pops two values from the stack and performs an unsigned comparison. If the first is less than or equal to the second, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determine the size of the operands.
gt (greater than)	1 byte	Pops two values from the stack and performs an unsigned comparison. If the first is greater than the second, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determine size of the operands.
ge (greater than or equal to)	1 byte	Pops two values from the stack and performs an unsigned comparison. If the first is greater than or equal to the second, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determine the size of the operands.
and (bit-wise AND)	1 byte	Pops two values from the stack and pushes the bit-wise *AND* of these values back onto the stack. The contents of the stack determine the size of the operands and the result. This is a bit-wise operator. Each bit of the operands is logically compared to produce the resulting bit
or (bit-wise OR)	1 byte	Pops two values from the stack and pushes the bit-wise *OR* of these values back onto the stack. The contents of the stack determine the operand size and the result. This is a bit-wise operator. Each bit of the operands is logically compared to produce the resulting bit
xor (bit-wise exclusive-OR)	1 byte	Pops two values from the stack and pushes the bit-wise exclusive-*OR* of these values back onto the stack. The contents of the stack determines the operand size and the result. This is a bit-wise operator. Each bit of the operands is logically compared to produce the resulting bit
not	1 byte	Pops a byte from the stack; if its value is non-zero, a byte containing 0 is pushed back onto the stack. Otherwise, a byte containing the value is pushed back onto the stack.

Table 33 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description
accept	1 byte	Conditionally accepts the packet that is being examined. Pops a byte from the stack. If its value is non-zero, the packet is accepted and evaluation of the filter ends immediately; otherwise, filter evaluation continues with the next instruction. Use `accept` with `and` and `or` operators when you have sequential tests and you would like the filter to accept a packet before the entire expression has been evaluated. Using `accept` can significantly improve the performance of certain types of filters. See "Implementing Sequential Tests in a Packet Filter" elsewhere in the chapter for more information.
reject	1 byte	Conditionally rejects the packet being examined. Pops a byte from the stack. If its value is non-zero, the packet is rejected and filter evaluation ends immediately; otherwise, the filter evaluation continues with the next instruction. Use `reject` with `and` and `or` operators when you have sequential tests and you would like the filter to reject a packet before the entire expression has been evaluated. Using `reject` can significantly improve the performance of certain types of filters. See "Implementing Sequential Tests in a Packet Filter" earlier in the chapter for more information.
shiftl (shift left)	1 byte	Pops two values from the stack and shifts the first operand left by the number of bits specified by the second operand. Bits shifted out of the left side of the operand are discarded, and zeros are shifted in from the right. The resulting value is pushed back onto the stack. The contents of the top of the stack determines the size of the first operand and the size of the result. The second operand is always 1 byte and only the low 5 bits of the byte are used as the shift count.
shiftr (shift right)	1 byte	Pops two values from the stack and shifts the first operand right by the number of bits specified by the second operand. Bits shifted out of the right side of the operand are discarded, and zeros are shifted in from the left. The resulting value is pushed back onto the stack. The contents of the top of the stack determines the size of the first operand and the size of the result. The second operand is always 1 byte and only the low 5 bits of the byte are used as the shift count.

Implementing Sequential Tests in a Packet Filter

Filter language expressions are normally evaluated to completion — a packet is accepted if the value remaining on the top of the stack is nonzero. Frequently, however, a single test is insufficient to filter packets effectively. When more tests are warranted, you want to accept a packet that satisfies one of two cases:

- At least one criterion specified in two or more tests (that is, ORs the results of the tests)

or

- All criteria specified in two or more tests (that is, ANDs the results of the tests)

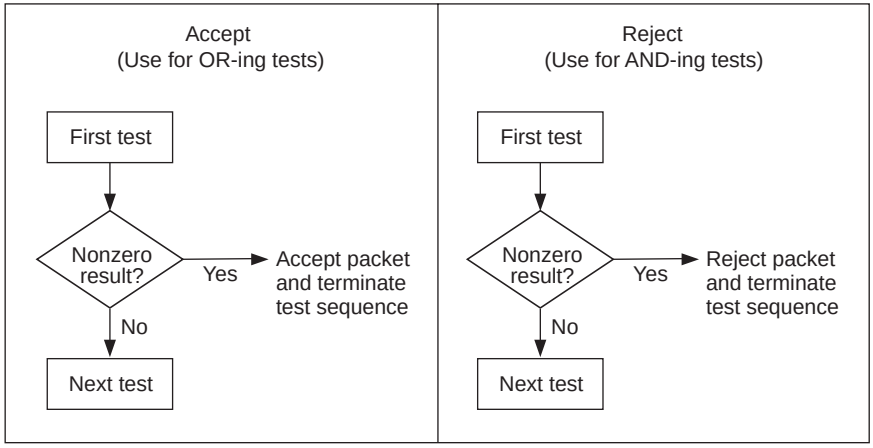
The *accept* and *reject* instructions are used to implement sequential tests, as shown in Figure 37.

In order to optimize a filter's performance, it is best to exit a filter as early as possible. If you wait until the last instruction to make the forward or filter decision, more processing is needed.

The accept and reject criteria allow you to exit a filter early. When using these instructions, construct the packet filter so that tests that apply to the majority of the network traffic are performed first. This ensures that the filter is exited after the first instruction for the majority of packets. Only a small number of packets will require additional tests.

For example, assume you want to create a filter that checks for particular IPX attributes that you want to filter, but most of the traffic on your network is IP traffic. In this case, it would be best to first check each packet to see if it is a IP frame. If it is, you could accept the packet immediately. Now only the smaller number of packets that contain IPX information would be subjected to additional tests.

Figure 37 Accept and Reject Instructions



The following example shows the use of both accept and reject in a packet filter. This packet filter was created for a network that is running both Phase I and Phase II AppleTalk software. The goal of the filter is to eliminate the AppleTalk traffic.

```
Name       "Filter AppleTalk datagrams"
pushField.w       12                    # Get the type field.
pushTop                                 # Make a copy.
pushLiteral       0x809b                # EtherTalk Phase I type.
eq                                      # Test if the packet type is
                                        # equal to the AppleTalk type,
reject                                  # reject the packet and end.
                                        # Otherwise,
pushLiteral.w     0x5dc                 # Largest 802.3 packet size
lt                                      # If this value is less than the
                                        # value in the packet's
                                        # type/length field, then this
                                        # is an Ethernet frame, so
accept                                  # accept the packet if it is not
                                        # 802.3, otherwise…
pushField.a       16                    # get the SNAP OUI and Ethertype
pushLiteral.a     0x03080007809b        # value to compare.
ne                                      # If not equal, then forward the
                                        # packet, otherwise drop it.
```

Common Syntax Errors

When you press the Escape key to exit from the Administration Console's built-in editor or when you load a packet filter definition from across the network, the software examines the definition for syntax errors. Table 34 lists syntax errors and their causes.

Table 34 Common Syntax Errors

Syntax Error	Description
Opcode not found OR Unknown opcode	An opcode was expected on the line and was not found. The opcode must be one of those described in "Packet Filter Opcodes" later in this chapter and must include the size, if any. The opcode and size must be separated by a single period (.) with no intervening spaces. Any mix of uppercase and lowercase characters is permitted.
Operands are not the same size	The opcode requires two operands of the same size. The top two operands on the stack are of different sizes.
Stack underflow	The opcode requires one or more operands. An insufficient number of operands are currently on the stack.
Stack overflow	The opcode pushes an operand on the stack. The stack does not have sufficient room for the operand.
No result found on top of stack	The program must end with a byte operand on the top of the stack. After the last instruction in the program is executed, the stack is either empty or contains an operand other than a byte.
Extra characters on line	The source line contains extraneous characters that are not part of the instruction and are not preceded by a comment character (#).
Expected a byte operand	The opcode requires a byte operand as one of its parameters. The operand is of a size other than a byte.
Offset not found	The opcode requires an offset to be specified. None was found on the line.
Literal not found	The opcode requires a literal value to be specified. None was found on the line.
String not found	The opcode requires a quoted string to be specified. None was found on the line.

Table 34 Common Syntax Errors (continued)

Syntax Error	Description
Invalid characters in number	The number specified as an offset or literal is improperly formatted. Possible causes are 1) lack of white space setting off the number, and 2) invalid characters in the number. Note: The radix of the number is determined by the first 1 or 2 characters of the number: ■ A number with a leading "0x" or "0X" is treated as hexadecimal. ■ All other numbers with a leading 0 are treated as octal. ■ All other numbers are treated as decimal.
Number is too large	The number that is specified as an offset or literal is too large. An offset is limited to 1518 minus the size of the operand. For example, the offset for pushField.b can be no more than 1517, and the offset for pushField.w no more than 1516. A literal value is limited to the number of bytes in the operand size (1, 2, 4, or 6).
Missing open quote on string	The string specified does not have a starting quotation mark (").
String is too long	The string specified is too long. Strings are limited to 32 characters exclusive of the opening and closing quotation marks.
Missing close quote on string	The string specified does not have an ending quotation mark (").
Multiple name statements in program	More than one name statement was found in the program. Only a single name statement is allowed.
Program too large	The program exceeds the maximum size allowed. The causes of this error include a source definition exceeding 4096 bytes, a stored format exceeding 254 bytes, or a run-time format exceeding 2048 bytes. All of these boundary conditions are checked when the filter is loaded.
Too many errors – compilation aborted	The program contains an excessive number of errors. No further syntax errors will be reported. The program stops compiling when this condition occurs.

Custom Packet Filter Examples

The following examples of packet filters, which were built using the packet filter language, start with basic concepts.

Destination Address Filter

This filter operates on the destination address field of a frame. It allows packets to be forwarded that are destined for stations with an Organizationally Unique Identifier (OUI) of 08-00-02. To customize this filter to another OUI value, change the literal value loaded in the last pushLiteral.l instruction. The OUI must be padded with an additional 00 to fill out the literal to 4 bytes.

```
name           "Forward to 08-00-02"
pushField.l    0              # Get first 4 bytes of
                              # destination address.
pushLiteral.l  0xffffff00     # Set up mask to isolate first
                              # 3 bytes.
and                           # Top of stack now has OUI
pushLiteral.l  0x08000200     # Load OUI value.
eq                            # Check for match.
```

Source Address Filter

This filter operates on the source address field of a frame. It allows packets to be forwarded that are from stations with an OUI of 08-00-02. To customize this filter to another OUI value, change the literal value loaded in the last pushLiteral.l instruction. The OUI must be padded with an additional 00 to fill out the literal to 4 bytes.

```
name           "Forward from 08-00-02"
pushField.l    6              # Get first 4 bytes of source
                              # address.
pushLiteral.l  0xffffff00     # Set up mask to isolate first
                              # 3 bytes.
and                           # Top of stack now has OUI
pushLiteral.l  0x08000200     # Load OUI value.
eq                            # Check for match.
```

Length Filter

This filter operates on the length field of a frame. It allows packets to be forwarded that are less than 400 bytes in length. To customize this filter to another length value, change the literal value loaded in the pushLiteral.w instruction.

```
name           "Forward < 400"
pushField.w    12             # Get length field.
pushLiteral.w  400            # Load length limit.
lt                            # Check for frame length <
                              # limit.
```

Type Filter

This filter operates on the type field of a frame. It allows packets to be forwarded that are IP frames. To customize this filter to another type value, change the literal value loaded in the pushLiteral.w instruction.

```
name           "Forward IP frames"
pushField.w    12              # Get type field.
pushLiteral.w  0x0800          # Load IP type value.
eq                             # Check for match.
```

Ethernet Type IPX and Multicast Filter

This filter rejects frames that have either a Novell IPX Ethernet type field (8134 hex) or a multicast destination address.

```
name           "Type > 900 or Multicast"
pushField.w    12              # Get type field.
pushLiteral.w  0x900           # Push type value to test
                               # against.
gt                             # Is type field > 900 (hex)?
reject                         # If yes: reject frame (done).
pushLiteral.b  0x01            # Multicast bit is low-order
pushField.b    0               # bit
and                            # Get 1 st byte of destination
not                            # Isolate multicast bit
                               # Top of stack 1 to accept,
                               # 0 to reject
```

Multiple Destination Address Filter

This filter operates on the destination address field of a frame. It allows packets to be forwarded that are destined for one of four different stations. To tailor this filter to other destinations, change the literal values.

```
name           "Forward to four stations"
pushField.a    0               # Get destination address.
pushTop                        # Make 3 copies of address.
pushTop                        #
pushTop                        #
pushLiteral.a  0x367002010203 # Load allowed destination
                               # address.
eq                             # Check for match.
accept                         # Forward if valid address.
pushLiteral.a  0x468462236526 # Load allowed destination
                               # address.
eq                             # Check for match.
accept                         # Forward if valid address.
pushLiteral.a  0x347872927352 # Load allowed destination
                               # address.
eq                             # Check for match.
accept                         # Forward if valid address.
pushLiteral.a  0x080239572897 # Load allowed destination
                               # address.
eq                             # Check for match.
```

Source Address and Type Filter

This filter operates on the source address and type fields of a frame. It allows XNS packets to be forwarded that are from stations with an OUI of 08-00-02. To customize this filter to another OUI value, change the literal value loaded in the last pushLiteral.l instruction. You must pad the OUI with an additional 00 to fill out the literal to 4 bytes. To customize this filter to another type value, change the literal value loaded into the pushLiteral.w instruction.

```
name           "XNS from 08-00-02"
pushField.w    12             # Get type field.
pushLiteral.w  0x0600         # Load type value.
ne                            # Check for mis-match.
reject                        # Toss any non-XNS frames.
pushLiteral.l  0xffffff00     # Set up mask to isolate first 3
                              # bytes.
pushField.l    6              # Get first 4 bytes of source
                              # address.
and                           # Top of stack now has OUI.
pushLiteral.l  0x09000200     # Load OUI value.
eq                            # Check for match.
```

Accept XNS or IP Filter

This filter operates on the type field of a frame. It allows packets to be forwarded that are XNS or IP frame. The pushTop instruction makes a copy of the type field.

```
name           "Forward IP or XNS"
pushField.w    12             # Get type field.
pushTop                       # Push copy of type.
pushLiteral.w  0x0800         # Load IP type value.
eq                            # Check for match.
pushLiteral.w  0x0600         # Load XNS type value.
eq                            # Check for match.
```

XNS Routing Filter This filter operates on the type and data fields of a frame. It discards all XNS routing packets.

```
name            "Drop XNS Routing"
pushField.w     12              # Get type field.
pushLiteral.w   0x0600          # Load XNS type value.
ne                              # Check for non-XNS packet.
accept                          # Forward if non-XNS packet.
pushLiteral.b   0x01            # Load XNS routing type.
pushField.b     19              # Get XNS type.
ne                              # Check for non-XNS routing
                                # packet.
```

Port Group Filter See "Using Port Groups in Custom Packet Filters" for a port group filter example.

Limits to Filter Size

A packet filter program is stored in a preprocessed format to minimize the space that is required by the packet filter definition. Comments are stripped. When assigned to a port, the packet filter is converted from the stored format to a run-time format to optimize the performance of the filter. Each system is limited to a maximum of 16 packet filter programs.

The maximum length of a packet filter source definition is 4096 bytes.

Storage rules for preprocessed packet filters

Each system provides a maximum of 2048 bytes of nonvolatile storage for *preprocessed* packet filter programs. In the preprocessed stored format:

- A single packet filter program is limited to 254 bytes.
- Each instruction in the packet filter program requires 1 byte for the opcode and size, plus additional bytes for any explicit operands.
- System overhead is 22 bytes, plus a per-packet-filter overhead of 13 bytes. For example, assume a packet filter program requires 200 bytes for storing the instructions in the program. If this packet filter is the only one loaded, the nonvolatile memory required is 22 bytes (for system overhead) plus 13 bytes (for packet filter overhead) plus 200 bytes (for the program itself) — a total of 235 bytes.

Run-time storage of packet filters

For *run-time* storage of packet filter programs, each systems provides a maximum of 8192 bytes. There is no explicit system or per-packet-filter overhead; however, performance considerations can result in unused areas of the run-time storage.

The run-time format is approximately eight times the size of the stored format. Thus a 200-byte packet filter program in stored format expands to approximately 1600 bytes in the run-time format. A single packet filter program cannot exceed 2048 bytes in the run-time format.

Using Port Groups in Custom Packet Filters

You can use a port group (a list of system ports) as filtering criteria in a packet filter.

A packet filter uses the group to make filtering decisions by accessing the group's source port group mask and destination port group mask. In the mask, 32 bits indicate to which of 32 possible groups a port belongs. For example, setting mask bit number 7 assigns the port to group number 7.

You reference these group masks using the opcodes SPGM (source port group mask), and DPGM (destination port group mask). What follows is an example of using port groups in packet filters.

Port Group Packet Filter Example

In this example, packets are not forwarded to ports in groups 3 and 8.

```
Name    "Discard Groups 3 and 8"
pushSPGM                      # Get source port group mask.
pushLiteral.l    0x0084       # Select bits 3 and 8.
and                           # If port group bits 3 & 8 are common
                              # with SPGM, then non-zero value is
                              # pushed onto stack.
pushLiteral.l    0            # Push zero.
eq                            # Only if SPGM is not in port groups
                              # corresponding to bits 3 & 8, then
                              # packet is forwarded.
```

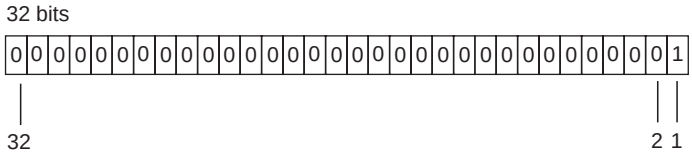
Port Group Filter Operation

When an address is learned on a port, the address and the port number the packet was received on are inserted into the bridge address table and a bit mask that is associated with the address that denotes the group membership is inserted into the port group mask table.

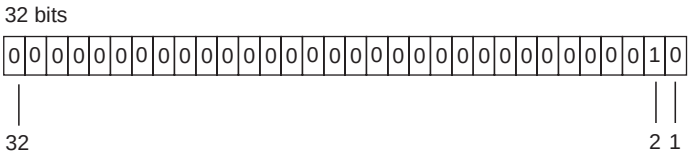
The bridge address table stores each SA/DA MAC address with the port number. The port group masks are stored in a smaller table associating port numbers to port group masks.

For example, assume you defined port group 1 with port 3 being a member and port group 2 with port 5 being a member.

If MAC address 00-80-3e-12-34-56 is learned on port 3 and port 3 belongs to port group 1, it has a port group bit mask for port group 1 inserted into the port group mask table that is associated with the MAC address in the bridge address table. The mask is 32 bits long and contains:



If MAC address 00-80-3e-aa-aa-aa is learned on port 5 and port 5 belongs to port group 2, it has a port group bit mask for port group 2 inserted into the port group mask table that is associated with the MAC address in the bridge address table. The mask is 32 bits long and contains:



When you use the source or destination port group mask (SPGM and DPGM) commands in your filter, you are referencing the port mask of the source port and the port mask of the destination port, respectively. You can use these commands to verify if the source and destination addresses of the packets are members of the same port group to implement your filtering algorithm.

A frame is received (unicast/multicast/broadcast) on the source port. The source port group mask (SPGM) is found in the table of port group masks, using the received port as the index. The destination port group mask (DPGM) is found after the bridge determines whether the port is to be forwarded (known DA unicast) or flooded (unknown DA unicast, or multicast, or broadcast).

For forwarded frames, the single SPGM and DPGM are used in the packet filter.

For flooded frames, each pair of SPGM and DPGM are individually processed. The filter is repeated for each pair of Source and Destination ports.

For example, port 1 has a packet filter using the DPGM assigned to the rxAll path of port 1 and a broadcast frame is received on port 1. The bridge determines that the frame will be flooded to the VLAN ports 2-5. The filter is processed 4 times:

1. Once for the RX port 1 - TX port 2 pair
2. Once for the RX port 1 - TX port 3 pair
3. Once for the RX port 1 - TX port 4 pair
4. Once for the RX port 1 - TX port 5 pair

Use the Standard Port Group filter to contain broadcast, multicast, and flooded frames:

```
pushSPGM
pushDPGM
and
```

If ports 1-3 are in port group 1, 4-5 are in port group 2, and the rxAll path filter is applied to 1-5, then the appropriate filtering restricts the flooding to the corresponding port group.

Table 35 and Table 36 show how each port pair filters or does not filter a broadcast frame that is received on port 1 and destined for ports 2,3,4,5:

Table 35 Port Group Mask Table

Port	Mask
1	0x00000001
2	0x00000001
3	0x00000001
4	0x00000002

Table 36 Filter Processing for Frame Flooded from Port 1 to Ports 2-5

Port Pair	SPGM	DPGM	Filter Result
1 and 2	0x00000001	0x00000001	0x00000001 - ACCEPTED.
1 and 3	0x00000001	0x00000001	0x00000001 - ACCEPTED.
1 and 4	0x00000001	0x00000002	0x00000000 - FILTERED.
1 and 5	0x00000001	0x00000002	0x00000000 - FILTERED.

The result is that the frame is flooded to ports 2,3, and the frame is filtered from ports 4,5.

Port Group Management and Control Functions

Management and control functions to define port groups are provided in the system.

Defining Port Groups

You can configure port groups from the `bridge packetFilter portGroup` menu of the Administration Console, as described in the *Command Reference Guide*.

This section briefly discusses the control and management functions that are implemented in the system for port groups.

You need to assign the port groups before you can assign packet filters to the port groups.

Important Considerations

- **Creating a new group** — When you create a new port group, an unused port group must be available. A port group is limited to the number of ports on a system.
- **Listing groups** — You can list the port groups currently defined on the system. The group id, group name (if any), group mask, and the slots where the group is loaded are displayed.
- **Displaying groups** — The display of a port group shows the group id, the name of the group, and all the addresses or ports included in that group.
- **Deleting groups** — When you delete port groups from a module, those groups are no longer available for use in packet filters.
- **Adding Ports to Groups** — When you add ports to an existing group, you can either enter the ports at the prompts or import them from a file. At least one port group must exist before you can add ports. The same port may be in multiple port groups.
- **Port group size** — The maximum number of ports that a port group can contain is 24, which is the maximum number of ports on a system.

- **Removing ports from a group** — At least one group must exist before you can remove a port.
- **Loading groups** — The Administration Console has no explicit menu item for loading port groups that are defined in a file on a remote host. However, you can *load* groups by creating a script on a remote host (which includes your port group) and then running that script on your Administration Console host.

The following example shows a script that builds two port groups: one named Mktg and the other named Sales:

```
bridge packetFilter portGroup create
15
Mktg
1,2,3
bridge packetFilter portGroup create
32
Sales
5,6
```

When you run the script, your groups are automatically created and stored on the system.

Long Custom Filter Example

The following solution shows a complex packet filter built from three simple packet filters. Each of the shorter, simpler packet filters can be used on its own to accomplish its own task. Combined, these filters create a solution for a larger filtering problem.

Filtering Problem

Your network contains market data feed servers that receive time-critical financial data needed for trading floor applications. At the center of the trading floor networks is a system that is being used to switch Ethernet traffic and to concentrate the market data feed servers onto the FDDI departmental backbone.

The difficulty is that the market data feed servers transmit data to users with broadcast packets that are forwarded to all stations on all segments attached to the system. Not all of the segments attached to the system have stations that require these broadcast updates. To optimize the performance of these Ethernet segments, you need to filter the broadcasts.

Packet Filter Solution

The solution described here is to create a highly sophisticated packet filter that prevents only the broadcast packets from the market data servers from being forwarded onto the segments that are not part of an active trading floor.

Before you write the packet filter, it is important to understand the functions that the filter must provide. The broadcast packets that are transmitted by the servers are based on either TCP/IP or XNS protocol. In both cases, the broadcast packets have socket values that are greater than 0x076c and less than 0x0898. The socket value is located 24 bytes into the packet in IP datagrams and 30 bytes into the packet in XNS datagrams.

You can use this information to create pseudocode that simplifies the process of writing the actual filter. It helps to first write the pseudocode in outline form, as shown here:

1. Determine if the packet has a broadcast address.
2. Determine if the packet is an XNS datagram.
3. Examine socket values and discard the packet if:
   - The socket value is greater than or equal to 0x76c

     and
   - The socket value is less than 0x898
4. Determine if the packet is an IP datagram.
5. If so, then examine socket values and discard the packet if:
   - The socket value is greater than or equal to 0x76c

     and
   - The socket value is less than 0x898
6. End the filter.

The pseudocode translates into the following complex packet filter:

```
Name            “IP XNS ticker bcast filter”
                                # Assign this filter in the multicast path
                                # of a port only--this is very important.
                                #
                                # XNS FILTERING SECTION
                                #
pushField.a     0               # Apply
pushLiteral.a   0xffffffffffff# filter
ne                              # only on broadcast traffic
accept                          #
pushField.w     12              # Get the type field of the packet and
                                # place it on top of the stack.
pushLiteral.w   0x0600          # Put the type value for XNS on top of
                                # the stack.
eq                              # If the two values on the top of the
                                # stack are equal, then return a non-zero
                                # value.
pushLiteral.w   0x76c           # Put the lowest socket value on top of
                                # the stack.
pushField.w     30              # Put the value of the socket from the
                                # packet on top of the stack.
ge                              # Compare if the value of the socket is
                                # greater than or equal to lower bound.
pushLiteral.w   0x0898          # Put the highest socket value on top of
                                # the stack.
pushField.w     30              # Put the value of the socket from the
                                # packet on top of the stack.
lt                              # Compare if the value of the socket is
                                # less than the upper bound
and                             # “and” together with “ge” and “lt” test
                                # to determine if the socket value is
                                # “within” the range. If it is, place a
                                # “one” on the stack
and                             # Compare if XNS & in range
                                #
                                # IP FILTERING SECTION
                                #
pushField.w     12              # Get the type field of the packet and
                                # place it on top of the stack.
pushLiteral.w   0x0800          # Put the type value for IP on top of
                                # the stack.
eq                              # If the two values on the top of the
                                # stack are equal, then return a non-zero
                                # value.
pushLiteral.w   0x76c           # Put the lowest socket value on top of
                                # the stack (1900).
pushField.w     24              # Put the value of the socket from the
                                # packet on top of the stack.
ge                              # Compare if the value of the socket is
                                # greater than or equal to lower bound.
pushLiteral.w   0x0898          # Put the highest socket value on top of
                                # the stack (2200).
pushField.w     24              # Put the value of the socket from the
                                # packet on top of the stack.
lt                              # Compare if the value of the socket is
                                # less than the upper bound
and                             # “and” together with “ge” and “lt”.
                                # Test to determine if the socket value is
                                # “within” the range. If it is in range,
                                # place a “one” will on the stack.
and                             # Compare if IP and in range.
or                              # Determine if the type field is either
                                # XNS or IP.
not                             # Discard if (IP & in range) and (XNS & in
                                # range).
```

The rest of this section concentrates on the parts of the complex filter, showing you how to translate the pseudocode's requirements into filter language. The large filter is broken down into subsets to show how you can create small filters that perform one or two tasks, and then combine them for more sophisticated filtering. Table 37 describes how the purpose of each pseudocode step is accomplished in the small series of packet filters.

Table 37 Pseudocode Requirements Mapped to the Packet Filter

Step	Accomplished through
1	The path to which you assign the packet filter. For administrative purposes, this path is specified in the first two comment lines in the filter definition. The filter must be assigned to a multicast path to filter packets that have broadcast addresses.
2	Packet Filter One — Forwarding XNS packets
3	Packet Filter Two — Looking for specified socket range
4 & 5	Combining a Subset of Filters — Forwarding IP packets within specified socket range

Packet Filter One

This filter is designed to leave a non-zero value on the stack for XNS broadcast packets. If used alone, this filter accepts the very packets we are trying to filter. The reason for doing this will become clear when the filter is combined later in this section.

These steps show how to create this filter.

1 Name the filter:

```
Name    "Forward only XNS packets"
```

It is important to distinguish the function of each filter when it is loaded onto a system that has more than one filter stored in memory. Naming is also useful for archiving filters on a remote system so that the filters can be saved and loaded on one or more systems.

2 Enter executable instruction #1:

```
pushField.a    0
# Clear the stack
```

3 Enter executable instruction #2:

```
pushField.a    0xffffffffffff
# Put the broadcast address on the top of the stack
```

4 Enter executable instruction #3:

```
ne
# not 0xffffffffffff
```

5 Enter executable instruction #4:

```
accept
# accept packet and go no further
```

This accepts all non-broadcast packets.

6 Enter executable instruction #5:

```
pushField.w    12
# Get the type field of the packet and
# place it on top of the stack.
```

7 Enter executable instruction #6:

```
pushLiteral.w    0x0600
# Put the type value for XNS on top
# of the stack.
```

8 Enter executable instruction #7:

```
eq
# If the two values on the top of the stack are equal,
# then return a non-zero value.
```

This returns non-zero for XNS broadcast frames.

Packet Filter Two

This filter is designed to accept packets within the socket range of 0x76c and 0x898. When combined with Filter One above, it forwards XNS packets. Follow these steps to create this filter.

1 Name the filter:

```
Name    "Socket range filter"
```

2 Enter executable instruction #1:

```
pushLiteral.w    0x76c
# Put the lowest socket value on top
# of the stack.
```

3 Enter executable instruction #2:

```
pushField.w    30
# Put the value of the socket from the
# packet on top of the stack.
```

4 Enter executable instruction #3:

```
ge
# Compare if the value of the socket is greater than
# or equal to the lower bound.
```

5 Enter executable instruction #4:

```
pushLiteral.w    0x0898
# Put the highest socket value on
# top of the stack.
```

6 Enter executable instruction #5:

```
pushField.w    30
# Put the value of the socket from the
# packet on top of the stack.
```

7 Enter executable instruction #6:

```
lt
# Compare if the value of the socket is less than the
# upper bound.
```

8 Enter executable instruction #7:

```
and
# “and” together with “ge” and “lt” test to determine
# if the socket value is “within” the range. If it is,
# place a non-zero value on the stack.
```

Combining a Subset of the Filters

The next filter places a non-zero value on the stack for IP packets with a socket range of 0x76c (1900) and 0x898 (2200). The filter combines packet filters one and two, modifying them for IP. These steps show how to create this filter.

1 Name the filter:

```
name    “Only IP pkts w/in socket range”
```

2 Perform steps 6 through 8 as described earlier in “Packet Filter One” except give the pushLiteral instruction (in step 7) a value of 0x0800 for IP.

3 Perform steps 2 through 8 as described earlier in “Packet Filter Two” except the socket value for IP (in steps 3 and 6) is located 24 bytes into the packet (instead of 30 as for XNS).

4 Add an *and* statement to compare the results of step 2 with the results of step 3:

```
and
# Compare if IP and in range.
```

This combination looks like this:

```
Name    “Only IP pkts w/in socket range”
pushField.w     12              # Get the type field of the packet and
                                # place it on top of the stack.
pushLiteral.w   0x0800          # Put the type value for IP on top of
                                # the stack.
eq                              # If the two values on the top of the
                                # stack are equal, then return a non-zero
                                # value.
pushLiteral.w   0x76c           # Put the lowest socket value on top of
                                # the stack (1900).
pushField.w     24              # Put the value of the socket from the
                                # packet on top of the stack.
ge                              # Compare if the value of the socket is
                                # greater than or equal to the lower bound
pushLiteral.w   0x0898          # Put the highest socket value on top of
                                # the stack (2200).
pushField.w     24              # Put the value of the socket from the
                                # packet on top of the stack.
lt                              # Compare if the value of the socket is
                                # less than the upper bound.
and                             # “and” together with “ge” and “lt” test
                                # to determine if the socket value is
                                # “within” the range. If it is in range,
                                # place a “one” will on the stack.
and                             # Compare if IP and in range.
```

Combining All the Filters

Together, the packet filters work to perform the solution to the problem: filtering the broadcast packets from the market data servers. These steps show how to create this filter:

1. Name the filter:

```
name    "Discard XNS & IP broadcast pkts w/in socket range"
```

2. Perform steps 2 through 8 as described earlier in "Packet Filter One."
3. Perform steps 2 through 8 as described earlier in "Packet Filter Two."
4. Add an *and* statement to compare the results of step 2 and the results of step 3:

```
and
# compare if XNS & in range
```

5. Perform steps 2 through 4 as described earlier in "Combining a Subset of the Filters."
6. Add an *or* statement:

```
or
# determine if the type field is either XNS or IP
```

7. Add a *not* statement to discard any matching packets:

```
not
# discard if (IP & in range) or (XNS & in range)
```

The complete packet filter discards IP and XNS packets that are within the specified range.

Optimizing the Filter with Accept and Reject Commands

The following combination filter performs the same function but uses the `accept`, `reject`, and `pushTop` commands to exit the filter as soon as possible to save processing time.

```
Name            “Optimized IP XNS ticker bcast filter”
                                # Assign this filter in the multicast path
                                # of a port only--this is very important.
                                #
                                # XNS FILTERING SECTION (Assuming more XNS traffic)
pushField.a     0               #
pushLiteral.a   0xffffffffffff#
ne                              #
accept                          #
pushField.w     12              # Get the type field of the packet and
                                # place it on top of the stack.
pushTop                         # push copy of type
pushLiteral.w   0x0600          # Put the type value for XNS on top of
                                # the stack.
eq                              # If the two values on the top of the
                                # stack are equal, then return a non-zero
                                # value.
pushLiteral.w   0x76c           # Put the lowest socket value on top of
                                # the stack.
pushField.w     30              # Put the value of the socket from the
                                # packet on top of the stack.
ge                              # Compare if the value of the socket is
                                # greater than or equal to lower bound.
pushLiteral.w   0x0898          # Put the highest socket value on top of
                                # the stack.
pushField.w     30              # Put the value of the socket from the
                                # packet on top of the stack.
lt                              # Compare if the value of the socket is
                                # less than the upper bound
and                             # “and” together with “ge” and “lt” test
                                # to determine if the socket value is
                                # “within” the range. If it is, place a
                                # “one” on the stack
                                #
and                             # Compare if XNS & in range
reject                          # reject if XNS and in  range
                                #
                                # IP FILTERING SECTION
                                #
                                # The type field of the packet was
                                # place on top of the stack by the PushTop command.
                                #
pushLiteral.w   0x0800          # Put the type value for IP on top of
                                # the stack.
ne                              # not IP
accept                          # go no further
                                #
pushLiteral.w   0x76c           # Put the lowest socket value on top of
                                # the stack (1900).
pushField.w     24              # Put the value of the socket from the
                                # packet on top of the stack.
ge                              # Compare if the value of the socket is
                                # greater than or equal to lower bound.
pushLiteral.w   0x0898          # Put the highest socket value on top of
                                # the stack (2200).
pushField.w     24              # Put the value of the socket from the
                                # packet on top of the stack.
lt                              # Compare if the value of the socket is
                                # less than the upper bound
and                             # “and” together with “ge” and “lt”.
                                # Test to determine if the socket value is
                                # “within” the range. If it is in range,
                                # place a “one” will on the stack.
not                             # Discard (IP & in range)
```

11 INTERNET PROTOCOL (IP)

This chapter provides guidelines and other key information about how to configure your system to route packets using the Internet Protocol (IP). Chapter contents include:

- Routing Overview
- Key Concepts
- Routing Models: Port-based and VLAN-based
- Key Guidelines for Implementing IP Routing
- Address Resolution Protocol (ARP)
- ARP Proxy
- Internet Control Message Protocol (ICMP)
- ICMP Redirect
- ICMP Router Discovery
- Broadcast Address
- Directed Broadcast
- Routing Information Protocol (RIP)
- Routing Policies
- Domain Name System (DNS)
- User Datagram Protocol (UDP) Helper
- Standards, Protocols, and Related Reading

You can configure and manage IP routing in either of these ways:

- *From the* `ip` *menu of the Administration Console. See the* Command Reference Guide.
- *From the IP folder of the Web Management software. See the* Web Management User Guide.

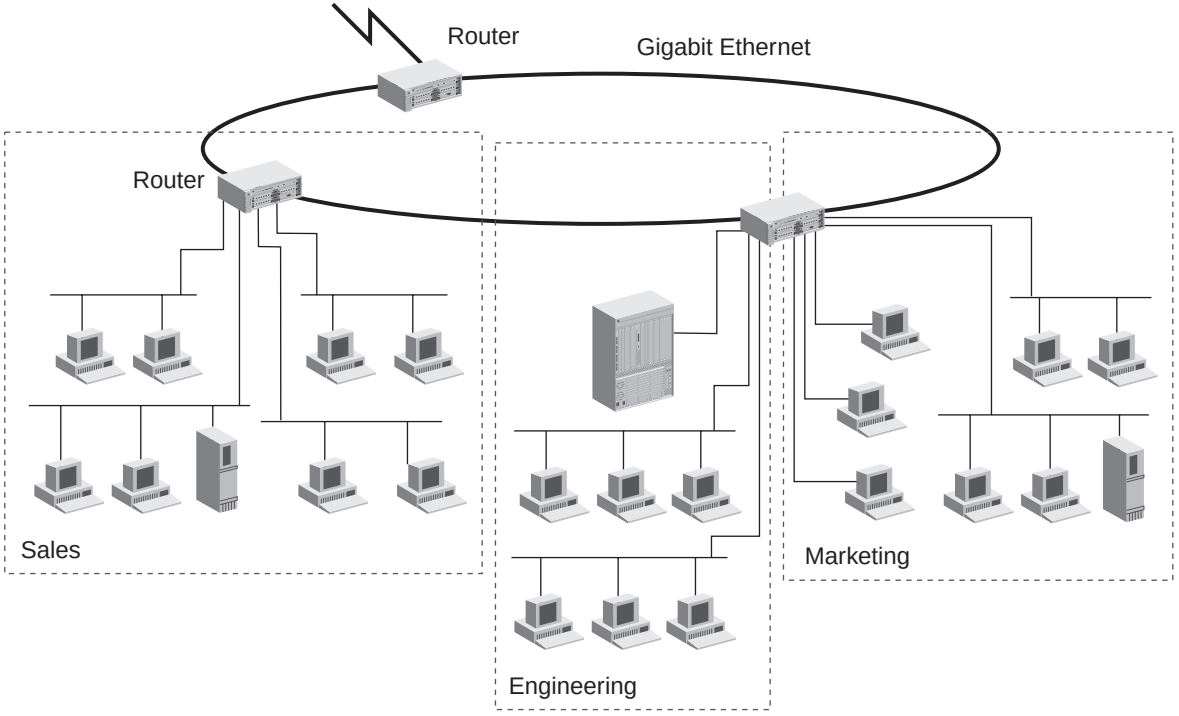
Routing Overview

The term *routing* refers to the action of sending information, in the form of packets, from one network to another. A *router* is the device that accomplishes this task. Your system, as a Layer 3 device, can act as a router. Routers typically:

- Connect enterprise networks.
- Connect subnets (or client/server networks) to the main enterprise network.

Figure 38 shows where routers are typically used in a network. Routing connects subnets to the enterprise network, providing connectivity between devices within a workgroup, department, or building.

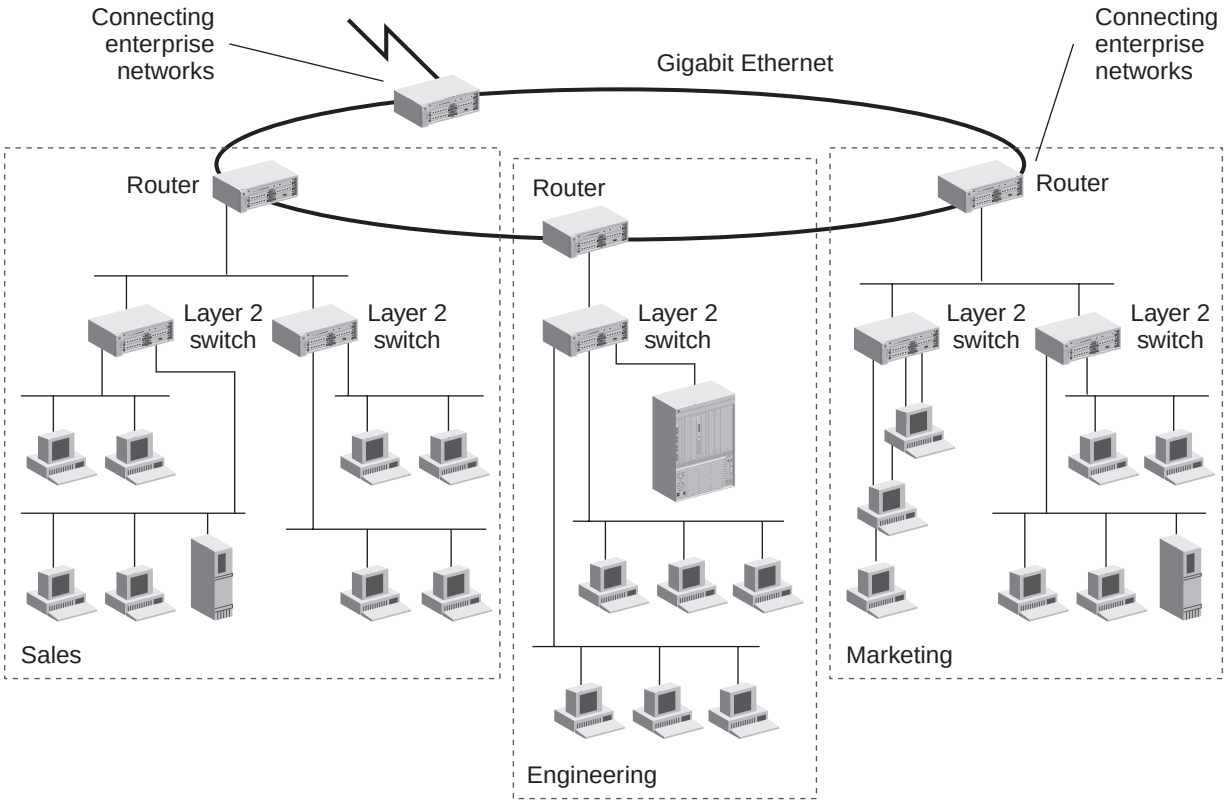
Figure 38 Subnet Routing Architecture



Routing in a Subnetted Environment

Use your system to fit Ethernet switching capability into subnetworked (subnetted) environments. When you put your system into such a network, the system streamlines your network architecture by *routing* traffic between subnets and *switching* within subnets. See Figure 39.

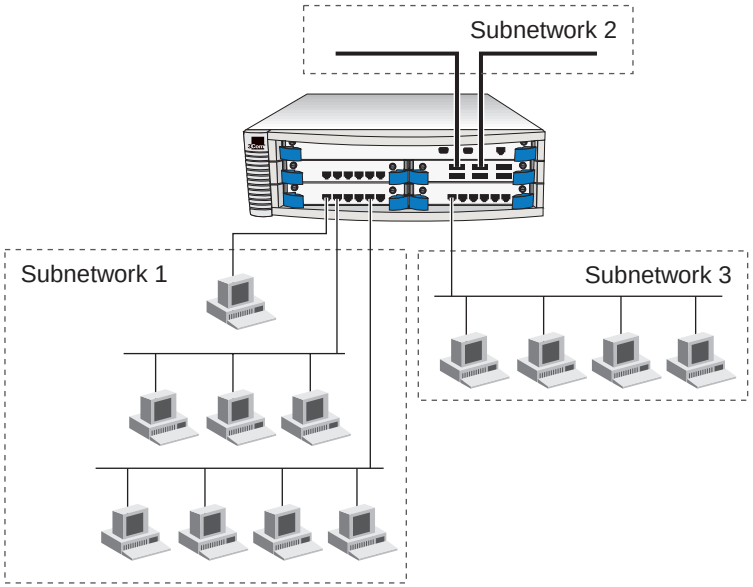
Figure 39 Typical Routing Architecture



Integrating Bridging and Routing

Your system integrates bridging and routing. You can assign multiple ports to each subnet. See Figure 40.

Figure 40 Multiple Ethernet Ports Per Subnet

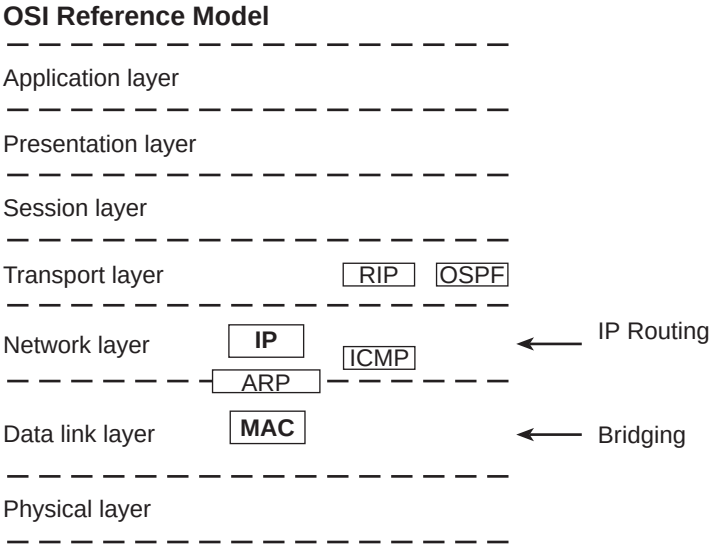


Bridging switches traffic between ports that are assigned to the same subnet. Traffic traveling to different subnets is routed using one of the supported routing protocols. For information about implementing bridging, see Chapter 7.

IP Routing Overview

An IP router, unlike a bridge, operates at the network layer of the Open Systems Interconnection (OSI) Reference Model. The network layer is also referred to as Layer 3. An IP router routes packets by examining the network layer address (IP address). Bridges use data link layer MAC addresses to perform forwarding. See Figure 41.

Figure 41 OSI Reference Model and IP Routing



When an IP router sends a packet, it does not know the complete path to a destination — only the next hop (the next device on the path to the destination). Each hop involves three steps:

1. The IP routing algorithm computes the *next hop* IP address and the next router interface, using routing table entries.
2. The Address Resolution Protocol (ARP) translates the next hop IP address into a physical MAC address.
3. Using the physical MAC address, the router sends the packet out the appropriate bridge port over the network to the next hop.

For more information about IP addresses and next hops, see "IP Addresses" in this chapter.

Features and Benefits

IP routing provides the following features and benefits:

- **Economy** — Because you can connect several segments to the same subnet with routing, you can increase the level of segmentation in your network without creating new subnets or assigning new network addresses. Instead, you can use additional Ethernet ports to expand existing subnets. You do not need to create additional subnets and assign new network addresses to existing hosts.
- **Optimal routing** — IP routing can be the most powerful tool in a complex network setup for sending devices to find the best route to receiving devices. (The best route here means the shortest and fastest route.)
- **Flexibility** — Using IP routing policies and ICMP, you can control the amount, the importance, and the type of traffic on your network. (Routing policies and ICMP are discussed later in this chapter.)
- **Resiliency** — If a router in the network goes down, the other routers update their routing tables to compensate for this occurrence; in a typical case, there is no need for you to manually intervene.

Key Concepts

IP routers use the following elements to transmit packets:

- Multiple IP Interfaces per VLAN
- Media Access Control (MAC) addresses
- Network addresses
- IP addresses
- Router interfaces
- Routing tables
- Address Resolution Protocol (ARP)
- Internet Control Message Protocol (ICMP)

Multiple IP Interfaces per VLAN

You can overlap IP interfaces without configuring a separate VLAN for each subnet. Multiple IP interfaces can share the same VLAN, allowing multiple subnets to be routed on the same 802.1Q VLAN.

You can define up to 32 IP interfaces on the system. This includes IP routing interfaces for static VLANs, IP VLANs created by router ports or any combination of static VLANs and router port IP VLANs.

If you define multiple interfaces for an IP VLAN, you cannot subsequently modify that IP VLAN to supply Layer 3 address information. If only one routing interface is defined for the IP VLAN, then you can supply Layer 3 address information as long as it matches the Layer 3 information that is specified for the routing interface. This action converts the IP VLAN into a network-based VLAN.

If you use network-based VLANs, you are limited to defining only *one* IP routing interface for that VLAN. When you define an IP routing interface for a static VLAN already configured, the system will not allow you to select a network-based IP VLAN that already has a routing interface defined for it.

If you add or change more than one IP interface associated with the same VLAN, disable the ICMP Redirect option for optimal performance.

Media Access Control (MAC) Address

The MAC address refers to a physical hardware address. On a LAN, the MAC address is the unique hardware number of your device. The MAC address on an Ethernet LAN is the same as your Ethernet address.

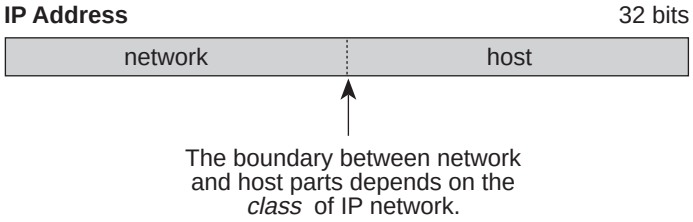
Network-Layer Address

The network-layer address refers to a logical address that applies to a specific protocol. A network-layer address exists at Layer 3 of the OSI reference model.

IP Addresses

IP addresses are 32-bit addresses that consist of a *network part* (the address of the network where the host is located) and a *host part* (the address of the host on that network). See Figure 42.

Figure 42 IP Address: Network Part and Host Part



IP addresses differ from Ethernet and Fiber Distributed Data Interface (FDDI) MAC addresses, which are unique hardware-configured 48-bit addresses. A central agency assigns the network part of the IP address, and you assign the host part. All devices that are connected to the same network share the same network part (also called the *prefix*).

Dotted Decimal Notation

The actual IP address is a 32-bit number that is stored in binary format. These 32 bits are segmented into 4 groups of 8 bits — each group is referred to as a *field* or an *octet*. Decimal notation converts the value of each field into a decimal number, and the fields are separated by dots.

Figure 43 Dotted Decimal Notation for IP addresses

```
10011110.01100101.00001010.00100000     = Binary notation

                    158.101.10.32       = Decimal notation
```

The decimal value of an octet whose bits are all 1s is 255.

Network Portion

The location of the boundary between the network part and the host part depends on the class that the central agency assigns to your network. The three primary classes of IP addresses are A, B, and C:

- **Class A address** — Uses 8 bits for the network part and 24 bits for the host part. Although only a few Class A networks can be created, each can contain a very large number of hosts.
- **Class B address** — Uses 16 bits for the network part and 16 bits for the host part.
- **Class C address** — Uses 24 bits for the network part and 8 bits for the host part. Each Class C network can contain only 254 hosts, but many such networks can be created.

The high-order bits of the network part of the address designate the IP network class. See Table 38.

Table 38 How Address Class Corresponds to the Address Number

Address Class	High-order Bits	Address Number (Decimal)
A	0nnnnnnn	0-127
B	10nnnnnn	128-191
C	11nnnnnn	192-254

Subnet Portion

The IP address can also contain a *subnet part* at the beginning of the host part of the IP address. Thus, you can divide a single Class A, B, or C network internally, allowing the network to appear as a single network to other external networks. The subnet part of the IP address is visible only to hosts and gateways on the subnet.

When an IP address contains a subnet part, a *subnet mask* identifies the bits that constitute the subnet address and the bits that constitute the host address. A subnet mask is a 32-bit number in the IP address format. The *1* bits in the subnet mask indicate the network and subnet part of the address. The *0* bits in the subnet mask indicate the host part of the IP address, as shown in Figure 44.

Figure 44 Subnet Masking

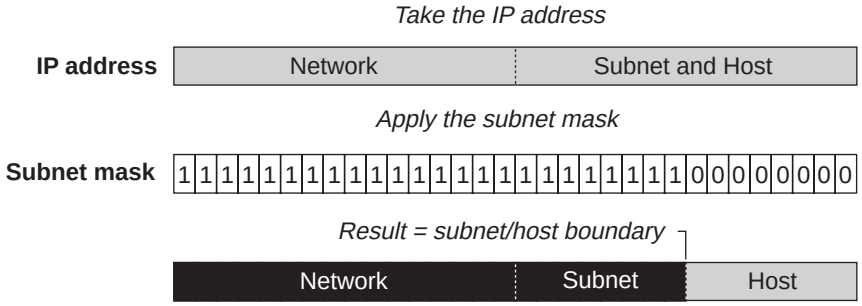


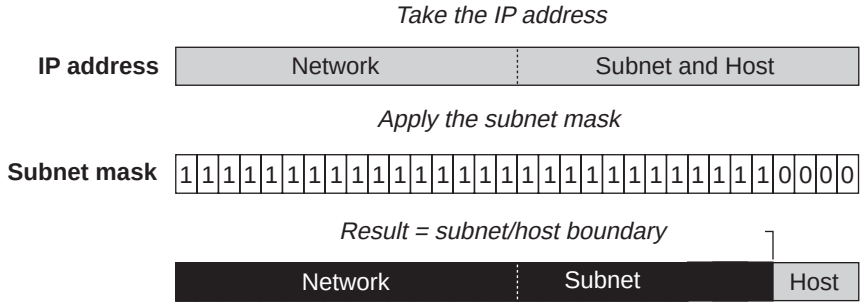
Figure 45 shows an example of an IP address that includes network, subnet, and host parts. Suppose the IP address is *158.101.230.52* with a subnet mask of *255.255.255.0*. Since this is a Class B address, this address is divided as follows:

- *158.101* is the network part
- *230* is the subnet part
- *52* is the host part

As shown in this example, the 32 bits of an IP address and subnet mask are usually written using an integer shorthand. This notation translates four consecutive 8-bit groups, octets, into four integers that range from 0 through 255. The subnet mask in the example is written as 255.255.255.0.

Traditionally, subnet masks were applied to octets in their entirety. However, one octet in the subnet mask can be further subdivided so that part of the octet indicates an *extension* of the network number, and the rest of the same octet indicates the host number, as shown in Figure 45.

Figure 45 Extending the Network Prefix



Using the Class B IP address from our example (158.101.230.52), the subnet mask is 255.255.255.240.

The number that includes both the Class B natural network mask (255.255) and the subnet mask (255.240) is sometimes called the *extended network prefix*.

Continuing with the previous example, the subnet part of the mask uses 12 bits, and the host part uses the remaining 4 bits. Because the octets are actually binary numbers, the number of subnets that are possible with this mask is 4,094 (2^{12}), and the number of hosts that are possible in each subnet is 16 (2^4).

Subnet Mask Numbering

There is an alternate method to represent the subnet mask numbers. This method is based on the fact that the subnet mask numbers are based on the number of bits that signify the network portion of the mask. Many Internet Service Providers (ISP) providers now use this notation to denote the subnet mask. See Table 39.

Table 39 Subnet Mask Notation

Standard Mask Notation	Network Prefix Notation
100.100.100.100 (255.0.0.0)	100.100.100.100/8
100.100.100.100 (255.255.0.0)	100.100.100.100/16
100.100.100.100 (255.255.255.0)	100.100.100.100/24

The subnet mask 255.255.255.255 is reserved as the default broadcast address.

Variable Length Subnet Masks (VLSMs)

With Variable Length Subnet Masks (VLSMs), each subnet under a network can use its own subnet mask. Therefore, with VLSM, you can get more subnet space out of your assigned IP address space.

How VLSMs Work

VLSMs get beyond the restriction that a single subnet mask imposes on the network. One subnet mask per IP network address fixes the number of subnets and the number of hosts per subnet.

For example, if you decide to configure the 158.100.0.0/16 network with a /23 extended-network-prefix, you can create 128 subnets with each having up to 510 hosts. If some of the subnets do not need that many hosts, you would assign many host IP addresses but not use them.

With VLSMs, you can assign another subnet mask, for instance, /27, to the same IP address. So you can assign a longer subnet mask that consequently uses fewer host IP addresses. As a result, routing tables are smaller and more efficient.

This method of further subdividing addresses using VLSMs is being used increasingly more as networks grow in size and number. However, be aware that this method of addressing can greatly increase your network maintenance and the risk of creating erroneous addresses unless you plan the addressing scheme properly.

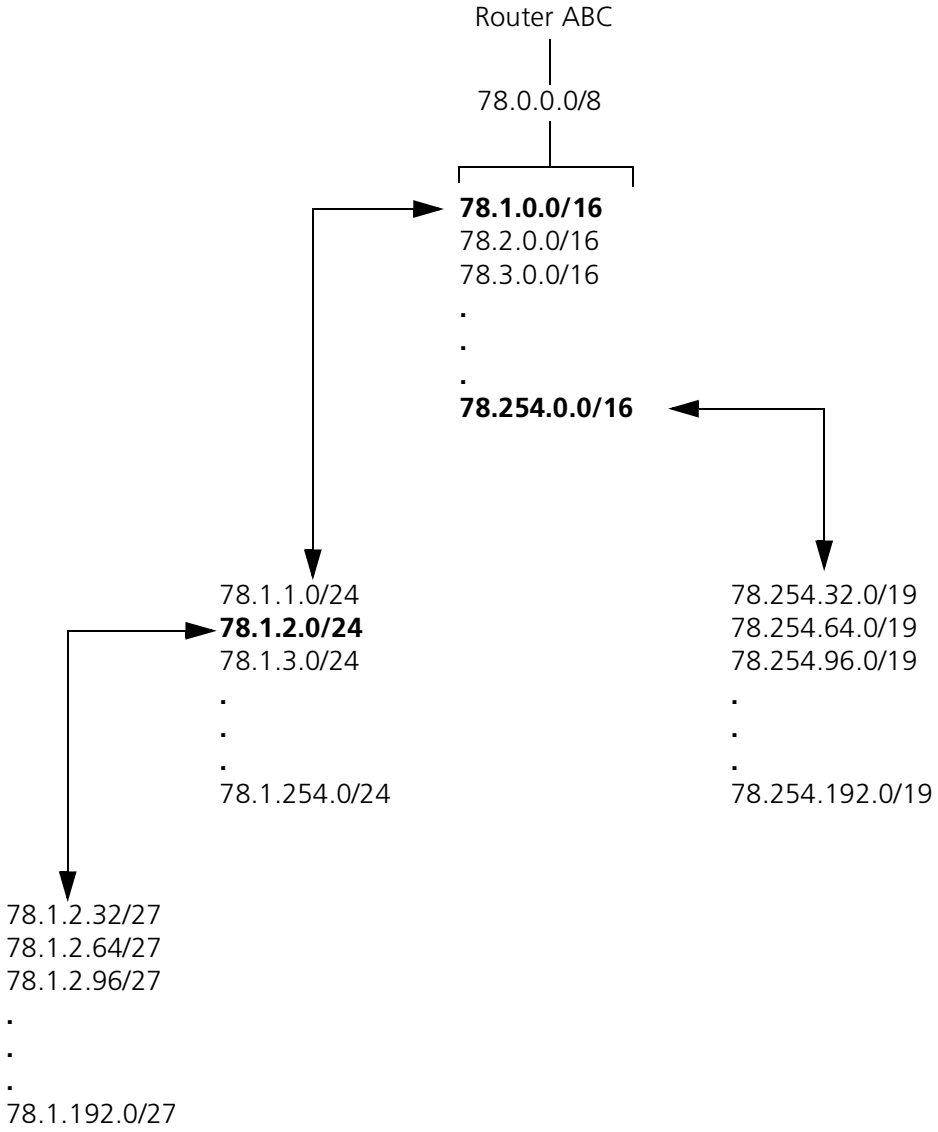
Route Aggregation

Route aggregation is a numbering scheme in which you can significantly reduce the total number of IP addresses that your organization uses.

With route aggregation, if you set up the IP address numbering scheme, including VLSMs, for the network properly, one network advertisement can reach all of the subnets that report to that network. Similarly, one subnet broadcast address can reach the subnets that report to the subnet. In this way, IP traffic that is addressed to a specific part of the network is advertised only to that part of the network.

For example, the Router ABC in Figure 46 can summarize any number of subnets that report to it into a single advertisement, and advertise a single route into the Internet routing table: 78.0.0.0/8. Similarly, the subnet 78.1.0.0 can advertise to its subnets with one advertisement, and can summarize advertisements from its subnets by means of a single advertisement.

Figure 46 Example of Route Aggregation



If you plan your subnet addresses carefully, you can improve your utilization of IP addresses and your routing tables will be easier to maintain.

Also, with route aggregation, you can forward RIP-2 updates to RIP-1 routers by "rolling up" or aggregating the subnet addresses into a single advertisement.

See the RIP-1 versus RIP-2 discussion later in this chapter.

Go to **`http://www.3com.com/technology/tech_net/white_papers`** for a thorough discussion of IP addressing, VLSMs, and route aggregation.

Guidelines for Using VLSMs

Consider the following guidelines when you implement VLSMs:

- When you design the subnet scheme for your network, do not estimate the number of subnets and hosts that you need. Work from the top down until you are sure that you have accounted for all the hosts, present and future, that you need. When you design the network, set up the address numbers so that you can take advantage of route aggregation.
- Use RIP-2 or Open Shortest Path First (OSPF) to carry the extended network prefix information with each route advertisement.
- Make sure that the routers forward routes based on what is known as the *longest match*.

  For example, assume that the destination IP address of a packet is 158.101.26.48 and that the following four routes are in the routing table:

  - 158.101.26.0/24
  - 158.101.3.10/16
  - 158.101.26.32/16
  - 158.95.80.0/8

  The router selects the route to 158.101.26.0/24 because its extended network prefix has the greatest number of bits that correspond to the destination IP address of the packet.
- To take advantage of route aggregation, use addresses that reflect the *actual network topology*, as illustrated in Figure 46. This is one of the best things you can do to increase routing efficiency.

For more information about understanding and using VLSMs, see RFC 1219 and RFC 1878.

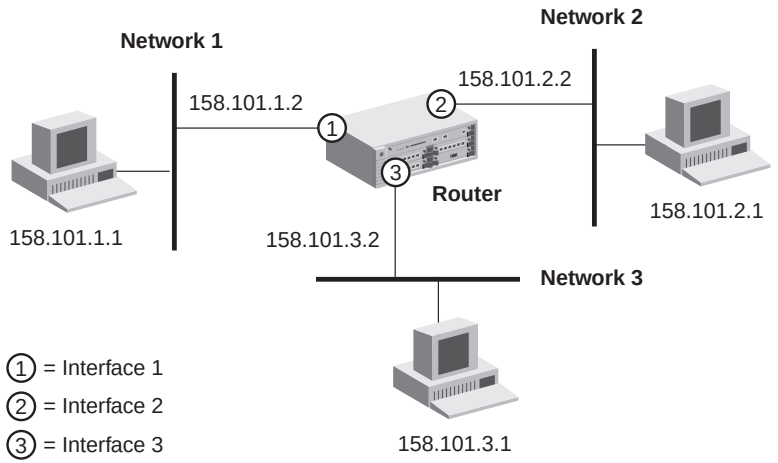
Router Interfaces

A router interface connects the router to a subnet. If you use your system for IP routing, more than one port can connect to the same subnet.

Each router interface has an IP address and a subnet mask. This router interface address defines both the number of the network to which the router interface is attached and its host number on that network. A router interface IP address serves two functions:

- Sending IP packets to or from the router
- Defining the network and subnet numbers of the segment that is connected to that interface

Figure 47 Routing Interfaces



To gain access to the system using TCP/IP or to manage the system using Simple Network Management Protocol (SNMP), set up an IP interface to manage your system, either in-band (with your regular network traffic) or out-of-band (with a dedicated network).

- **In-Band Management** — Set up an IP routing interface and at least one virtual LAN (VLAN). See Chapter 9 for information about how to define a VLAN.
- **Out-of-Band Management** — Assign an IP address and subnet mask for the out-of-band Ethernet port on your system through the Administration Console or through the Web Management system. (See "IP Addresses" earlier in this chapter for background information about IP addresses and subnet masks.) The out-of-band Ethernet port is the 10BASE-T port on the system processor module. It is not associated with a port number.

Routing Table

With a routing table, a router or host determines how to send a packet toward its ultimate destination. The routing table contains an entry for every learned and locally defined network. The size of the routing table on your system is dynamic and can hold at least 25,600 entries; the actual number depends upon what other protocols are being routed.

A router or host uses the routing table when the destination IP address of the packet is not on a network or subnet to which it is directly connected. The routing table provides the IP address of a router that can forward the packet toward its destination.

The routing table consists of the following elements:

- **Destination IP address** — The destination network, subnet, or host.
- **Subnet mask** — The subnet mask for the destination network.
- **Metric** — A measure of the distance to the destination. In the Routing Information Protocol (RIP), the metric is the number of hops through routers.
- **Gateway** — The IP address of the router interface through which the packet travels on its next hop.
- **Status** — Information that the routing protocol has about the route, such as how the route was put into the routing table.
- **Time-to-live (TTL)** — Time-to-live measured in seconds before this learned route will time out.

Figure 48 shows the routing table contents of the router in Figure 47.

Figure 48 Sample Routing Table

Routing table

Destination	Subnet mask	Metric	Gateway	Status	TTL
default route	255.255.255.0	2	160.1.1.254	learned - RIP	170
158.101.1.0	255.255.255.0	2	160.1.1.254	learned - OSPF - INTRA	---
158.101.2.0	255.255.255.0	2	160.1.1.254	learned - OSPF - INTRA	---
158.101.3.0	255.255.255.0	2	160.1.1.254	learned - OSPF - INTRA	---

Routing table data is updated statically or dynamically:

- **Statically** — You manually enter static routes in the routing table. Static routes are useful in environments where no routing protocol is used or where you want to override some of the routes that are generated with a routing protocol. Because static routes do not automatically change in response to network topology changes, manually configure only a small number of reasonably stable routes. Static routes do not time out, but they can be learned.
- **Dynamically** — Routers use a protocol such as RIP or OSPF to automatically exchange routing data and to configure their routing tables dynamically. Routes are recalculated at regular intervals. This process helps you to keep up with network changes and allows the system to reconfigure routes quickly and reliably. Interior Gateway Protocols (IGPs), which operate within networks, provide this automated method.

Default Route

In addition to the routes to specific destinations, a routing table can contain a *default route*. The router uses the default route to forward packets that do not match any other routing table entry.

A default route is often used in place of static routes to numerous destinations that all have the same gateway IP address and interface number. The default route can be configured statically, or it can be learned dynamically.

A drawback to implementing a default static route is that it is a single point of failure on the network. You can implement Virtual Router Redundancy Protocol (VRRP) on your network to remedy this problem. For more information about VRRP, see Chapter 12.

Routing Models: Port-based and VLAN-based

There are two basic routing models for implementing how a bridge and a router interact within the same 3Com switch. They are:

- **Port-based routing (routing versus bridging)** — The system first tries to route packets that belong to recognized protocols, and all other packets are bridged.
- **VLAN-based routing (routing over bridging)** — The system first tries to determine if the frame will be switched or routed. The system does this by examining the destination MAC address:
  - If the destination MAC address is *not* an internal MAC address, then the frame must be switched and is forwarded according to the IEEE 802.1D protocol.
  - If the destination MAC address *is* an internal MAC address, the frame is further examined to determine if the frame is a routed frame (Layer 3) or a request to the switch itself (Layer 2).

  This model allows the system to give the frame first to Layer 2 to be bridged by the VLAN, and then given to the router only if the frame cannot be bridged. This scheme gives you the flexibility to define router interfaces on top of several bridge ports.

Your system, as a Layer 3 routing device, has the ability to implement either type of routing scheme, "routing over bridging" and "routing versus bridging." Each kind of routing scheme requires its own interface type:

- **Routing over Bridging requires a VLAN-based IP Interface** — A VLAN-based interface requires you to first configure a VLAN and then create a router interface over that VLAN.
- **Routing versus Bridging requires a Port-based IP Interface** — A port-based interface requires you to configure a router interface on top of a single physical port.

Role of VLANs in IP Routing

It is important to keep in mind that, except for the out-of-band management port, there is a VLAN index associated with every IP interface, whether the interface is port-based or VLAN-based:

- **Port-based router interface** — The system creates a VLAN index and associates it with the interface as you define the router interface.

  The hardware requires that every packet be associated with a VLAN ID. Even though port-based packets do not use VLAN software, the hardware must recognize the packets as VLAN entities. Therefore, the system configures every router port interface as a single port, nontagged, protocol-based VLAN.

- **VLAN-based router interface** — You explicitly create a VLAN index, then define the IP interface and associate the interface with the index.

In addition, port-based routing requires the VLANs on a system to be in AllClosed mode. This potentially has major effects on your network configuration. For more information about allClosed and allOpen mode with regard to IP routing, see "Important Considerations" later in this chapter.

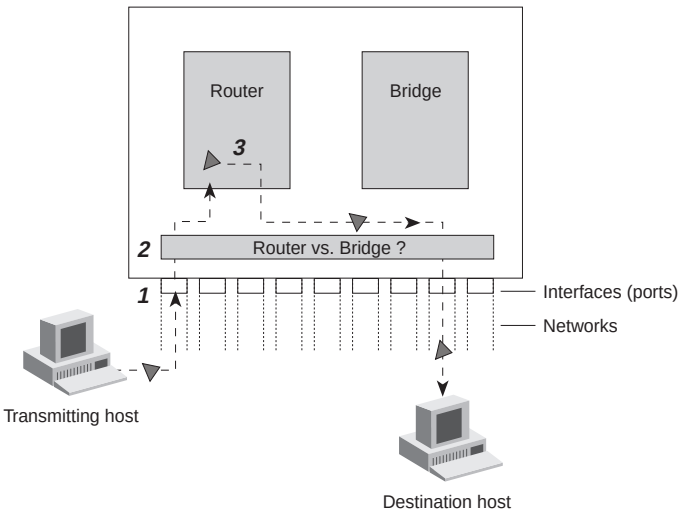
Port-based Routing

In the communications industry, Layer 3 devices have traditionally employed port-based routing: routed packets over interfaces that are associated with a *single physical port*.

Figure 49 illustrates traditional routing:

1. The packet enters the switch.
2. The bridge or router determines that the packet belongs to a recognized routing protocol, so the packet is passed to the router.
3. The router examines the destination network address and forwards the packet to the interface (port) that is connected to the destination subnetwork.

Figure 49 Routing versus Bridging

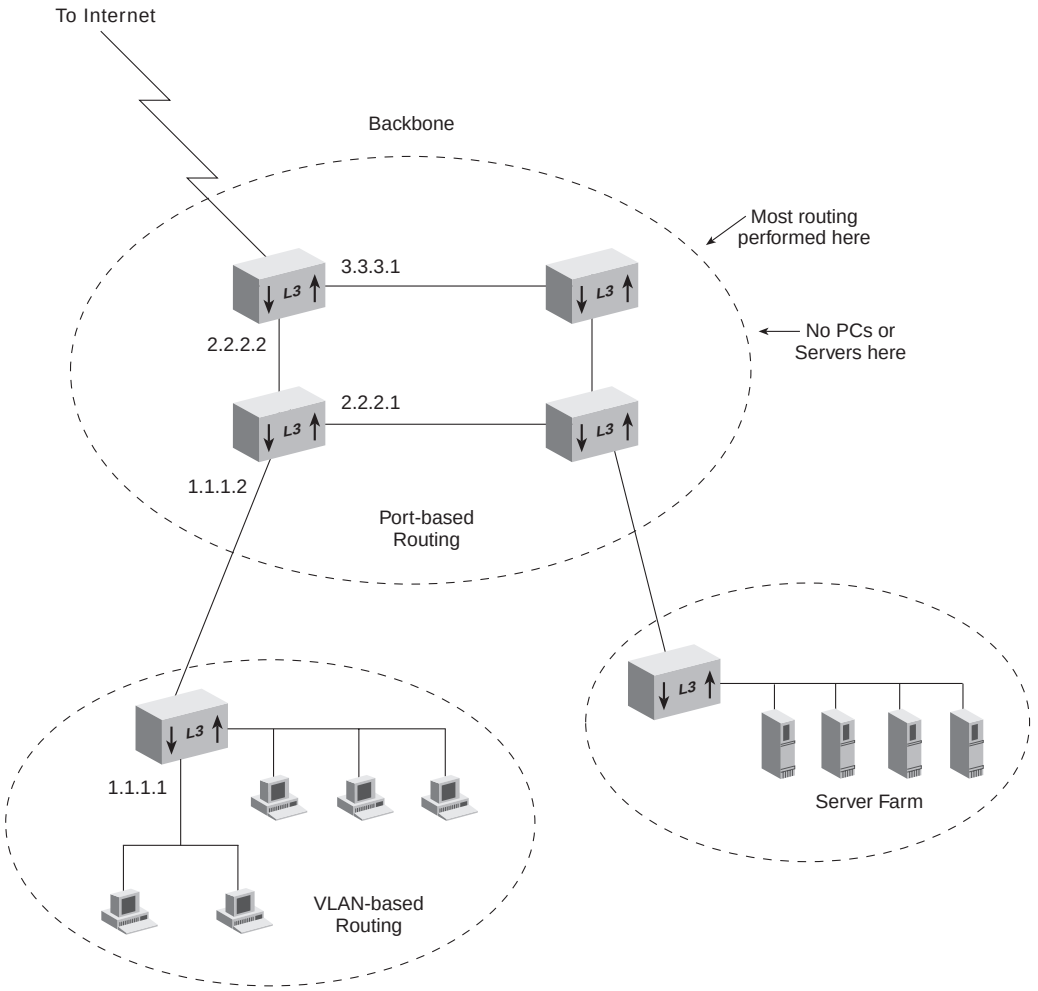


Port-based routing is advantageous for networks or network segments whose emphasis is heavily on routing rather than bridging.

Port-based Routing Examples

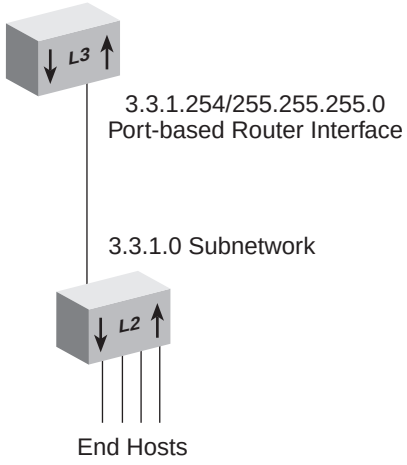
In Figure 50, four Layer 3 switches act as the campus backbone. Because very little bridging takes place within the backbone, port-based routing actually makes operations more efficient.

Figure 50 Port-based Routing (Backbone Operations)



In Figure 51, a Layer 2 switch is acting as a port aggregator for the corporate or campus VLAN. Because the traffic going from the Layer 2 switch to the Layer2/Layer 3 switch is only going to be routed, port-based routing between these two devices is more efficient.

Figure 51 Port-based Routing (Port Aggregation)



Benefits of Port-based Routing

Because the bridge and router topologies are different, Spanning Tree Protocol (STP) can build a spanning tree that is not associated with the routing topology.

This is especially useful if you want to configure two routers in parallel and let the routing protocol or protocols manage the routing loop while STP simultaneously manages any potential bridge loop. This redundant router configuration is widely used to incorporate:

- Network redundancy.
- Load sharing between two routers on the same two LANs.

Limitations of Port-based Routing

The system can only supply to the interface the physical bandwidth limit of the port, since the interface is strictly associated with one physical port.

Important Considerations

Be aware of the following points when you use port-based routing:

- Your system can be in only allOpen or allClosed VLAN mode. You cannot create mixed VLAN modes on the same device.
- You can establish up to 32 IP interfaces on a single VLAN.
- If you set up your IP router as a port-based router, you are not required to use 802.1Q tagging. The default VLAN that the system creates for a port-based router interface does not enable tagging by default.
- If you set up your IP router as a port-based router, you determine whether or not you want unknown destination packets flooding that port. To prevent the possibility of flooding unknown destination packets onto that port, you must manually remove the port-based router interface from the default VLAN.
- A port can either be part of a VLAN interface or part of a router port interface, but never both.

  Once you have configured a port to be part of a protocol VLAN, you cannot use it to create a router port interface *of the same protocol*, and vice versa. However, if a port is part of an IPX VLAN, then you can use that port to create an IP router port interface.
- In order to support router port IP interfaces, the system must operate in allClosed VLAN mode:
  - To prevent MAC addresses from being shared between other VLANs and the router port
  - To ignore Spanning Tree states on the router port

    This means you can configure loops among router port interfaces and have the routing protocol make decisions rather than being governed by the bridge.
- The current default mode for VLANs is allOpen. In changing from allOpen to allClosed mode, you must reconfigure your VLANs and router interfaces.
- allClosed mode prevents unicast traffic from being forwarded (bridged) between VLANs.
- In allClosed mode, you must define router interfaces in order to pass traffic between VLANs. A traditional router allows the user to configure whether or not bridging is allowed. To disable bridging on a router port interface, the user will be required to remove that port from the default VLAN.

- You can only remove a VLAN associated with a router port interface using `ip interface remove`. If you try to remove the VLAN using `bridge vlan remove`, an error is returned. This protects the router port VLAN from inadvertent deletion.
- The ip interface summary must display the parameters of both VLAN-based and port-based IP interfaces. For this reason, the field labeled `ID` displays the VLAN interface index (if the interface type equals `VLAN`) or displays the physical port number (if the interface type equals `Port`).

VLAN-based Routing

In VLAN-based routing, the destination MAC address determines whether the system bridges or routes a packet. Before a host system sends a packet to another host, the host system compares its own network address to the network address of the other host as follows:

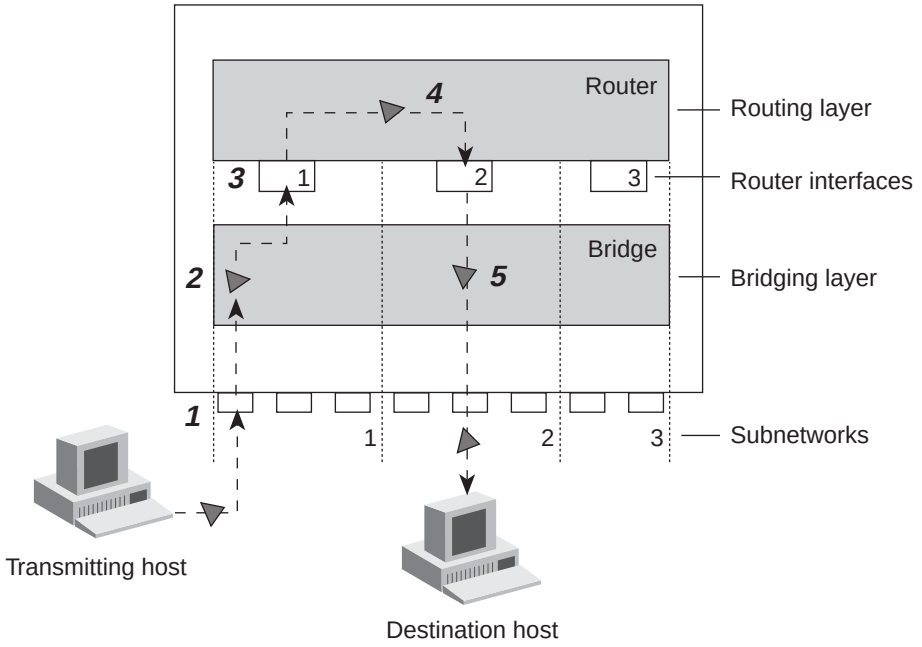
- If network addresses are on the same subnet, the packet is bridged directly to the destination address of the host.
- If network addresses are on different subnets, the packet must be routed from one to the other. In this case, the host sends an ARP request for its gateway MAC address, then transmits the packet using the MAC address of the gateway.

Figure 52 illustrates the process when the packet is routed:

1. The packet enters the system.
2. The bridging layer examines the destination MAC address of the packet. The destination MAC address corresponds to the address of one of the system ports that are configured for routing (as opposed to a learned end station address).
3. The packet is passed to the router interface that is associated with the port where the packet was received.
4. The routing layer:
   - **a** Selects a destination interface based on the destination network address
   - **b** Determines the MAC address of the next hop (either the destination host or another gateway)
   - **c** Passes the packet back to the bridging layer using the destination MAC address of the next hop

5 The bridging layer then selects a segment (port) based on the destination MAC address and forwards the packet to that segment.

Figure 52 3Com VLAN-based Routing Model



Benefits of VLAN-based Routing

If your network traffic is apt to be more mixed between routing and bridging, VLAN-based routing permits your system to act as both a bridge and a router, adding both port density and bandwidth to any interface without requiring any additional hardware.

Limitations of VLAN-based Routing

With VLAN-based routing, the router is subservient to the bridge. The bridge topology dictates the router topology. Router loops rely on the bridge to resolve them, not the routing protocols. For large amounts of IP traffic, this situation is not ideal.

Key Guidelines for Implementing IP Routing

To route network traffic using IP, you must perform these tasks in the following order:

1. Configure trunks (optional).
2. Configure IP VLANs (VLAN-based routing).
3. Establish your IP interface.
4. Enable IP routing.

Configure Trunks (Optional)

Trunks (also known as aggregated links) work at Layer 2 and allow you to combine multiple Fast Ethernet, Gigabit Ethernet, or FDDI ports into a single high-speed link between two switches.

If you intend to use trunking on an IP device, configure your trunks *before* you set up VLANs and IP interfaces. In this case, you must specify the anchor port (the lowest-numbered port) to associate with the trunk. For example, if ports 7 through 12 are associated with a trunk, specifying 7 to 12 defines the VLAN to include all of the physical ports in the trunk (ports 7 through 12).

For more information about trunking, see Chapter 8.

Configure IP VLANs (VLAN-based Routing)

If you want to use VLAN-based routing, you must first configure the VLAN to use IP. An IP VLAN is called a *protocol-based VLAN*. Protocol-based VLANs such as IP VLANs group one or more switch ports together for one or more specified Layer 3 protocols. (You can also create network-based VLANs which are IP VLANs grouped according to the IP network address and mask.)

If you want to use port-based routing, you do not have to explicitly configure a VLAN; the VLAN index is created automatically when you define the IP interface.

See Chapter 9 in this guide to learn more about VLANs.

Establish Your IP Interfaces

To establish an IP interface, follow these steps:

1. Determine your interface parameters.
2. Define the IP interfaces.

Interface Parameters

Each IP routing interface has these standard characteristics:

- **IP address** — An address from the range of addresses that the Internet Engineering Task Force (IETF) assigns to your organization. This address is specific to your network and system.
- **Subnet mask** — The 32-bit number that uses the same format and representation as an IP address. The subnet mask determines which bits in the IP address are interpreted as the network number, the subnet number, and the host number. Each IP address bit that corresponds to a 1 in the subnet mask is in the network/subnet part of the address. Each IP address bit that corresponds to a 0 is in the host part of the IP address.
- **Type** — The type of interface: VLAN or port.
- **VLAN interface index (for VLAN-based routing)** — The number of the IP VLAN that is associated with the IP interface. When the system prompts you for this option, the menu identifies the available VLAN indexes.
- **Bridge port (for port-based routing)** — The number of the physical port associated with this IP interface.

Important Considerations

Consider the following issues before you establish an IP interface:

- You should have already mapped out the entire network and subnet IP addressing scheme before you assign IP addresses. This includes planning for future expansion of address numbers.
- If you can, take advantage of out-of-band IP interfaces. An out-of-band interface does not need to use bandwidth from a port used for network traffic in order to manage the network. Also, out-of-band management makes it easier to troubleshoot if there is a problem with the network.
- You cannot create a port-based router interface on a system that has existing VLANs in AllOpen mode unless you choose to have the system remove all existing VLANs and redefine the default VLAN.

- You must define a router interface if your system is in allClosed mode and want to forward traffic between VLANs.
- In allClosed mode, the system does not forward unicast traffic.

The `ip interface define` (in-band) and `management ip interface define` (out-of-band) options are documented in the *Command Reference Guide*. To learn how to use the Web Management Console to set up IP interfaces, see the *Web Management User Guide*.

Defining an IP Interface

After you determine the characteristics for each IP interface, you are ready to define each interface. You can use the Administration Console or the Web Management Console to define an IP interface.

Your system can contain up to 32 IP interfaces. These interfaces can be defined all on one VLAN, all on one router port, or on any combination of VLANs and router ports.

To define your IP interface, decide if and how to implement the following IP features:

- ARP proxy
- ICMP
- ICMP Redirect
- ICMP Router Discovery
- Broadcast address
- Directed broadcast (broadcast forwarding)
- RIP
- Routing policies
- DNS
- UDP Helper

These features are discussed later in this chapter.

Enable IP Routing

To enable IP routing, use the `ip routing` command on the Administration Console or use the IP Configuration form in the Web Management software. By default, IP routing is disabled on the system.

You can use the Routing Information Protocol (RIP) or the Open Shortest Path First (OSPF) protocol to take advantage of routing capabilities. RIP is discussed in this chapter; OSPF is discussed in Chapter 14.

Administering IP Routing

Keep these points in mind while you administer the IP network:

- Flush the ARP cache regularly if you set the age time to 0.
- Set up a default route.

  The system uses the default route to forward packets that do not match any other routing table entry. You may want to use the default route in place of routes to numerous destinations that all have the same gateway IP address. If you do not use a default route, ICMP is more likely to return an "address not found" error.
- Before you can define static routes, you must define at least one IP interface. See the section "Defining an IP Interface" for more information. Remember the following guidelines:
  - Static routes remain in the routing table until you remove them or the corresponding interface.
  - Static routes take precedence over dynamically learned routes to the same destination.
  - Static routes are included in periodic RIP updates sent by your system.

Address Resolution Protocol (ARP)

ARP is a low-level protocol that locates the MAC address that corresponds to a given IP address. This protocol allows a host or router to use IP addresses to make routing decisions while it uses MAC addresses to forward packets from one hop to the next.

You do not need to implement ARP — your system has ARP capability built in, but you can manipulate and display the contents of the ARP cache.

When the host or router knows the IP address of the *next* hop towards the packet destination, the host or router translates that IP address into a MAC address before sending the packet. To perform this translation, the host or router first searches its *ARP cache*, which is a table of IP addresses with their corresponding MAC addresses. Each device that participates in IP routing maintains an ARP cache. See Figure 53.

Figure 53 Example of an ARP Cache

ARP cache

IP address	MAC address
158.101.1.1	00308e3d0042
158.101.2.1	0080232b00ab

If the IP address does not have a corresponding MAC address, the host or router broadcasts an *ARP request* packet to all the devices on the network. The ARP request contains information about the target and source addresses for the protocol (IP addresses). See Figure 54.

Figure 54 Example of an ARP Request Packet

ARP request packet

00802322b00ad	Source hardware address
158.101.2.1	Source protocol address
?	Target hardware address
158.101.3.1	Target protocol address

When devices on the network receive this packet, they examine it. If their address is not the target protocol address, they discard the packet. When a device receives the packet and confirms that its IP address matches the target protocol address, the receiving device places its MAC address in the target hardware address field and sends the packet back to the source hardware address. When the originating host or router receives this *ARP reply*, it places the new MAC address in its ARP cache next to the corresponding IP address. See Figure 55.

Figure 55 Example of ARP Cache Updated with ARP Reply

ARP cache

IP address	MAC address
158.101.1.1	00308e3d0042
158.101.2.1	0080232b00ab
158.101.3.1	0134650f3000

After the MAC address is known, the host or router can send the packet directly to the next hop.

Important Considerations

Keep the following things in mind about this protocol:

- Enter a static ARP entry when the ARP resolution does not result in an ARP entry in the cache. For example, some applications do not respond to ARP requests and, consequently, specific network operations may time out for lack of address resolution.
- Enter a static ARP entry in a test environment if your test analyzer cannot respond to an ARP request.
- Setting an ARP cache age time of zero (no aging) is useful in the middle of lengthy tests so that ARP requests do not have to be issued.

  If you do set an ARP cache age time of zero, be aware that the ARP cache can quickly grow in size and consume system resources. In this case, be sure to flush the ARP cache after your tests are complete.

ARP Proxy

ARP proxy allows a host that has no routing ability to determine the MAC address of a host on another network or subnet.

When ARP proxy is enabled and a workstation sends an ARP request for a remote network, the system determines if it has the best route and then answers the ARP request by sending its own MAC address to the workstation. The workstation then sends the frames for the remote destination to the system, which uses its own routing table to reach the destination on the other network.

Important Considerations

Consider the following issues with ARP proxy:

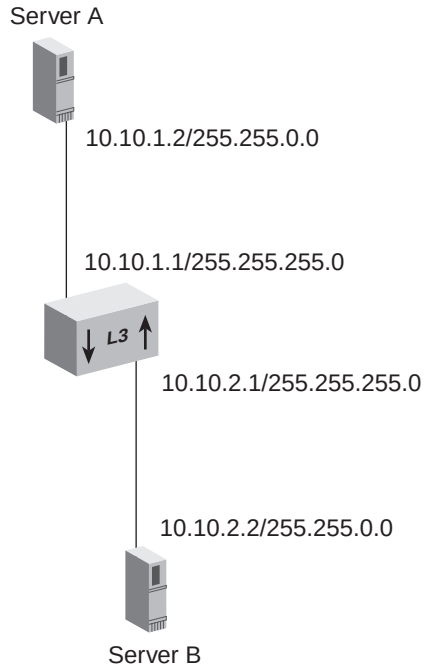
- Do not use ARP proxy if you are using VLSMs because ARP proxy works by seeing the entire network configuration as one network.
- ARP proxy increases ARP traffic to handle the increased mapping of IP addresses to MAC addresses.

Example

In the following example, Server A cannot use the router as a gateway to Server B (if ARP proxy is disabled) because Server A has its subnet mask set to broadcast (using ARP) its IP network address as 158.101.0.0, while the IP network address of the router is 158.101.1.0.

However, if the router has ARP proxy enabled, the router answers the request of Server A with its own MAC address — thus, all traffic sent to Server B from Server A is addressed to the corresponding IP interface on the router and forwarded appropriately.

Figure 56 Proxy ARP



With ARP proxy enabled on the router, the MAC address of IP interface 10.10.1.1 is returned to server A when server A sends an ARP message for Server B's MAC address.

Internet Control Message Protocol (ICMP)

Because a router knows only about the next network hop, it is not aware of problems that may be closer to the destination. Destinations may be unreachable if:

- Hardware is temporarily out of service.
- You specified a nonexistent destination address.
- The routers do not have a route to the destination network.

To help routers and hosts discover problems in packet transmission, a mechanism called Internet Control Message Protocol (ICMP) reports errors back to the source when routing problems occur. With ICMP, you can determine whether a failure resulted from a local or a remote problem.

ICMP performs these tasks:

- Tests whether nodes can be reached (ICMP Echo Request and ICMP Echo Reply)

  A host or gateway sends an ICMP echo request to a specified destination. If the destination receives the echo request, it sends an ICMP echo reply to the sender. This process tests whether the destination is reachable and responding and verifies that the network transport hardware and software are working. The `ping` option is frequently used to invoke this process.

- Creates more efficient routing (ICMP Redirect)

  Often the host route configuration specifies the minimum possible routing data that is needed to communicate (for example, the address of a single router). The host relies on routers to update its routing table. In the process of routing packets, a router may detect that a host is not using the best route. The router sends an ICMP Redirect to this host, requesting that the host use a different gateway when it sends packets to a destination. The host sends packets to that destination using the new route if it is able to interpret ICMP Redirect directives.

- Uses the router with the highest preference level as the default gateway (ICMP Router Discovery)

  ICMP Router Discovery is useful if you have multiple gateways that connect a particular subnet to outside networks. By using the preference setting, you can select which gateway is the preferred choice.

- Informs sources that a packet has exceeded its allocated time to exist within the network (ICMP Time Exceeded)

For more information about ICMP Redirect and ICMP Router Discovery, see "ICMP Redirect" and "ICMP Router Discovery" later in this chapter.

ICMP Redirect

ICMP Redirect adds another layer of intelligence to routing. ICMP Redirect:

- Informs the sending device of the frame that there is a more efficient route to the destination.
- Routes the frame via the more efficient route.

Use the Administration Console or the Web Management software to enable ICMP Redirect.

Important Considerations

Keep the following things in mind with ICMP Redirect:

- ICMP Redirect determines if the sending interface is the same as the receiving interface.
- ICMP Redirect determines if the source device of the frame is on a direct-connect network. See Figure 57.
- You can enable or disable ICMP Redirect on a per-interface basis.
- There is a performance cost associated with this redirect activity. You have to monitor the activity to gauge its effect on the network.
- Performance can be affected if the sending device ignores the recommendations of ICMP Redirect, in which case the performance cost of ICMP Redirect is incurred while the benefits are wasted.

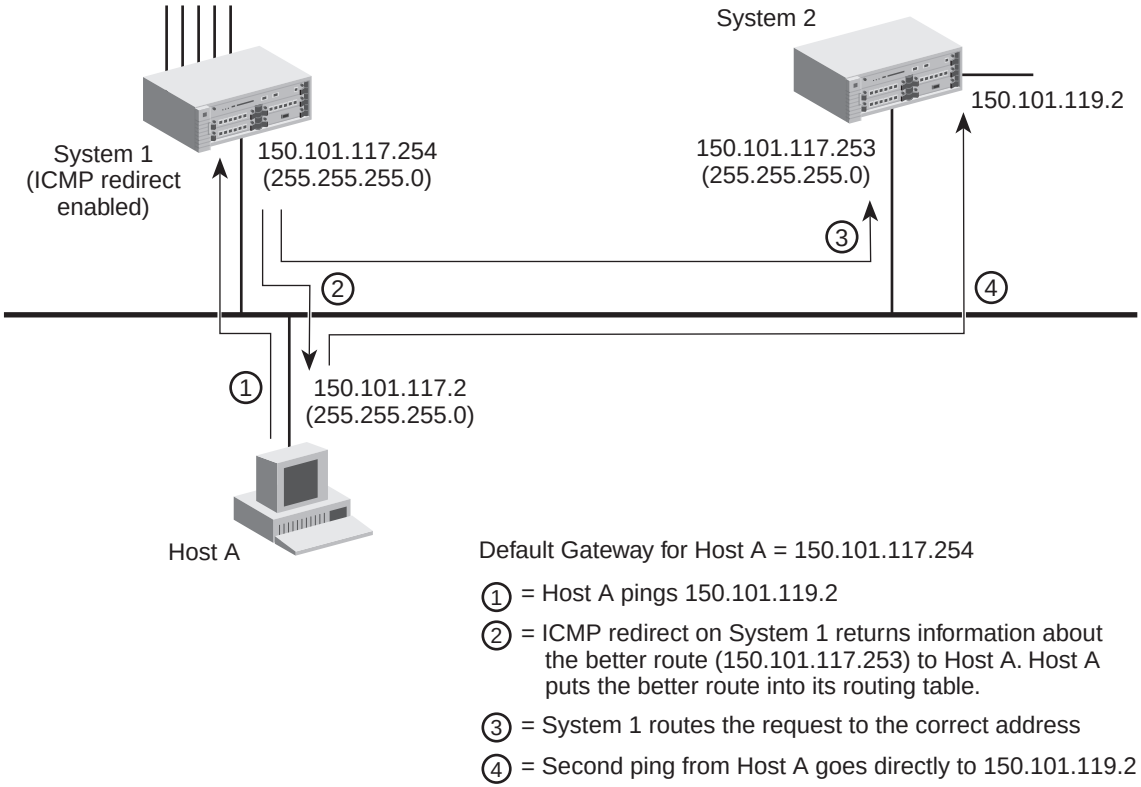
If you add or change more than one IP interface associated with the same VLAN, disable the ICMP Redirect option for optimal performance.

- If you disable ICMP Redirect, the hardware routes the frame, and no messages are sent back to the sending device. At some point, however, the number of retries associated with less intelligent hardware routing overtake any benefits that are associated with the speedier routing that hardware provides.
- To maximize the effectiveness of ICMP Redirect, have ICMP Redirect on the system that is connected to the greatest number of other routing devices.

- Disable ICMP Redirect if you have overlapped IP interfaces on ports that are not configured to use 802.1Q VLAN tagging. Doing so provides better routing performance between the overlapped subnets.
- If you have two interfaces that belong to different VLANs that share a given port and you want to completely disable ICMP redirects for that port, disable the redirects for *each* interface that shares that port. If you disable ICMP Redirect for only one interface and enable it for the other, you may not get the performance improvement that you want.

Example Figure 57 shows how ICMP Redirect works.

Figure 57 ICMP Redirect Example



ICMP Router Discovery

ICMP Router Discovery directs a host to use the router with the highest preference level as the default gateway. ICMP does this by enabling hosts that are attached to multicast or broadcast networks to discover the IP addresses of their neighboring routers and determine which router to use for a default gateway. You can choose this default gateway yourself.

Important Considerations

Keep the following points in mind with ICMP Router Discovery:

- You need not manually configure a default route.

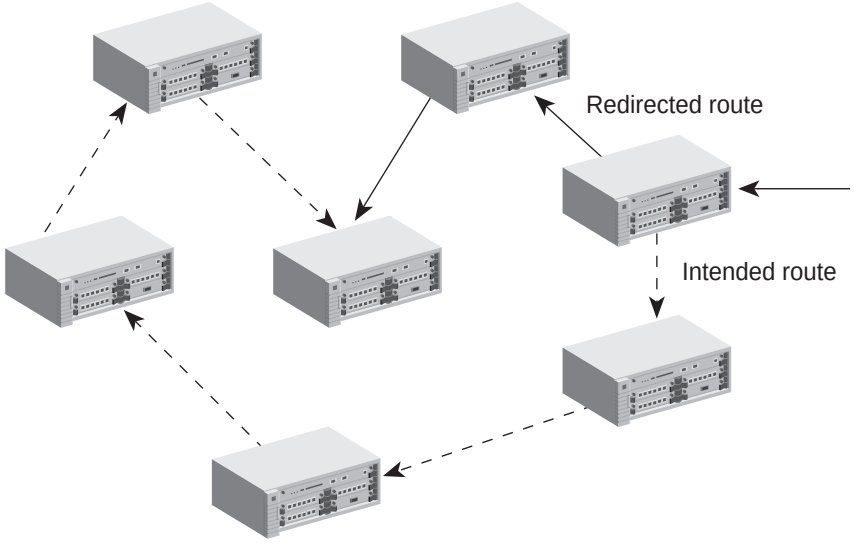
  Although IP traffic may initially be directed to any of the routers on the LAN, ICMP Redirect messages subsequently channel IP traffic to the correct router.

- ICMP Router Discovery is useful on large networks, or if the network topology has undergone a recent change.
- If you are on a small network that is relatively stable, consider using a static route to the gateway instead of ICMP Router Discovery to reduce network traffic.

Example

Figure 58 shows how ICMP dynamically determines a default gateway.

Figure 58 ICMP Router Discovery



See the documentation for your workstation to determine whether you can configure your workstation to use this protocol.

See RFC 1256 for detailed information about ICMP Router Discovery.

Broadcast Address

You can set a broadcast address for each defined IP interface. Your system uses this broadcast address when forwarding directed broadcast packets, and when advertising RIP packets.

When you define an IP interface, the broadcast address is 255.255.255.255. This is the default address.

Important Considerations

Keep the following points in mind when you use broadcast address:

- You cannot change the broadcast address for an IP interface if you have already defined any RIP advertisement addresses.
- If you are concerned with security, filter all inbound and outbound broadcast traffic. Many hosts are set up to respond to an echo request to their broadcast address with an echo reply, which can breach security.

Directed Broadcast

A directed broadcast contains 1s in the host portion of the address field. You can choose to have your system, on a per-interface basis, enable or disable the forwarding of directed broadcast frames.

Important Considerations

Keep the following points in mind when you use directed broadcast:

- When your system receives a directed broadcast and the destination is different from the interface on which it was received:
  - Your system forwards the directed broadcast if directed broadcast is enabled
  - Your system drops the directed broadcast if directed broadcast is disabled
- Set the directed broadcast to reflect your security requirements. If you have a critical IP interface, disabling directed broadcast can, for example, protect against denial-of-service attacks by malicious users.

Routing Information Protocol (RIP)

RIP is the protocol that implements routing. RIP does this by using Distance Vector Algorithms (DVAs) to calculate the route with the fewest number of hops to the destination of a route request. Each device keeps its own set of routes in its routing table. RIP is an Interior Gateway Protocol (IGP) for TCP/IP networks.

RIP operates using both active and passive devices.

- *Active* devices, usually routers, broadcast RIP messages to all devices in a network or subnet and update their internal routing tables when they receive a RIP message.
- *Passive* devices, usually hosts, listen for RIP messages and update their internal routing tables, but do not send RIP messages.

An active router sends a broadcast RIP message every 30 seconds. This message contains the IP address and a metric (distance) from the router to each destination in the routing table. In RIP, each router through which a packet must travel to reach a destination counts as one network *hop*.

Basic RIP Parameters

RIP has several parameters to consider when you set up RIP to use in your network. When you configure an IP interface, the system already has the RIP parameters set to the defaults listed in Table 40.

Table 40 RIP Parameters

RIP Parameter	Default Value
RIP-1 Mode	learn
Compatibility	disable
Cost	1
Poison Reverse	enabled
Advertisement Address	limited broadcast address (255.255.255.255)

RIP Mode

The four available settings for RIP mode are as follows:

- **Disabled** — The system ignores all incoming RIP packets and does not generate any RIP packets of its own.
- **Learn** — The system processes all incoming RIP packets, but it does not transmit RIP updates.
- **Advertise** — The system broadcasts RIP updates, but it does not process incoming RIP packets.
- **Enabled** — The systems broadcasts RIP updates and processes incoming RIP packets.

Compatibility Mode

The RIP-1 compatibility mode determines how the software sends periodic RIP-2 updates. (RIP-1 always uses the advertisement list when sending RIP-1 advertisements.)

- When the system is configured to advertise RIP-2 packets and compatibility mode is `disabled`, the software uses the multicast address of 224.0.0.9 when sending periodic updates. Doing so reduces the load on hosts that are not configured to listen to RIP-2 messages.
- When the system is configured to advertise RIP-2 packets and compatibility mode is `enabled`, the software uses the advertisement list for RIP-2 updates.

Cost

You can use RIP to calculate the route metrics (the *cost*) for you. The cost is the number of hops that the packet needs to get to its destination. The RIP cost is a number between 1 and 15. (A number higher than 15 is not allowed, because RIP cannot negotiate more than 15 hops.)

Most facilities assign a cost of 1 to all interfaces. However, if you have two links with differing speeds, such as a dial-up link versus a direct link, you may want to raise the cost of the dial-up link so that the direct link is more likely to be used.

Poison Reverse

Poison Reverse is a RIP feature that you use specifically with a scheme called *Split Horizon*. Your system enables Poison Reverse by default.

Split Horizon avoids the problems that reverse-route updates can cause. Reverse-route updates are sent to a neighboring router and include the routes that are learned from that router. Split Horizon omits the routes that are learned from one neighbor in the updates that are sent to that neighbor (the reverse routes).

Poison Reverse is essentially another layer of protection against advertising reverse routes.

- When you enable (default mode) Poison Reverse, the system advertises reverse routes in updates, but it sets the metrics to 16 (infinity). Setting the metric to infinity breaks the loop immediately when two routers have routes that point to each other.
- When you disable Poison Reverse, such reverse routes are not advertised.

You can disable Poison Reverse because it augments what Split Horizon already does, and it puts additional information that you may not need into RIP updates.

Advertisement Address

The system uses the advertisement address specified to advertise routes to other stations on the same network. The system uses this address for sending updates. (Note that RIP-2 updates depend on the setting of RIP compatibility mode.)

Each interface that you define initially uses the default broadcast address (255.255.255.255) as the advertisement address. If you change the broadcast address, the address that you specify becomes the new RIP advertisement address. If you subsequently use RIP-2 (configure the interface to send RIP-2 advertisements) and have the RIP-1 compatibility mode disabled, the multicast address is used for updates.

Effects and Consequences

- After you add an advertisement address, you cannot subsequently change the broadcast address.
- If you are using RIP-2 for the interface and if you want the system to use the advertisement list instead of the multicast address for RIP updates, enable RIP compatibility mode.

Route Aggregation

Route aggregation mode determines which route table entries are sent during a RIP-2 update:

- If route aggregation mode is enabled, RIP-2 can function like RIP-1 and "collapse" route table entries for all subnets of a directly connected network.

  For example, if route aggregation is enabled, and the system is advertising subnets 150.100.31.0 and 150.100.32.0, only the entry for network 150.100.0.0 is sent in the update. With RIP Version 2, you *must* enable route aggregation mode if you want the interface to collapse the route table entries and function like RIP-1.

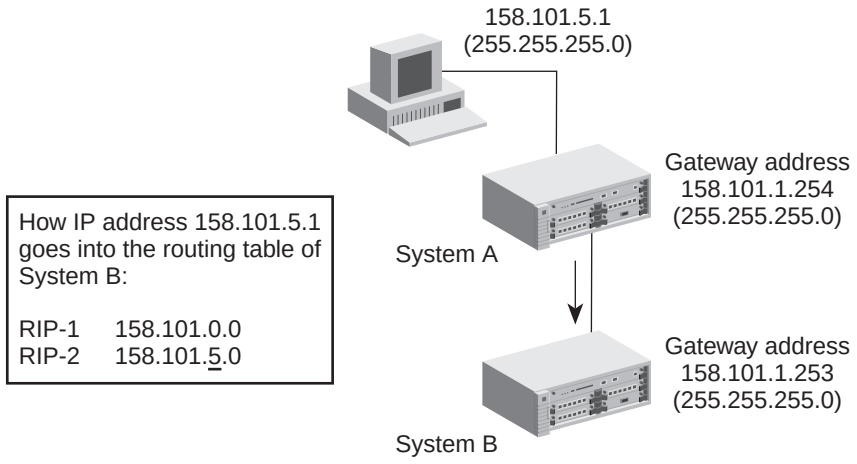
- If route aggregation mode is `disabled` (the default), a RIP-2 update sends all routing table entries.

RIP-1 Versus RIP-2

Like RIP-1, RIP-2 allows the system to dynamically configure its own routing table. RIP-2 is much more flexible and efficient than RIP-1, however, because RIP-2 advertises using the multicast method, which can advertise to a subset of the network (RIP-1 uses the broadcast method, which advertises to the whole network). RIP-2 can do this because it includes a subnet mask in its header. (See Figure 59.)

If your system receives a RIP-2 packet, your system puts the route into the routing table with the subnet mask that was advertised.

Figure 59 RIP-1 Versus RIP-2



Important Considerations

Consider the following issues when you implement RIP on your system:

- Use RIP-2 rather than RIP-1 if possible, because RIP-2 uses subnet masking and the next hop field. Subnet mask advertising allows you to use VLSM. (See "Variable Length Subnet Masks (VLSMs)" earlier in this chapter for more information.)
- Set RIP as follows:
  - **RIP-1** — `learn`
  - **RIP-2** — `enabled`

  In this way, the system keeps track of the RIP-1 and RIP-2 address routes in its routing table and forwards the routes as well.
- 3Com recommends that you not advertise RIP-1 and RIP-2 together. If you do, two different sets of IP addresses may go into to the routing table for every one RIP advertisement, which quickly reduces the efficiency of the routing table.

Routing Policies

IP routing policies allow you to control how routes are sent from and received by the routing table in your system. Both RIP and OSPF have routing policy capabilities. This section describes the RIP routing policies; OSPF routing policies are discussed in Chapter 14.

There are two basic types of routing policies:

- **Import policies** — Import policies control what routes are added to the routing table. (That is, the import policies control which routes your system can accept from other routers.) When RIP or OSPF forwards a route to the routing table, the router searches its import policies before adding the route to the routing table.
- **Export policies** — Export policies control what routes from the routing table are advertised by the RIP and OSPF protocols to other routers. (That is, export policies control which routes your system can forward to other routers.) When RIP or OSPF are preparing a route advertisement, the router searches its export policies before advertising the route to the network.

You can create up to 128 routing policies. The total is shared between OSPF and RIP policies.

Routing policies can control the entire flow of routing information among the network, the protocols, and the routing table manager.

Routing Policies are often referred to as Route Filters because defining policies for accepting and forwarding routes is very much like defining filters to screen which routes may be forwarded or accepted.

How Routing Policies Work

Each router keeps a table of current routing information, called the *routing table*. The router protocols on the system receive routes from or advertise routes to the network.

When a route needs to be added to the routing table:

1. The protocol (OSPF or RIP) that receives the route sends that route to the routing table manager.
2. The routing table manager searches the Import policies.
3. If the import policy allows the route to be accepted, the routing table manager adds the route to the routing table; otherwise, the route is discarded. See Figure 60.

The router also needs to periodically advertise routes to other routers:

1. The protocol (OSPF or RIP) polls the routing table manager for routes to advertise to other routers.
2. The routing table manager searches the Export policies.
3. If the export policy allows the route to be advertised, the routing table manager advertises the route on the network; otherwise, the route is not sent. See Figure 60.

Figure 60 IP Routing Policies

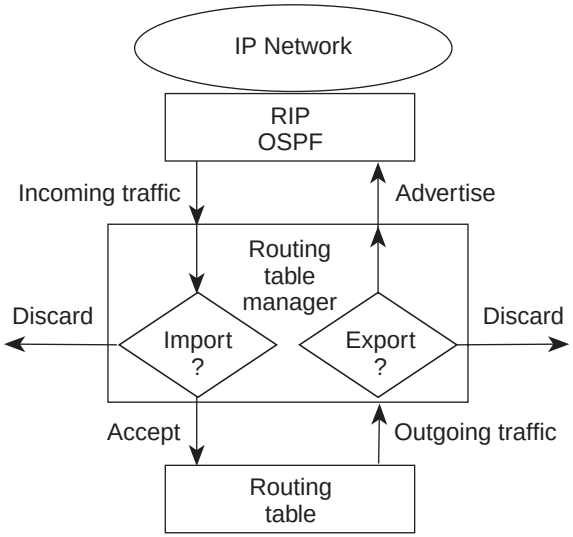


Figure 60 shows the first level of decision-making in routing policies. Routing policies also contain two parameters that help further refine this system: metrics and administrative weight.

- **Metric (cost) adjustment** — Specifies how many hops to assign to the route. The range of the metric is 0 through 16 hops. (If you specify 0, the system does not modify the metric; if you specify 16, you are specifying that the route is unreachable — 16 represents infinity.)
- **Administrative weight** — Controls the relative weight of each policy with respect to another policy. The range extends from 1 to 16, with 16 taking the greatest precedence.

Important Considerations

Even though Routing Policies are not true routing protocols and are considered optional, they can increase network efficiency.

- You can increase speed *and* security simply by limiting the number of devices from which the router receives data.
- You can establish a neighbor list of devices, which is a list of trusted systems whose addresses you have confidence in.
- You can associate the list of devices with a specific traffic direction, either incoming or outgoing. As a result, you can assign very precise routes of traffic, thereby keeping tight control over them.
- By adjusting the relative importance of certain policies over others, you can exercise great control over the type and amount of traffic to and from your system.
- If an incoming RIP route has a metric set to 16, which indicates that the route is invalid, an existing IP RIP Import Accept Policy does not change the metric.

  This ensures that an IP RIP Import Accept Policy does not overwrite RIP poison reverse triggered updates, which could cause incorrect route information to be placed into the routing tables.

Implementing RIP Routing Policies

RIP routing policies determine which RIP routes can be accepted into the routing table, and which RIP and OSPF routes can be advertised.

RIP Metric Adjustments

Use the arithmetic operators in Table 41 to adjust the RIP metrics.

Table 41 RIP Metric Adjustments

Metric	Description
+nn	Increase metric by nn
-nn	Decrease metric by nn
*nn	Multiple metric by nn
/nn	Divide metric by nn
%nn	Modulus — returns the remainder of the metric

CAUTION: *Use caution if you use arithmetic operators to adjust the relative value of the number of hops that you allow a route to have; you can inadvertently make a route unreachable.*

RIP Import Policy Conditions for Specified Interfaces

Table 42 lists the policy conditions for RIP import policies.

Table 42 RIP Import Policy Conditions

Source Router	Route (address/mask)	Action	Description
Specified router	Specified route/mask	accept	Accept specified route from specified source router on specified interfaces with or without metric adjustments (+, -, *, /, %).
Specified router	all (0.0.0.0)	accept	Accept all routes from specified router on specified interfaces with or without metric adjustments (+, -, *, /, %).
all (all routers)	Specified route/mask	accept	Accept specified route on specified interfaces with or without metric adjustments (+, -, *, /, %).
all	all	accept	Accept all routes on specified interfaces with or without metric adjustments (+, -, *, /, %).
Specified router	Specified route/mask	reject	Reject specified route from specified router on specified interfaces. (Metrics do not apply because the route itself is rejected.)
Specified router	all	reject	Reject all routes from specified router on specified interfaces.
all	Specified route/mask	reject	Reject specified route from all routers on specified interfaces.
all	all	reject	Reject all routes on specified interfaces.

RIP Export Policy Conditions for Specified Interfaces

Table 43 lists the policy conditions for the RIP export policies.

Table 43 RIP Export Policy Conditions

Protocol	Source Router	Route	Action	Description
RIP, OSPF, static	Specified router or all routers	Specified route/mask	accept	Advertise RIP/OSPF/static specified route from specified source router on specified interfaces with or without metric adjustments (+, -, *, /, %).
RIP, OSPF, static	Specified router or all routers	all (0.0.0.0)	accept	Advertise all RIP/OSPF/static routes from specified router on specified interfaces with or without metric adjustments (+, -, *, /, %).
RIP, OSPF, static	Specified router or all routers	Specified route/mask	reject	Do not advertise the RIP/OSPF/static specified route on specified interfaces.
RIP, OSPF, static	Specified routers or all routers	all	reject	Do not advertise all RIP/OSPF/static routes on specified interfaces.

Multiple Matched Routing Policies

Because you can use a wildcard parameter (`all`) to specify a source or target route, there are times when several policies can apply to the same route.

When the system perceives that there is more than one policy for the same route, it follows this hierarchy of rules to resolve the policy conflict:

- The policy with the highest administrative weight
- The policy that matches the specific source
- The policy that matches the most number of bits for the route
- The policy that matches the origin protocol
- The policy with the lowest index

Setting Up RIP Routing Policies

To configure a routing policy, follow these general steps:

1. Establish an Export policy that controls the advertisement of routes through RIP, regardless of the source from which the route is learned.
2. Establish an Import policy that accepts or refuses to accept information on routes learned by RIP from a trusted neighbor.
3. To control the reporting of routes that are learned from specific sources, establish the following policies:
   - Export policy for routes learned from OSPF
   - Static policy for reporting static (user-configured) routes

   If you decide to have routes reported with a metric that is calculated from the routing table, you can manipulate the conversion formula that RIP uses to convert a routing table metric into one that RIP understands.
4. Establish a policy to report OSPF routes so that the metrics that are reported with these routes are imported into RIP without being changed.

Effects and Consequences

Consider these points when you use routing policies:

- Configure the administrative weight setting carefully because this setting has the highest priority in resolving policy conflicts.
- If you use routing policies, do not implement static routes. Routing policies work with routes that are updated dynamically.
- Use routing policies only if you need the security, or if you need more control over the routing tables than other IP features, such as VLSMs, give you.
- To control whether a route is accepted or forwarded without making specific changes to your network configuration, consider setting the Cost metric as high as possible, and the administrative weight as low as possible.

Creating RIP Routing Policies

To set a routing policy, you need to know the following parameters:

- **Policy type** — The determination whether to accept a route into the routing table (import) or advertise a route from the routing table (export)
- **Source address** — The routing device that is sending the route to your system
- **Route address** — The actual device IP address of the route origin
- **Route subnet mask** — The subnet mask of the device IP address of the route origin
- **IP interface** — The IP interface on your system that the route is coming in on

The policy takes effect on the selected interface only if the origin protocol matches the protocol enabled for the interface selected.

- **Policy action** — The determination whether to accept or reject the route
- **Metric adjustment** — The determination to increase or decrease the route metric (the number of hops) for the route
- **Administrative weight** — The level of importance of this policy: 1 is low priority, 16 is high priority

Sketch a topology of your routers and the proposed routing policies of each to get an understanding of how the routers work together and how traffic flows.

Example Figure 61 and Table 44 show an example of how to set a RIP import routing policy.

Figure 61 RIP Routing Policies Example

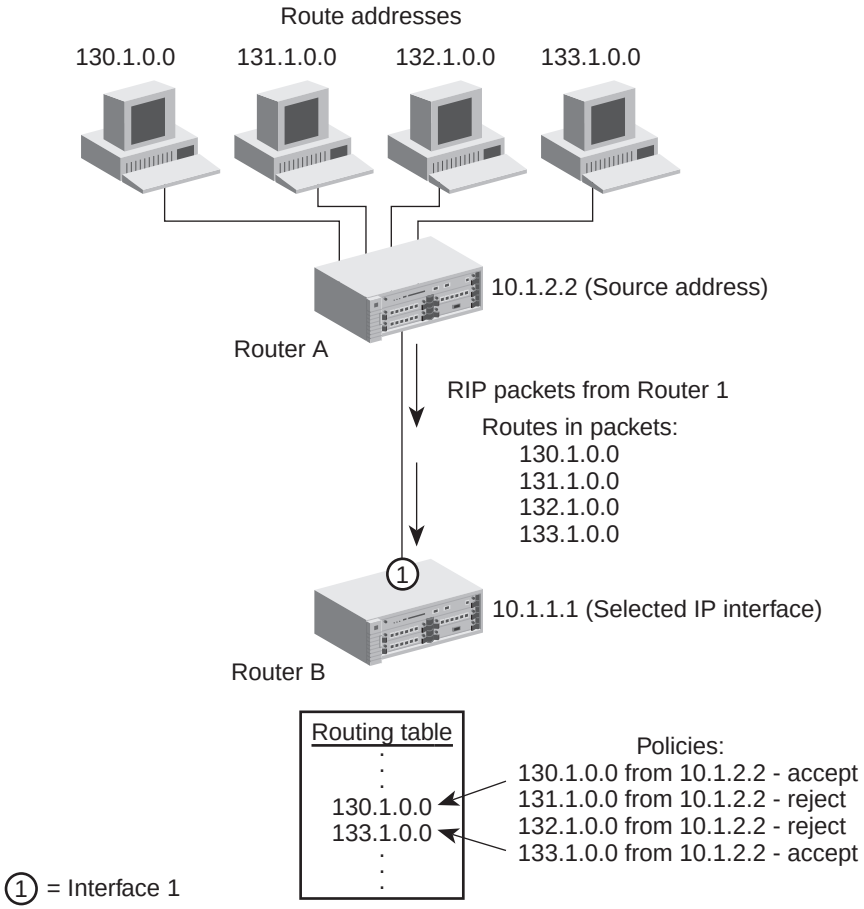


Table 44 lists the import policies for Router B from Figure 61.

Table 44 Router B Routing Policies

Policy Type	Source Address	Route Address	Route Subnet Mask	IP Interface	Policy Action	Metric	Weight
Import	10.1.2.2	130.1.0.0	255.255.0.0	1	accept	1	1
Import	10.1.2.2	131.1.0.0	255.255.0.0	1	reject	–	2
Import	10.1.2.2	132.1.0.0	255.255.0.0	1	reject	–	1
Import	10.1.2.2	133.1.0.0	255.255.0.0	1	accept	1	2

In this example, only routes 130.1.0.0 and 133.1.0.0 are accepted into the routing table of Router B.

Domain Name System (DNS)

The Domain Name System (DNS) client allows you to specify a hostname rather than an IP address when you perform various operations (for example, when you use `ping` or `traceRoute` to contact an IP station).

With DNS, you can specify one or more name servers that are associated with a domain name. Each name server maintains a list of IP addresses and their associated host names. When you use `ping` or `traceRoute` with a hostname, the DNS client attempts to locate the name on the name servers that you specify. When the DNS client locates the name, it resolves it to the associated IP address.

You can resolve an IP address to a host name or a host name to an IP address on a name server. Enter either the host name or the IP address; the DNS client displays the pair.

Important Considerations

When you set up DNS servers on your LAN, remember the following:

- Always set up more than one DNS name server (a primary and secondary server) so that the lookup service does not have a single point of failure.
- If your ISP changes the Classes of Internetwork Service, change the DNS settings on each host that the ISP services.

See UNIX NFS documentation for information about how to create and maintain lists of domain names and IP addresses on the name servers.

For information about how to use `ping` *and* `traceRoute`*, see Chapter 18 in this guide and the IP chapter in the* Command Reference Guide.

User Datagram Protocol (UDP) Helper

User Datagram Protocol (UDP) Helper allows IP applications to route broadcast packets from one subnet to another part of the network.

Two common uses of the UDP Helper feature are:

- Bootstrap Protocol (BOOTP)

  BOOTP allows diskless workstations to obtain their own IP addresses, the address of a server host, and the name of the boot file, which is loaded into memory from the server and executed. RFC 951 is the official specification for BOOTP with clarifications and extensions provided in RFC 1542.

- Dynamic Host Configuration Protocol (DHCP)

  DHCP provides a framework for passing configuration information to hosts on an IP network. DHCP is based on BOOTP, but adds the capability of automatic allocation of reusable network addresses and other configuration options. DHCP captures the behavior of BOOTP relay agents and DHCP participants can interoperate with BOOTP participants. RFC 2131 is the official specification for DHCP.

Both BOOTP and DHCP use the logical port number 67 for their servers. However, 3Com implements a generic UDP Helper agent in the system that can apply to any port.

Implementing UDP Helper

Configure UDP Helper by specifying a logical port number on which to listen for broadcast IP packets and an IP forwarding address for the packet.

The system handles the packet based on the way VLANs are set up:

- **If the broadcast packet is received on the *same* VLAN as the forwarding address** — Your system bridges the packet.
- **If the broadcast packet is received on an IP interface that is associated with a *different* VLAN than the forwarding address** — Your system updates the BOOTP/DHCP gateway IP address of the packet to use the interface on which the packet was received; the system then routes the frame to the forwarding IP address.

You have to set the following UDP Helper parameters:

- **UDP port number** — A logical address, not a port (interface) on your system. BOOTP (including DHCP) uses UDP port 67.
- **IP forwarding address** — The IP address to which the packets are forwarded. You can have up to 63 combinations of port numbers and IP forwarding addresses per router. You can also have multiple IP address entries for the same ports.
- **Hop count** — The number of interfaces the system uses to forward a packet through the router.
- **Threshold** — The maximum number of times that the system forwards a packet to the network. By default, there is no BOOTP relay threshold. (The default value is `0`.)

The commands to implement these parameters are described in the chapter "IP Routing" in the *Command Reference Guide*.

You need to have a thorough understanding of your network configuration to use UDP Helper. Review the network topology before you implement UDP Helper.

Configuring Overlapped Interfaces

Overlapped IP interfaces are multiple logical interfaces that are defined for a single physical port. You can specify how UDP Helper forwards packets from overlapped IP interfaces with one of these interface options:

- **First** — The system uses the first overlapped IP interface of the port as the source network for forwarded packets.
- **Even** — The system hashes the MAC address of the client to determine the source network for forwarded packets. This arrangement evenly distributes the interface among those on the network.
- **Sequential** — The system assigns each overlapped IP interface, in turn, as the source network for forwarded packets.

You can view the UDP Helper configuration when you configure the forwarding address.

Important Considerations

Consider the following points when you use UDP Helper:

- The maximum BOOTP hop count (how many steps the system uses to forward a packet through the router) is 16; the default hop count limit is 4. Keep the hop count as low as possible for performance purposes.
- 3Com recommends that you keep the UDP port number at 67. The Port number 67, which is the industry standard, helps ensure that UDP packets do not get dropped due to an unknown destination failure.
- You can always add or remove a port number or IP forwarding address defined for UDP Helper.
- If possible, the system always bridges the UDP broadcast packet on the same VLAN from which the packet was received. Therefore, if the configuration has overlapped VLAN interfaces on a particular port, and one of those interfaces is associated with the same VLAN as an IP forwarding address, then the frame is broadcast to that VLAN, regardless of the interface option that you have selected.

Standards, Protocols, and Related Reading

This section describes how to obtain more technical information about IP.

Requests For Comments (RFCs)

Documents called Requests for Comments (RFCs) contain information about the entire set of protocols that make up IP. Some of the RFCs that pertain to the discussions in this chapter are:

- **RFC 791** — Internet Protocol
- **RFC 951, 1542** — UDP Helper
- **RFC 1219** — Subnet Numbers
- **RFC 1878** — VLSMs
- **RFC 1256** — ICMP Router Discovery Messages
- **RFC 1058** — RIP
- **RFC 1723** — RIP Version 2
- **RFC 1786** — IP Routing Policies
- **RFC 2131** — DHCP
- **RFC 2400** — Internet Official Protocol Standards

You can obtain copies of RFCs from the Web site of the Internet Engineering Task Force (IETF):

```
http://www.ietf.org
```

Standards Organizations

Standards organizations ensure interoperability, create reports, and recommend solutions for communications technology. The most important standards groups are:

- International Telecommunications Union (ITU)
- Electronic Industry Association (EIA)
- American National Standards Institute (ANSI)
- International Standards Organization (ISO)
- Institute of Electrical and Electronic Engineers (IEEE)
- Internet Engineering Task Force (IETF)
- National Institute of Standards and Technology (NIST)

Related Reading

For more information about the IP protocol suite, refer to the following books:

- *High Speed Networks: TCP/IP and ATM Design Principles* by William Stallings, Prentice Hall, 1998
- *Local Area Networks: Architectures and Implementations* by James Martin, Prentice Hall, 1994
- *Internetworking with TCP/IP: Principles, Protocols, and Architecture* by Douglas Comer, Prentice Hall, 1995

12 VIRTUAL ROUTER REDUNDANCY PROTOCOL (VRRP)

The Virtual Routing Redundancy Protocol (VRRP) can prevent a loss of network operations for end hosts due to the failure of the static default IP gateway. VRRP accomplishes this by allowing you to designate a number of other routers as Backup routers in the event that the Master router (the default router) should fail for any reason.

Topics covered in this chapter include:

- VRRP Overview
- Key Concepts
- Important Considerations
- Implementing VRRP
- VRRP and Other Networking Operations
- Standards, Protocols, and Related Reading

Before you implement VRRP, be sure that you have a good understanding of how IP networks function. See Chapter 3 for more information about IP networks. Also, be sure to read this chapter thoroughly before you set up VRRP on your network.

You can configure and manage VRRP in either of these ways:

- *From the* `ip vrrp` *menu of the Administration Console. See the* Command Reference Guide.
- *From the IP VRRP folder of the Web Management software. See the* Web Management User Guide.

VRRP Overview

A critical component of IP networking is the way in which hosts and routing devices find the next-hop address in a connectionless environment. There are several different ways of determining the next-hop address, but they all fall into two basic categories:

- Router to Router
- Host to Host and Host to Gateway

Router to Router

Router-to-router communication is usually accomplished by means of a routing protocol such as Routing Information Protocol (RIP) or Open Shortest Path First (OSPF), or by static routes. Routers consult their own routing tables to make intelligent next hop decisions for the forwarding of IP packets.

Host to Host and Host to Gateway

IP host-to-host communication typically begins with an ARP request to the destination host address, providing that the destination resides on the same subnet as the sending device. If the destination address resides on a non-local subnet, then the sending device must use one of the following methods to learn the route to the remote network:

- Routing protocol
- ICMP Router Discovery
- Static route
- Default gateway

Routing Protocols

Routing protocols provide dynamic updates to end stations in the event of a network failure, but they are typically not used on most hosts because they require additional setup, processing power and, in some cases, additional software.

ICMP Router Discovery

ICMP Router Discovery directs a host to use the router with the highest preference level as the default gateway. Internet Control Message Protocol (ICMP) does this by enabling hosts that are attached to multicast or broadcast networks to discover the IP addresses of their neighboring routers and determine which router to use for a default gateway. If you prefer, you can make this default gateway choice yourself.

Static Route

A static route is an IP address that is user-configured and fixed. Static routes are useful if the host only needs to access a few networks; in this case, static routes actually require less overhead than dynamic routing protocols. However, in today's networking environment in which traffic patterns are less predictable, many routes are usually required, and static routes then become prohibitive to maintain.

Default Gateway

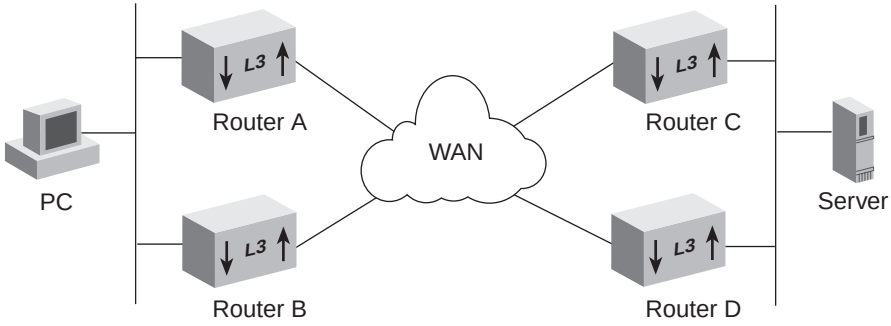
Most host stations today use a default gateway to facilitate routing. You simply define for the host an IP address on the local subnet of a router that is responsible for routing packets to their destinations. This approach is widely deployed today; however, it has one major drawback: if the default gateway becomes unavailable, then all routing to remote networks stops, requiring manual intervention to restore connectivity even if there are alternate paths available.

VRRP addresses this drawback by defining an election protocol that dynamically assigns responsibility for a *virtual router* to one of the VRRP routers on a LAN. The election process automatically detects a failure of the primary (Master) router, and transfers all traffic forwarding to the backup router. All of this is done without your intervention, which dramatically increases uptime in a Layer 3 IP network.

Example In the simplest scenario, a VRRP configuration includes two routers, a primary router (called the Master router) and a backup router. If the Master router fails for any reason, the backup router assumes all forwarding functions for the Master router. The backup router monitors the network for hello packets, which are periodically sent by the Master router (the default time period is 1 second). If the backup router misses three hello packets in succession, that router assumes forwarding functions for the Master.

See Figure 62 for a visual representation of a simple virtual router configuration.

Figure 62 Simple VRRP Configuration



In the example shown in Figure 62, Router A is the default gateway for the workstation named PC, which provides access to the Wide Area Network (WAN) and to the device named Server. Assume that no router discovery protocols have been configured and that the default gateway is static.

If the workstation loses its connection to Router A, the workstation loses all remote connectivity because its default gateway is no longer available. However, if VRRP is enabled in this same scenario, Router B detects the loss of connectivity to Router A, and Router B assumes all forwarding responsibilities on behalf of Router A. This transfer of forwarding responsibilities allows the workstation to have continued access across the WAN to the server.

Key Concepts

This section contains some VRRP definitions that you should know before reading further.

- **VRRP router** — A router running the VRRP protocol. A VRRP router can:
  - Act as a Master router with actual addresses on a interface
  - Act simultaneously as a Backup for other routers with additional virtual router mappings and priorities for those routers
- **Virtual router** — A logical entity, managed by VRRP, that acts as the default router for hosts on a shared LAN. The virtual router has a unique identifier called the Virtual Router Identifier (VRID), and has a set of associated IP addresses across the LAN.

- **Virtual router master** — The VRRP router that forwards packets sent to the IP addresses associated with the virtual router. Also called the Master router. A virtual router is the Master when:
  - You configure it (using the Administration console, the Web Management console, or SNMP) as the primary IP address for a given interface
  - Backing up a Master that has been disconnected or disabled
- **Virtual router backup** — The VRRP router available to assume packet forwarding responsibility for a virtual router should the Master fail. Also called the Backup router.
- **IP address owner** — The router that is the original owner of the IP addresses that virtual router incorporates into its own IP address set. The IP address owner must be a VRRP participant.
- **Primary IP address** — An IP address selected from the set of available interface addresses. This is the IP address that VRRP uses in its advertisements that supply the source of the IP packet.
- **Virtual router initialize** — A state in which a virtual router is defined but not enabled. A virtual router is also in the initialize state when its associated interface is not operational.

How VRRP Works

When you assign Master router responsibilities to one of the virtual routers on the LAN, the Master controls the IP addresses associated with a virtual router. The Master router forwards the IP packets sent to the IP addresses it controls.

The backup process works as follows. The Master router sends out periodic VRRP advertisement messages, at time intervals you set, to the other VRRP routers and to the hosts on the common LAN. (A VRRP advertisement consists of the IP addresses that the Master owns and the virtual MAC address of the virtual router.) If the Master stops forwarding advertisements to the other routers for a predetermined period of time, the other routers automatically enter an election process to determine which router takes over Master responsibilities.

After the original Master again become operational, it begins again to broadcast advertisements to the other virtual routers if preempt mode is enabled. Packet forwarding responsibility then shifts back to the original Master router.

For this scheme to work, the association between VRIDs and IP addresses must be coordinated among all VRRP routers across the LAN: otherwise, the backup router does not have a valid set of IP addresses to use.

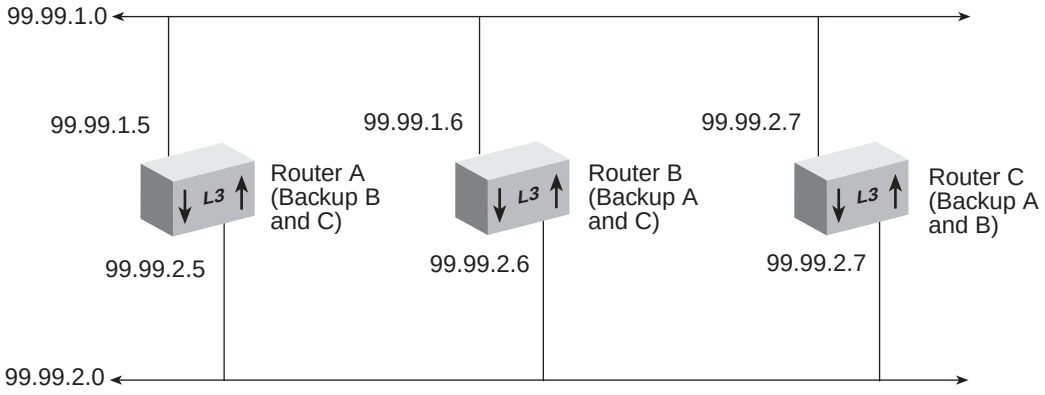
Virtual Router Decision-making

The example in Figure 62 shows only two routers, so there is no ambiguity as far as which router should have assumed responsibility upon a failure. However, there can be more than one virtual router on a network because there can be more than one backup router for each static gateway. This is because a single backup router, at the time of assuming primary router responsibilities, becomes the single point of failure.

See Figure 63 for an example of a network topology that:

- Allows all routers on the LAN to be backed up by more than one virtual router.
- Allows hosts on any subnetwork to reach destinations on any other subnetwork in the extended network.

Figure 63 Multiple Virtual Routers Backing Up Each Other



The parallel design in Figure 63 takes advantage of the capabilities of VRRP. This design can be extended to include more routers and more subnetworks. In a more complex virtual router scheme with many backup routers, this method ensure that all routers have adequate backup in the event of a failure.

VRRP provides for this by making you assign each virtual router on the LAN a priority value between 1 and 255. (255 means that the virtual router is the actual owner of the IP addresses.) If the Master fails, the virtual router with the next-highest priority takes over Master responsibilities until the original Master comes back online.

If two routers have the same priority, VRRP resolves the conflict by selecting the virtual router with the numerically-highest primary IP address. In other words, if Virtual Router A (primary IP address of 1.1.1.2) and Virtual Router C (primary IP address of 1.1.1.3) both have a priority of 100, Virtual Router C would have a higher priority than Virtual Router A.

CAUTION: *Configure all of the routers participating in the VRRP scheme on your network to have the same representation of the network. If some routers have a different view of the topology than others, a backup router failure is more likely, with the resultant loss of some or all end hosts' connection to the network.*

Important Considerations

This section provides information to be aware of when you implement VRRP:

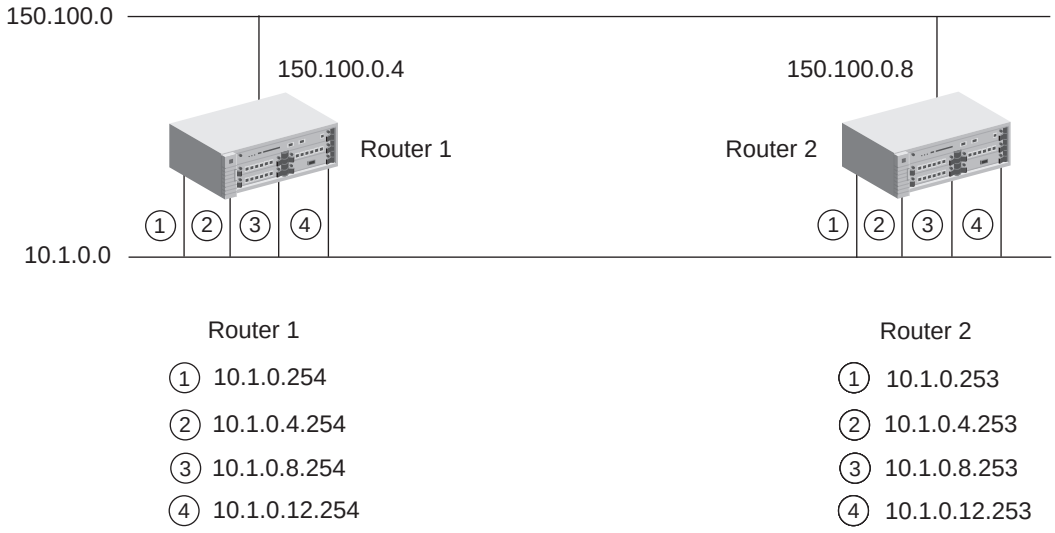
- The Master router forwards the IP addresses that you have associated with the primary virtual router, and:
  - Responds to ARP requests for the IP address or addresses that are associated with the virtual router.
  - Forwards packets that have a destination Link Layer MAC address that matches the virtual router MAC address. In other words, the Master forwards packets that hosts have sent to the virtual router to be routed.
  - Discards packets addressed to the IP address or addresses associated with the virtual router if the virtual router is not the IP address owner. Otherwise, ping, SNMP, and Telnet do not function properly.
  - Sends periodic VRRP advertisement messages. (Set the advertising interval to be short enough to provide a timely transition to another router should the Master fail. Try an advertising interval of 1 second.)
- A Backup router monitors the availability and state of the Master, and:
  - Does not respond to ARP requests for the IP address or addresses associated with the virtual router.
  - Discards packets that have a destination Link Layer MAC address that matches the virtual router MAC address.
  - Does not accept packets addressed to the IP address or addresses associated with the virtual router
- Hosts obtain the virtual router's MAC address by means of an ARP broadcast.
- VRRP is *not* a routing protocol, and it is only as useful as the design of the network upon which it is implemented. Good network design is critical in ensuring the success of router redundancy.
- The virtual routers must be on the same VLAN.

- VRRP supports Proxy ARP; the virtual router uses the virtual router MAC address in Proxy ARP replies.
- VRRP supports Fiber Distributed Data Interface (FDDI) and Ethernet
- Consider using VRRP in conjunction with port-based routing to provide router redundancy on your campus backbone. See Chapter 11 for an example of port-based routing on a campus backbone.

Implementing VRRP

This section contains a sample VRRP configuration. Specifically, it shows you how to configure interfaces on Router 2 and Router 1 (in the topology shown in Figure 64) to back up each other.

Figure 64 Sample Configuration



To implement this configuration, you must perform the following tasks:

- Create VLANs
- Configure IP Interfaces
- Configure the Router Protocol
- Enable Routing
- Configure VRRP
- Enable VRRP

Create VLANs

This section identifies the VLAN parameters that you must configure for both Router 1 and Router 2. Use `bridge vlan define` in the Administration console or use the Bridge VLAN Define form in the Web Management console to configure the VLANs.

If you configure closed VLANs, the system removes all existing VLANs and recreates the Default VLAN.

VLANs for Routers 1 and 2

VLAN	VID	Port	Protocol Suite	Tagging
VLAN1	2	1	IP	None
VLAN2	3	2	IP	None
VLAN3	4	3	IP	None
VLAN4	5	4	IP	None
VLAN5	6	5	IP	None

VID1, with a protocol of `unspecified`, is used for the Default VLAN. For more information about VLANs, refer to your Implementation Guide.

Configure IP Interfaces

This section identifies the IP interfaces that you must configure for both Router 1 and Router 2, as shown in Figure 64. Use `ip interface define` in the Administration Console or use the IP Interface Define form in the Web Management console to configure the IP interfaces.

IP interfaces for Router 1

Index	Type	IP Address	Subnet Mask	VLAN Index
1	VLAN	10.1.0.254	255.255.252.0	2
2	VLAN	10.1.4.254	255.255.252.0	3
3	VLAN	10.1.8.254	255.255.252.0	4
4	VLAN	10.1.12.254	255.255.252.0	5
5	VLAN	150.100.0.4	255.255.255.0	6

IP interfaces for Router 2

Index	Type	IP Address	Subnet Mask	VLAN Index
1	VLAN	10.1.0.253	255.255.252.0	2
2	VLAN	10.1.4.253	255.255.252.0	3
3	VLAN	10.1.8.253	255.255.252.0	4
4	VLAN	10.1.12.253	255.255.252.0	5
5	VLAN	150.100.0.8	255.255.255.0	6

Configure the Router Protocol

Configure a dynamic routing protocol (RIP-2 or OSPF) for both Router 1 and Router 2. In this case, the sample configuration uses RIP-2. Use the `ip rip` menu in the Administration Console or use the IP RIP Web Management forms to configure RIP on both routers.

Index	RIP-1 Mode	RIP-2 Mode	Compatibility Mode	Route Aggregate	Cost	Poison Reverse	Advertisement Addresses
1	learn	enabled	disabled	enabled	1	enabled	255.255.255.255
2	learn	enabled	disabled	enabled	1	enabled	255.255.255.255
3	learn	enabled	disabled	enabled	1	enabled	255.255.255.255
4	learn	enabled	disabled	enabled	1	enabled	255.255.255.255
5	learn	enabled	disabled	enabled	1	enabled	255.255.255.255

Enable Routing

You must explicitly turn routing on to enable your router to learn and advertise IP packets. Use `ip routing` in the Administration Console or use the IP Routing Web Management form to enable the routing state on both routers.

Configure VRRP

Configure the virtual router parameters on Router 1 and Router 2 so that each routing interface has a backup in case of failure. Each VRID number is associated with both a VLAN index number and the VID's IP interface established earlier in this procedure. In the case of a backup router, notice that you associate the IP address of the interface you want to back up.

VRRP parameters for Router 1

VLAN Index	IP Interface	Virtual Router Type	VRID	IP Association Address	Interval	Priority	Preempt
2	10.1.0.254	Primary	1	10.1.0.254	1	100	yes
3	10.1.4.254	Primary	2	10.1.4.254	1	100	yes
4	10.1.8.254	**Backup**	3	**10.1.8.253**	1	100	yes
5	10.1.12.254	**Backup**	4	**10.1.12.253**	1	100	yes
6	150.100.0.4	Primary	5	150.100.0.4	1	100	yes

VRRP parameters for Router 2

VLAN Index	IP Interface	Virtual Router Type	VRID	IP Association Address	Interval	Priority	Preempt
2	10.1.0.253	**Backup**	1	**10.1.0.254**	1	100	yes
3	10.1.4.253	**Backup**	2	**10.1.4.254**	1	100	yes
4	10.1.4.253	Primary	3	10.1.8.253	1	100	yes
5	10.1.4.253	Primary	4	10.1.12.253	1	100	yes
6	150.100.0.8	**Backup**	5	**150.100.0.4**	1	100	yes

Enable VRRP

You must explicitly turn virtual routing on for *each* virtual router in order to enable your virtual router to become an active component of VRRP on your network. Use `ip vrrp mode` in the Administration console or use the IP VRRP Mode form in the Web Management console to enable VRRP.

If you want to redefine a VLAN that is being used for a virtual router, you must remove the virtual router, then remove the IP interface used by the virtual router, then finally remove the VLAN.

VRRP and Other Networking Operations

Read this section for information about how VRRP interacts with other networking functions, including:

- Spanning Tree Protocol (STP)
- Dynamic routing protocols:
  - Routing Information Protocol (RIP)
  - Routing Information Protocol version 2 (RIP-2)
  - Open Shortest Path First (OSPF)
- IP Multicast
- ICMP Redirect
- Quality of Service (QoS)

Spanning Tree Protocol (STP)

Figure 63, earlier in this chapter, shows how you can set up VRRP parallel routers to provide total redundancy in your inter-LAN operations. However, because VRRP uses MAC addresses in its advertisements, this topology can represent a bridge loop to STP. In this parallel topology, VRRP advertisements must go out on the network in AllClosed mode with IgnoreSTP (Ignore Spanning Tree Mode) *enabled*.

Carefully evaluate your bridging and routing topologies before you incorporate VRRP into your network operations.

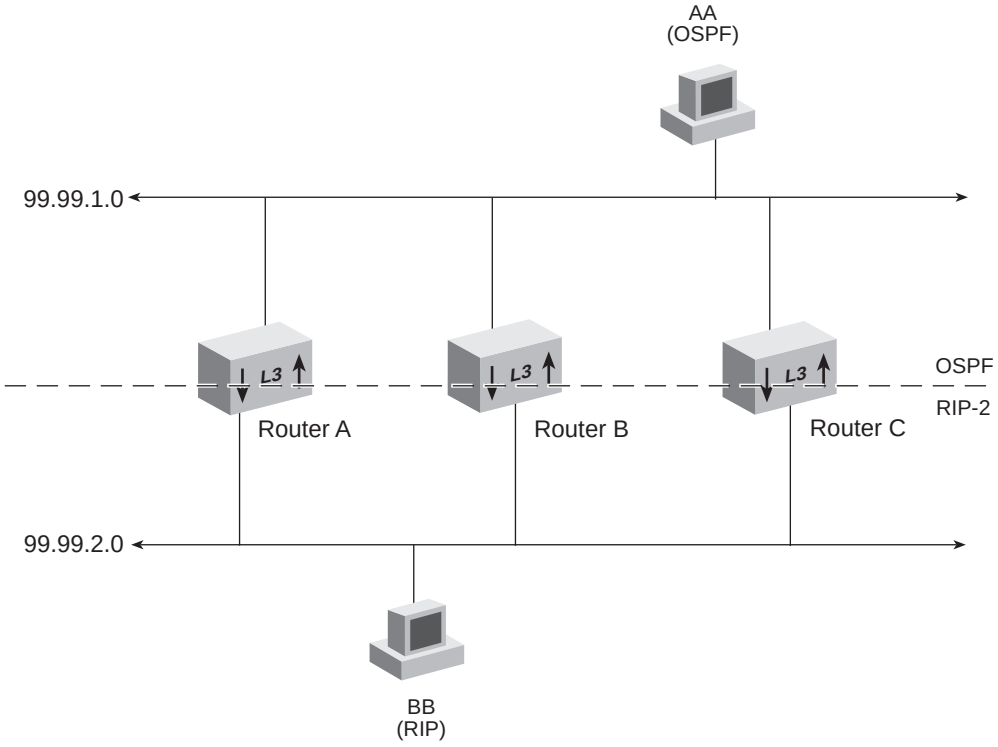
Dynamic Routing Protocols (RIP, RIP-2, OSPF)

The dynamic routing protocols RIP, RIP-2, and OSPF have their own facilities to track routes across networks. You can continue to use these protocols with VRRP routers, but on any given subnetwork, you must configure the same routing protocols with the same parameters.

Figure 65 shows how, in a parallel routing environment, OSPF is configured on each interface in the 99.99.1.0 subnetwork, and RIP-2 is configured on each interface in the 99.99.2.0 subnetwork. The device AA has a gateway of Router A.

If Router A becomes unavailable, Router B can take over because the 99.99.1.0 subnetwork has OSPF configured for each routing interface. If, however, dynamic routing protocols are configured on a router-per-router basis, so that Router B had RIP-2 configured on the router's interface to the 99.99.1.0 subnetwork, the gateway becomes unavailable because of a dynamic routing protocol mismatch.

Figure 65 Proper Use of Dynamic Routing Protocols with VRRP



IGMP Queries

IP multicast routers use IGMP to query subnetworks in order to detect host members of multicast groups. IGMP specifies a querier election process in which one router per subnetwork is designated to issue the IGMP Query messages to host members. The designated router, called the *Querier*, always has the lowest IP address in the subnetwork.

If the Querier goes down, another router can be designated to take its place. The fewer routers that you have designated as possible Queriers, the more efficient the handover is.

Be aware that, if you introduce a parallel router topology to take advantage of VRRP, you can introduce a topology that is not optimal for IGMP operations, especially as the number of routers increase. Carefully gauge the effect of VRRP on your IGMP operations.

ICMP Redirect

Using ICMP Redirect in conjunction with VRRP might cause gateway access problems due to potential conflicts between actual MAC addresses and the virtual MAC addresses that VRRP uses. Disable ICMP Redirect if you are using VRRP.

Quality of Service

You can enable Quality of Service (QoS) to run on systems running the VRRP protocol. As with the case of dynamic routing protocols, however, you must configure QoS parameters consistently across the routing interfaces on the same subnetwork.

Also, periodically examine how QoS uses the route cache because the cache may fill up faster in a VRRP environment. Consider classifying specific port ranges in this case.

IP Routing Policies

If you are using IP routing policies to control traffic on your network you must apply the same rate limits to all virtual routers on the LAN, Master as well as Backups. Failure to match routing policies among all virtual routers on the LAN could, for example, leave some routing destinations unreachable.

Dynamic Host Configuration Protocol (DHCP)

Consider using VRRP if your network uses the Dynamic Host Configuration Protocol (DHCP). DHCP provides for a default gateway and an end-host IP address, and therefore is at risk to be a single point of failure. Configure DHCP exactly the same across all the virtual routers on the LAN.

Standards, Protocols, and Related Reading

The Virtual Router Redundancy Protocol is defined in the IETF Request For Comment document (RFC) 2338. RFC2338 can be found at the following site:

`http://www.ietf.cnri.reston.va.us/rfc/rfc2338.txt`

The IANA (Internet Assigned Numbers Authority), assigns and maintains lists of all assigned numbers used for operation of the Internet (protocol type, Ethernet codes, PPP codes, IP port numbers, ICMP parameters, IP Multicast addresses, HTTP parameters, IEEE 802 numbers, and so forth).

The "Directory of General Assigned Numbers" can be found at the following site:

`http://www.iana.org/numbers.html`

13 IP MULTICAST ROUTING

This chapter provides conceptual information, configuration options, and implementation guidelines for IP multicast routing on your system.

This chapter covers the following topics:

- IP Multicast Overview
- How a Network Supports IP Multicast
- Key Concepts
- How IGMP Supports IP Multicast
- How DVMRP Supports IP Multicast
- Key Guidelines for Implementation
- Configuring IGMP Options
- Configuring DVMRP Interfaces
- Configuring DVMRP Tunnels
- Configuring DVMRP Default Routes
- Viewing the DVMRP Routing Table
- Viewing the DVMRP Cache
- Using IP Multicast Traceroute
- Standards, Protocols, and Related Reading

You can manage IP multicast routing parameters from the `ip multicast` *menu of the Administration Console. See the* Command Reference Guide.

IP Multicast Overview

The easiest way to begin to understand multicasting is to compare it against two other address types and their communication models.

Unicast Model

A *unicast* address is designed to transmit a packet from a source to a single destination. Unicast transmissions are for *one-to-one* communication. If multiple users need to receive the same communication, the source operating in unicast mode generates and sends each copy separately.

Broadcast Model

A *broadcast* address is used to send a datagram from a source to multiple destinations — an entire subnetwork, for example. Broadcast transmissions produce *one-to-many* communication, but some of the receivers may not want or need to receive the communication.

Multicast Model

A *multicast* address is used for *one-to-many* and *many-to-many* communication in an environment where users and network devices either explicitly or implicitly communicate their desire to receive the communication.

In contrast to unicast, a source that uses IP multicast generates and sends only *one* copy of the information that is desired by multiple receivers. At point where the delivery path that reaches group members diverges, network devices replicate and forward the packets. This approach makes efficient use of both source processing power and network bandwidth.

When using the Internet Protocol (IP) as the basis for multicast communication, the requests for and delivery of the communication is fundamentally controlled by referencing certain IP addresses or their MAC-based equivalents. These addresses are called group addresses or *groups* and hosts that reference these addresses are called *group members.*

IP multicast group members can be scattered across multiple subnetworks; thus, successful transmission from a source to group members can occur within a campus LAN, a MAN, or over a WAN.

As an extension to the standard IP network-level protocol, IP multicast was first defined in 1985 in RFC 966. Certain other protocols are used to support IP multicast processes. These are explained later in this chapter.

Benefits of IP Multicast

New applications that are designed to increase productivity within and across organizations are driving the need for network infrastructures to support IP multicast. When the application content is time-sensitive or requires significant bandwidth (for example, a video stream), the IP multicast process provides an efficient delivery mechanism.

The business benefits of using IP multicast are that it:

- Enables the simultaneous delivery of information to many receivers in the most efficient, logical way
- Vastly reduces the load on the source (for example, a server) because it does not have to produce multiple copies of the same data
- Makes efficient use of network bandwidth and scales as the number of participants or collaborators expands
- Works in concert with other protocols and services, such as Quality of Service (QoS) and Resource Reservation (RSVP) requests to support real-time multimedia

How a Network Supports IP Multicast

To support IP multicast, the sending and receiving nodes, as well as the network infrastructure between them, must be multicast-enabled. Specifically, there must be cohesive support for IP multicast in the following components: TCP/IP protocol stack, operating systems, application software, NICs, and Layer 3 devices. Support for IP multicast in Layer 2 devices is not required by the standard, but is desirable, as explained later in this section.

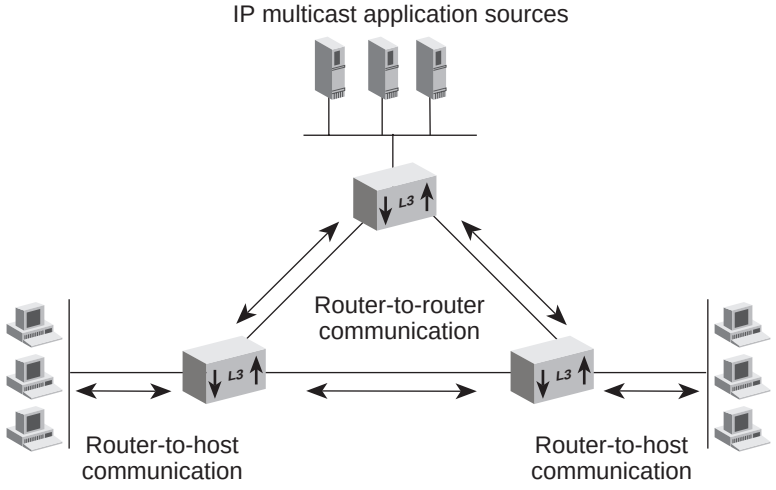
IP Multicast Routing

IP multicast transmissions fundamentally depend on multicast-enabled Layer 3 devices (traditional routers or Layer 3 switches; hereafter both are called *routers*) to direct packets on an efficient path from sources to destinations.

As shown in Figure 66, routers that support IP multicast must accomplish two important tasks:

- Communicate with other routers to determine the shortest, loopfree delivery path between an IP multicast source and its group members
- Communicate with hosts on its directly attached subnetworks to determine which hosts want to join or leave IP multicast groups

Figure 66 IP Multicast Communication Processes



Supporting Protocols in Your System

To communicate with other routers, your system supports the Distance-Vector Multicast Routing Protocol (DVMRP) version 3.6. DVMRP functions and configuration options are explained later in this chapter.

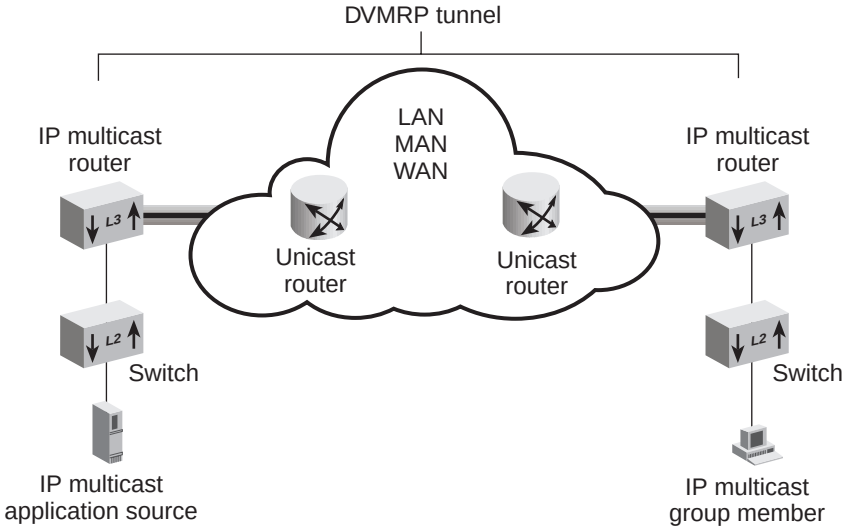
To communicate with group members on directly attached subnetworks, your system supports the Internet Group Management Protocol (IGMP) version 1 and version 2. IGMP functions are covered later in this chapter.

IP Multicast Tunnels

IP multicast routers are key connection points for delivering IP multicast traffic between sources and multicast group members. In the event that some routers in your network only transmit unicast packets, you can configure a transitional technique called *tunneling* to extend the service area. Tunnels provide a virtual point-to-point link between two multicast routers, where the path between them includes one or more routers that do not support multicast routing (unicast routers).

Figure 67 depicts a network configuration that requires an tunnel in order for the PC to receive the IP multicast application. The multicast routers support DVMRP, thus the tunnel is also configured with that protocol.

Figure 67 DVMRP Tunnel Example



A multicast router is required at each end of the tunnel. At each tunnel entrance, the router encapsulates the IP multicast packets in standard IP unicast packets — that is, it puts them in a format that the unicast routers can understand. When these packets reach the end of the tunnel, the router strips the encapsulation away and returns the packet to its native IP multicast format.

Supporting Protocol in Your System

Your system uses the Distance-Vector Multicast Routing Protocol (DVMRP) to form IP multicast tunnels. Specific aspects of tunnel configuration are described later in this chapter.

IP Multicast Filtering

When a router discovers that at least one IP multicast group member resides on a directly attached subnetwork, it forwards group traffic on that interface until it determines that group members no longer require the traffic. If multiple ports are configured in an interface, a router sends copies of the group traffic to all ports, even if only one port of those ports leads to group members. This is because the multicast routing protocol does not track *exactly* where group members reside on that interface.

The ability to filter IP multicast traffic on ports within a routing interface that do not lead to group members is highly desirable (although it is not required in the IP multicast standard) because it allows you to further optimize the LAN environment. Through targeted filtering, a router can conserve even more network bandwidth and minimize unnecessary interruptions to endstation CPUs.

It is also important to have a similar IP multicast filtering capability in Layer 2 devices. Your 3Com options include the CoreBuilder® 9400 switch, CoreBuilder 9000 Layer 2 modules, SuperStack® II Switch 3900, and SuperStack II Switch 9300.

Supporting Protocol in Your System

To track which ports in an interface require IP multicast group traffic, your system supports the Internet Group Management Protocol (IGMP) version 1 and version 2. IGMP covers two main functions: querying and snooping. These are explained later in this chapter.

Internet Support for IP Multicast

The MBONE is the Internet's experimental multicast backbone network. It is an interconnected set of Internet routers, subnetworks, and tunnels that support the delivery of IP multicast traffic.

The MBONE was first configured in 1992 as a test zone to enable IP multicast applications to be deployed without waiting for multicast routers to replace unicast routers across the entire Internet. The MBONE is actually a virtual network located within portions of the physical Internet. Its construction reflects several multicast zones connected together via IP multicast tunnels. When it was created in 1992, the MBONE spanned four countries and 40 subnetworks; today it spans over 25 countries and thousands of subnetworks.

You can connect to the MBONE through most Internet service providers (ISPs). You can use it to test multicast applications and technology or to connect private multicast LANs. Some organizations broadcast public information over the MBONE; examples include IETF (Internet Engineering Task Force) meetings and NASA (National Aeronautics and Space Administration, United States) space shuttle launches.

Key Concepts

This section describes several terms and concepts related to IP multicast routing.

Traffic Movement

Application sources generate the majority of IP multicast packets, but group members and routers that are communicating (DVMRP and IGMP messages) to establish the delivery path also generate IP multicast packets.

Traffic from application sources always travels in one direction — *downstream* from the source to group members. Using various protocols, network devices are responsible for determining where group members exist and coordinating a loop-free delivery path from the source to them.

Traffic that relates to the delivery path can travel both *upstream* and *downstream* — between routers and between routers and group members.

IP Multicast Groups

Users can join or leave an IP multicast group at any time. Users request and cancel membership through mechanisms built into their desktop application — perhaps visible to the user as *Go* and *Quit* buttons. There are no restrictions on the physical location or number of members in a group. A user may belong to one or more groups at any time.

Source-Group Pairs

Each IP multicast transmission can be linked to a unique pairing of a source address and multicast group address (destination address). In addition, network devices form a unique delivery path for each source-group pair. Multicast routers and switches track information about each source-group pair — mainly, the location of group members — and dynamically adjust the delivery path to ensure that IP multicast packets are delivered only where they need to go.

Multicast Addresses

A multicast packet differs from a unicast packet by the presence of a *multicast group address* in the destination address field of the IP header. IP multicast uses a Class D destination address format, which has the high-order four bits set to *1-1-1-0* followed by a 28-bit multicast group identifier.

Registered Groups

The Internet Assigned Numbers Authority (IANA) maintains a list of registered IP multicast groups. Expressed in standard dotted decimal notation, group addresses range from 224.0.0.0 – 239.255.255.255 and are classified by IANA as follows:

- Addresses 224.0.0.0 – 224.0.0.225 are reserved for use by protocols and other special functions. See Table 45 for examples of permanent reserved addresses, or for a complete and current list, visit the IANA Web site:

  `http://www.iana.org`

- Addresses 224.0.1.0 – 239.255.255.255 are either assigned to various multicast applications or remain unassigned. From this range, addresses 239.0.0.0 – 239.255.255.255 are reserved for site-local applications, not Internet-wide applications.

Table 45 Examples of Class D Permanent Address Assignments

Address	Meaning
224.0.0.0	Base Address (Reserved)
224.0.0.1	All systems on this subnet
224.0.0.2	All routers on this subnet
224.0.0.4	All DVMRP routers
224.0.0.5	All OSPF routers
224.0.0.6	All OSPF designated routers
224.0.0.7	All ST routers
224.0.0.8	All ST hosts
224.0.0.9	All RIP version 2 routers
224.0.0.11	Mobile agents
224.0.0.12	DHCP server/relay agent
224.0.0.13	All PIM routers
224.0.0.14	RSVP, Encapsulation
224.0.0.15	All CBT routers

Reserved MAC Addresses

IANA also controls a reserved portion of the IEEE-802 MAC-layer multicast address space. All addresses in this block use hexadecimal format and begin with 01-00-5E. A simple procedure maps Class D addresses to this block, so that IP multicasting can take advantage of the hardware-level multicasting supported by network interface cards (NICs).

The mapping process involves placing the low-order 23 bits of the Class D address (binary format) into the low-order 23 bits of the MAC address (hexadecimal format). For example, the Layer 3 address 224.10.8.5 maps to the Layer 2 MAC address 01-00-5E-*0A-08-05*.

To send a multicast packet, a source station inserts the Class D address in the IP packet, the network interface card maps that address to a IEEE-802 Ethernet multicast MAC address, and sends the frame. A host that wants to receive packets that are addressed to this group notifies its IP layer as such.

How IGMP Supports IP Multicast

IGMP provides a way for routers and switches to learn where group members exist on a network, and thus provides a critical function in the IP multicast packet delivery process.

Electing the Querier

On each subnetwork or broadcast domain (VLAN), the communication between routers, switches, and group members begins with one IGMP-capable device being elected as the *querier* — that is, the device that asks all hosts to respond with a report of the IP multicast groups that they wish to join or to which they already belong. The querier is always the device with the lowest IP address in the subnetwork. It can be a router or a Layer 2 switch. The network traffic flows most efficiently if the querier is positioned close to the source of the IP multicast traffic.

Query Messages

The querier normally sends messages called *IGMP Host Membership Query Messages*, or *queries*, every 125 seconds. All the hosts hear the query because it is addressed to 224.0.0.1, the *all systems on this subnetwork* Class D address. A query is not forwarded beyond the subnetwork from which it originates.

Host Messages

Hosts use IGMP to build their own types of IP multicast messages, as described in this section.

Response to Queries

Hosts respond to queries with *IGMP Host Membership Report* messages, or simply *IGMP reports*. These reports do not travel beyond their origin subnetworks, and hosts send them at random intervals to prevent the querier from being overwhelmed.

A host sends a separate report for each group that it wants to join or to which it currently belongs. Hosts do not send reports if they are not group members.

If a router does not receive at least one host report for a particular group after two queries, the router assumes that members no longer exist and it prunes the interface for that source-group spanning tree.

Join Message

Rather than wait for a query, a host can also send an IGMP report on its own initiative to inform the querier that it wants to begin receiving a transmission for a specific group (perhaps by clicking a *Go* or *Start* button on the client interface). This is called a *join* message. The benefit is faster transmission linkages, especially if the host is the first group member on the subnetwork.

Leave-Group Messages

Leave-group messages are a type of host message defined in IGMP version 2. If a host wants to leave an IP multicast group, it issues a leave-group message addressed to 224.0.0.2, the *all routers in this subnetwork* Class D address. Upon receiving such a message, the querier determines whether that host is the last group member on the subnetwork by issuing a *group-specific query*.

Leave-group messages lower *leave latency* — that is, the time between when the last group member on a given subnetwork sends a report and when a router stops forwarding traffic for that group onto the subnetwork. This process conserves bandwidth. The alternative is for the router to wait for at least two queries to go unanswered before pruning that subnetwork from the delivery tree.

Role of IGMP in IP Multicast Filtering

To further refine the IP multicast delivery process and maximize bandwidth efficiency, a Layer 3 device filters IP multicast packets on appropriate ports using a process called *IGMP snooping*. Both bridged interfaces and routed interfaces record which ports receive host IGMP reports and then set their filters accordingly so that IP multicast traffic for particular groups is not forwarded on ports or VLANs that do not require it.

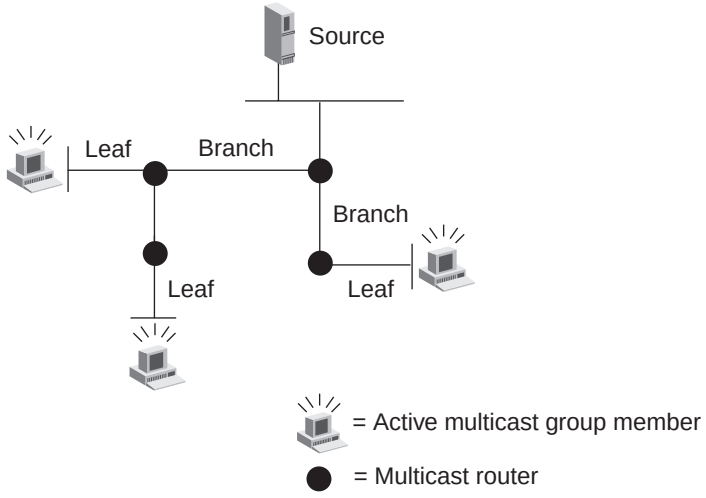
How DVMRP Supports IP Multicast

DVMRP is a distance-vector routing protocol that allows routers to establish shortest-path, source-rooted, IP multicast delivery trees. While it is similar to the Routing Information Protocol (RIP), one important difference is that DVMRP focuses on the *previous hop* back to a multicast source, not the next hop to a destination. Multicast routers are concerned with moving packets *away from the source* on a loopless path so that multicast storms do not occur.

Spanning Tree Delivery

DVMRP version 3.x uses the Reverse Path Multicast (RPM) algorithm to construct a delivery tree that begins at the source and spans out to reach group members on a loopless path through the network. Hence, DVMRP seeks to form a *source-rooted spanning tree* for each source-group pair. The shape of each tree changes dynamically, depending on the location of hosts that join and leave the group. As shown in Figure 68, any routing interface that contains group members is called a *leaf interface*.

Figure 68 Sample IP Multicast Spanning Tree



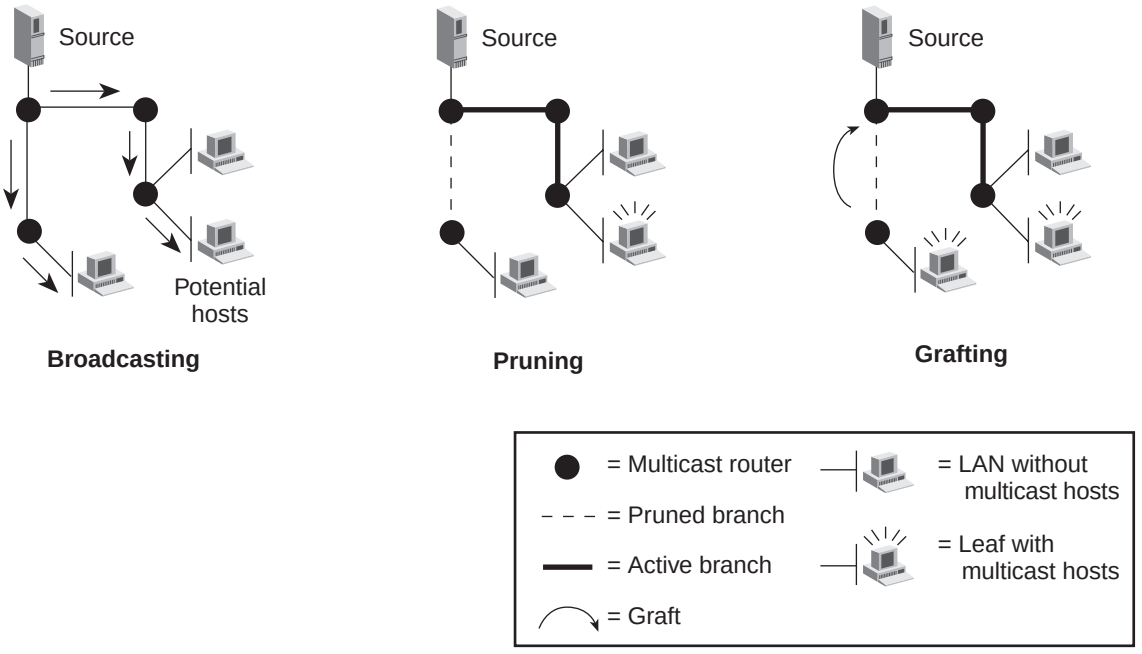
The term *spanning tree* applies to any loopless graph that spans intelligent nodes. The DVMRP spanning tree structure provides only one active path to connect any two multicast routers in the network. This approach provides a logical, efficient path to reach group members and prevents multicast storms from decreasing network performance.

The Spanning Tree Algorithm that is specified in the IEEE 802.1D MAC Bridges base standard is a different implementation of the spanning tree concept; it is not used with IP multicast.

Managing the Spanning Tree

RPM uses three main techniques to dynamically adjust the shape of an IP multicast spanning tree: broadcasting, pruning, and grafting. These techniques balance the goal of an efficient delivery path with the goal of effective service for all potential group members. Figure 69 shows the broadcasting, pruning, and grafting processes.

Figure 69 RPM Techniques for Managing the Multicast Spanning Tree



Interface Relationships

The interface on which a router receives source-origin traffic for a given source-group pair is called the incoming or *parent* interface. Each interface over which the router forwards source-group traffic is called an outgoing or *child* interface. A child interface on one router can:

- **Be a leaf interface** — A subnetwork with group members
- **Lead to the parent interface of a downstream router** — The next router in the delivery path to reach group members

Broadcasting

The first packet for any source-group pair is broadcast across the entire network, as far as packet time-to-live (TTL) and router TTL thresholds permit. If a packet arrives on an interface that the router determines to be the shortest path back to the source (by comparing interface metrics), then the router forwards the packet on all interfaces except the incoming interface. Downstream routers quickly send either:

- Prune messages (explained next) to upstream routers if their interfaces do not lead to group members
- IGMP reports if they want to continue receiving traffic for that source-group pair.

Some IP multicast applications try to actively send traffic on the network, even if no group members are requesting their traffic. Your system can detect which ports lead to routers and send these infrequent broadcast *packets only to those ports. Otherwise, the system filters all IP multicast group traffic for which it has received no IGMP Reports or graft messages.*

Pruning

A parent interface transmits *a prune* message to its upstream neighboring router if there are no group members on its child interfaces. A prune message directs the upstream router not to forward packets for a particular source-group pair in the future. Prune messages always affect the entire routing interface; they cannot be targeted to prune individual port segments that belong to an interface (IGMP snooping effectively achieves this, however).

Prune messages always begin at the leaf routers and are sent one hop back toward the source. Each successive router determines whether to prune its connections.

Inside the prune message is a prune *lifetime*, or prune *timer*, which is a period of time for which the prune message is valid. When the prune lifetime expires, the interface is added back into the multicast delivery tree — that is, until it generates another prune message.

Even though routers must use memory to store membership and prune information, this approach regains the bandwidth that would have been wasted on branches that do not lead to group members.

Grafting

If a router that has previously sent a prune message discovers a new group member (from IGMP Reports) on one of its connections, it sends a *graft* message to the previous hop router. When an upstream router receives this message, it cancels the prune message it previously received. Hop by hop, graft messages cascade back toward the source until they reach the nearest live branch point on the IP multicast spanning tree.

DVMRP Interface Characteristics

All DVMRP interfaces and DVMRP tunnels have two characteristics: a metric that specifies the *cost* for the interface and a time-to-live (TTL) threshold.

- **Metric Value** — The DVMRP metric is a numeric value or cost for that path. The higher the assigned cost, the less likely it is that the multicast packets will be routed over that interface (provided that other path options exist).
- **TTL Threshold** — Each IP multicast packet uses the TTL field of the IP header to provide a scope-limiting parameter for routers to work with. The initial value that the source sets in the TTL field controls the number of router hops that the IP multicast packet can make through the network. Each time that a router forwards a packet, it decrements the packet TTL by one. As long as the multicast packet TTL is greater than the TTL threshold of the multicast router interface, the router forwards the packet. If not, the router filters (drops) the packet.

In all cases where the multicast router drops multicast packets, the router does not provide an error notification to the source because IP multicast is a connection-less technology.

Key Guidelines for Implementation

You need to enable IP multicast routing features only if network users require access to IP multicast application traffic from local or remote sources.

Configuration Procedure

To activate IP multicast routing and filtering capabilities in your system, follow this general procedure:

1. Configure VLANs and IP routing interfaces on the system. See Chapter 9 for more information about VLANs. See Chapter 11 for more information about IP routing.
2. Ensure that IGMP snooping and querying functions are enabled on the system.

   For general information about IGMP, see "How IGMP Supports IP Multicast" earlier in this chapter.

   For information about configuring IGMP functions on your system, see "Configuring IGMP Options" later in this chapter.
3. Enable DVMRP on each interface that is to perform IP multicast routing.

   You can modify the default TTL threshold and DVMRP metric values that the system assigns to each interface.

   For general information about DVMRP see "How DVMRP Supports IP Multicast" earlier in this chapter.

   For information about configuring DVMRP, see "Configuring DVMRP Interfaces" later in this chapter.
4. If your network requires a multicast tunnel and if your system is going to serve as one or more tunnel endpoints, configure the tunnels now. You can configure up to 8 tunnels per system. Remember to configure the tunnels on the remote systems as well.

   See "Configuring DVMRP Tunnels" later in this chapter.
5. Configure a default route on an interface (if applicable to your network).

   See "Configuring DVMRP Default Routes" later in this chapter.

6 View the various displays, routing table and cache to see how the system is processing IP multicast traffic.

   See "Viewing the DVMRP Routing Table" and "Viewing the DVMRP Cache" later in this chapter.

7 Use the traceroute option for troubleshooting or to determine the traffic paths.

   See "Using IP Multicast Traceroute" later in this chapter.

Impact of Multicast Limits

As described in Chapter 7, you can use the `bridge port multicastLimit` option to set per-port multicast limits. This optional feature can prevent segments of your network from being adversely affected by multicast or broadcast storms. If network users have trouble receiving IP multicast application traffic, verify that bridge ports are not configured with a multicast limit that is too low.

Impact of IEEE 802.1Q on Multicasts

Multicasting in 802.1Q VLAN tagging environments may have performance implications for your system. Specifically, if you have multiple VLANs associated with a single port, the system is forced to replicate multicast packets to each VLAN that has multicast group members, even if the path to reach the members is the same physical link. To achieve wirespeed multicast performance, 3Com recommends that you configure only one VLAN per port. Contact your 3Com representative about network design options.

Protocol Interoperability

Routing protocols other than DVMRP exist to support IP multicast functions. Interoperability issues between these routing protocols require that you plan your routing infrastructure carefully.

Configuring IGMP Options

You can enable or disable IGMP snooping and querying functions, set the interface time-to-live (TTL) threshold, and obtain summary and detail displays of IGMP-related information.

Querying and Snooping Modes

Your system divides IGMP functions into two modes:

- **Querying** — Allows an IP multicast routing interface to function as the querier if so elected.
- **Snooping** — Allows the system to forward multicast packets only to the appropriate ports within its routing or bridging interfaces.

Important Considerations

- Both modes are enabled as the factory default. These settings apply to the entire system. You cannot enable or disable snooping or querying on specific interfaces.
- 3Com recommends that you keep both modes enabled at all times. They add little processing overhead to the system.

Configuring DVMRP Interfaces

DVMRP is the protocol used to develop source-rooted spanning trees between routers in the network. You can enable or disable DVMRP on individual routing interfaces.

Important Considerations

- The default setting for DVMRP on each new interface is `disabled`.
- If DVMRP is disabled, the interface cannot participate in forming IP multicast spanning trees. If the Internet Group Management Protocol (IGMP) is enabled, the system can still forward IP multicast traffic.
- Enabling DVMRP causes the system to assign default values of `1` for both the TTL threshold and metric on the interface. You can modify these values at any time.
- A TTL threshold value of `1` means the interface forwards all IP multicast packets except those which have expired (packet TTL is 0). Before you change the TTL threshold value, consider the relative location of the system in the network and your networking objectives.

Table 46 lists conventional numeric values and network objectives.

Table 46 Conventional TTL Scope Control Values

TTL Value	Objective
0	Restricted to the same host
1	Restricted to the same subnetwork
16	Restricted to the same site
64	Restricted to the same region
128	Restricted to the same continent
255	Unrestricted in scope

Configuring DVMRP Tunnels

A DVMRP tunnel allows IP multicast packets to traverse a portion of your network infrastructure that is not multicast-aware. In your system, you can define tunnels, modify tunnel characteristics, display information about tunnels you have defined, and remove tunnels.

Important Considerations

- All networks do not require DVMRP tunnels. A network needs a tunnel only if IP multicast packets must go through one or more unicast routers to reach IP multicast group members.
- You can configure any routing interface on the system to be a DVMRP tunnel end point. The other tunnel end point must be a multicast interface on a different system and subnetwork.
- You can configure up to 8 tunnels per CoreBuilder 3500 system.
- Before you can define a tunnel end point, you must configure a routing interface and enable DVMRP on the interface. Think of a tunnel end point as being layered on top of an existing IP multicast routing interface.
- To define a tunnel, you specify the following tunnel characteristics:
  - The index number of the local router interface that serves as the tunnel end point.
  - The IP address of the destination multicast router. *This address must be a remote address*. The destination multicast router cannot be directly connected to the same subnetwork.
  - When you define a tunnel, the system assigns tunnel metric and tunnel TTL threshold values of 1. You can modify these at any time.

- You must define the tunnel on both end points — that is, on both the local system and the remote system — even though you specify the address of the remote router interface in the local system.
- DVMRP interfaces and tunnels have similar characteristics (metric and TTL threshold), but the tunnel characteristics do not have to match the interface characteristics.
- If you try to remove an IP interface, and you have a tunnel defined on that interface, the system warns you with an error message. You must remove the tunnel before you can remove the IP interface.
- You can define multiple multicast tunnel end points on the same local routing interface, but each must lead to different remote end points.
- When you define a tunnel, the system assigns a tunnel index number to it. The multicast tunnel display lists tunnels in ascending order by the tunnel index number. Tunnel index numbers provide a way to identify and remove individual tunnels, which is especially useful when multiple tunnel end points are configured on the same routing interface.
- When you remove a tunnel, the system does not dynamically reorder remaining tunnels in the multicast tunnel display. For example, if you had three tunnels with tunnel index numbers 1, 2, and 3 and you then removed tunnel 2, the multicast tunnel display lists the remaining tunnels and identifies them with their original tunnel index numbers (1 and 3). The system does not dynamically reassign tunnel index numbers (does not change 3 to 2). In this example, the system assigns tunnel index 2 to the next *new* tunnel that you define. After the system uses index 2, it can assign index 4 to the next new tunnel, and so on.
- Removing a tunnel end point on one system destroys that tunnel's functionality, but 3Com recommends that you remove the tunnel configuration from both systems.

Configuring DVMRP Default Routes

You can configure a default route for IP multicast traffic on any DVMRP routing interface in the system.

How Default Routes Work

If an interface is configured as a default route, it advertises source 0.0.0.0 to neighboring DVMRP routers. In their DVMRP routing tables, these neighboring routers list 0.0.0.0 as a source and list the advertising router interface as the *gateway* to reach that source.

Thus, if a neighboring router receives an IP multicast packet for which it has no normal routing information in its routing table, instead of filtering the packet, the router forwards it to the router which advertises the default route.

How to Configure A Default Route

To configure a default route on an interface, follow these steps:

1. Specify the interface index number.
2. Set the default route metric (cost).

   Enter a value from 1 through 32 to signify the cost of the route.

   The value 0 indicates that no default route is configured.
3. Set the default route mode.

   There are two options:

   - **all** — The interface advertises the default route plus all other known routes to neighboring DVMRP routers.
   - **only** — The interface advertises only the default route to neighboring DVMRP routers.

Important Considerations

- If the system learns a default route, it propagates it no matter which mode is set on a given interface.
- The system allows you to configure an interface as a DVMRP default route, even when DVMRP is disabled on the interface. If DVMRP is disabled, the interface does not advertise itself as a default route.

Viewing the DVMRP Routing Table

Your system records DVMRP route information in a table that you can access from the management interface. Your system learns source-based route information from neighboring DVMRP routers and also advertises routes that it learns to its neighbors. The routing table does not consider group membership or prune messages. It simply records path information it has learned on its own or from other routers, including:

- Subnetworks from which IP multicast traffic originates
- Upstream routers (gateway) from which the system should expect to receive traffic from origin subnetworks
- Index number of the interface (parent) that is connected to the upstream router
- Outgoing interfaces (children) on which it could forward traffic if group members exist.

The system may never receive process IP multicast traffic from the sources listed in the routing table. Receipt of IP multicast traffic depends on whether group members exist on directly-attached subnetworks or on subnetworks from downstream routers.

See the *Command Reference Guide* for definitions of the fields of information and symbols used in the DVMRP route display.

Viewing the DVMRP Cache

Your system records information about the IP multicast group traffic it has processed. You can see this information in the DVMRP cache.

To display the DVMRP cache, the system prompts you to enter:

- A multicast source address
- A multicast group address

This process limits the cache table to displaying information for one source-group pair at a time. To display cache information for all source-group pairs, enter `255.255.255.255` at both prompts.

See the *Command Reference Guide* for definitions of the fields of information and symbols used in cache display.

Using IP Multicast Traceroute

You can perform an IP multicast traceroute from the system management interface. The ability to trace the path of a IP multicast group packet from a source to a particular destination is desirable for troubleshooting purposes.

Unlike unicast traceroute, IP multicast traceroute requires the ability for routers to understand a special IGMP packet type and the related processes.

Beginning a trace from an IP multicast source would be difficult because, at forks in the network paths, there is no way to determine which direction to take. You would have to flood the entire tree and wait for responses (or the lack thereof) to find the path. Thus, a more efficient approach is to start at the destination and travel backwards toward the source, using the knowledge held by IP multicast routing protocols that work by calculating previous hops back toward sources.

An IP multicast traceroute proceeds as follows:

1. At the destination node (your system), you specify a source and group address.
2. The system sends a traceroute Query packet to the last-hop multicast router (the upstream router for this source-group pair).
3. The last-hop router turns the Query packet into a Request packet by adding a response data block containing its interface addresses and packet statistics. It then forwards the Request packet via unicast to the router that it believes is the previous hop for the given source-group pair.
4. Each previous hop router adds its response data to the end of the Request packet, then forwards it via unicast to the next previous hop router.
5. Finally, the first-hop router — that is, the router that believes that the source-group packets originate on one of its directly-attached subnetworks — adds its data, changes the Request packet to a Response packet, and sends the completed response back to the destination node that issued the traceroute query.
6. You see a display that shows IP addresses of the interfaces that span from your system back to the source that you specify. The display also shows the number of hops back to those interfaces, the multicast routing protocols used, and the amount of time it takes to reach each hop from the receiver.

Important Considerations

- When using IP multicast traceroute, the system assumes that it is the destination for the source-group traffic. You cannot enter a different destination address.
- A Response packet may be returned to your system before reaching the first-hop router if a fatal error condition such as "no route" is encountered along the path. All interim devices must support IP multicast traceroute for you to see a complete path on the display.

Standards, Protocols, and Related Reading

DVMRP was first defined in RFC 1075 and has been modified in various Internet drafts. IGMP was first defined in RFC 1112 and has been modified in various Internet drafts. To learn more about DVMRP and IGMP, IP multicast technology, or related events, consult the following Web resources:

- **`http://www.3com.com`**
- **`http://www.ipmulticast.com`**
- **`http://www.ietf.org`**
- **`http://www.stardust.com`**

14 OPEN SHORTEST PATH FIRST (OSPF)

This chapter provides guidelines and other key information about how to configure Open Shortest Path First (OSPF) on your system.

This information includes:

- OSPF Overview
- Key Concepts
- Key Guidelines for Implementing OSPF
- Autonomous System Boundary Routers
- Areas
- Default Route Metric
- OSPF Interfaces
- Link State Databases
- Neighbors
- Router IDs
- OSPF Memory Partition
- Stub Default Metrics
- Virtual Links
- OSPF Routing Policies
- OSPF Statistics

You manage OSPF routing from the `ip ospf` *menu of the Administration Console. See the* Command Reference Guide.

OSPF Overview

The OSPF link-state protocol dynamically responds to changes in network topology that occur within a group of networks and routers known as an *autonomous system*. OSPF tracks the states of links and routers in each autonomous system, and when a change occurs, calculates new routes based on the new topology. The OSPF protocol responds to network topology changes with a minimum of administrator involvement and routing traffic.

All OSPF routers within an autonomous system build and synchronize databases of the autonomous system's network topology. Using its database, each router calculates the shortest path trees to every destination within the autonomous system. With this dynamic table of shortest paths, OSPF converges on an optimum route faster than other routing algorithms, such as the Routing Information Protocol (RIP).

Routers that use a distance-vector protocol like RIP periodically exchange all or a portion of their tables, but only with their neighbors. Routers using a link-state protocol like OSPF send small portions of their tables throughout the network by flooding.

For information about how to perform IP routing, see Chapter 11.

Features

Your system supports OSPF Version 2 as defined in RFC 1583. OSPF routing on your system includes these features:

- **Areas** — You can subdivide an autonomous system (AS) into more manageable contiguous networks called areas. Areas increase stability, conserve router resources, and support route summarization — the consolidation of network addresses. For more information, see "Areas" later in this chapter.
- **Default route metric** — You can configure a router to advertise itself as the default router for the area, and you can specify a cost to be advertised with the default route. When area routers fail to find a specific match for a packet's destination, the router then forwards the packet to the default router, which then forwards the packet to the most logical destination. For more information, see "Default Route Metric" later in this chapter.

- **OSPF interfaces** — An OSPF interface is an IP interface that you configure to send and receive OSPF traffic. When you configure an OSPF interface, you define the behavior and role of the interface within the OSPF routing domain. For example, router priority determines designated router selection, cost determines the expense associated with using the interface, and the Hello interval directly affects how fast topological changes are detected. For more information, see "OSPF Interfaces" later in this chapter.
- **Link state databases** — OSPF routers advertise routes using link state advertisements. The link state database contains the link state advertisements from throughout the area to which an OSPF interface is attached. For more information, see "Link State Databases" later in this chapter.
- **Neighbors** — OSPF interfaces attached to a common network are called neighbors; adjacent neighbors exchange link state database information. On broadcast networks, neighbors are discovered dynamically using the Hello protocol. On nonbroadcast multiaccess networks, you must statically configure neighbors. Your system allows you to display all neighbors in the locality of the router, as well configure them when needed. For more information, see "Neighbors" later in this chapter.
- **Router IDs** — A router ID identifies the router to other routers within the autonomous system. In addition, it serves as a tie-breaker in the designated router election. You systems gives you three methods by which you can configure a router ID for an OSPF interface. For more information, see "Router IDs" later in this chapter.
- **OSPF memory partition** — You can display how much memory your system allocates for OSPF data processing and adjust this memory allocation if needed. For more information, see "OSPF Memory Partition" later in this chapter.
- **Stub default metrics** — External link state advertisements are not propagated into stub areas. Instead, the area border router for a stub area injects a single external default route into the area. Your system allows you to configure an area border router to advertise a single external default route into the stub area while specifying the cost of the default route. For more information, see "Stub Default Metrics" later in this chapter.

- **Virtual links** — All areas of an OSPF routing domain must connect to the backbone area. In cases where an area does not have direct, physical access to the backbone, you can configure a logical connection to the backbone, called a *virtual link*. Virtual links can also add fault-tolerance and redundancy to the backbone. For more information, see "Virtual Links" later in this chapter.
- **OSPF routing policies** — Routing policies let you control what external routes OSPF routers store in their routing tables, as well as what external routes they advertise. Although routing policies are not part of the OSPF protocol itself, you can use them for increased security, enhanced performance, and overall control of OSPF routing data. For more information, see "OSPF Routing Policies" later in this chapter.
- **OSPF statistics** — You can also display general statistics for specific OSPF interfaces. These statistics can give you an overview of OSPF activity on the interface. For more information, see "OSPF Statistics" later in this chapter.

Benefits

The benefits of OSPF are what set it apart from both RIP and other Shortest Path First-based algorithms before it. While designing OSPF, the Internet Engineering Task Force (IETF) proposed a number of modifications which dealt with improving the existing SPF model. These modifications ranged from improving fault-tolerance to reducing routing traffic overhead. This focus toward improving the existing SPF model resulted in the following OSPF capabilities:

- **No hop count limitation** — OSPF places no limit on hop count. This capability is extremely important in larger networks. For example, a RIP network that spans more than 15 hops (15 routers) is considered unreachable. With OSPF, hop count is no longer an issue.
- **Efficient use of bandwidth for router updates** — OSPF uses IP multicast to send link-state updates only when routing changes have occurred, or once every 30 minutes. RIP, on the other hand, uses a 30 second interval. This policy ensures less processing on routers that are not listening to OSPF packets and better use of bandwidth.

- **Ability to partition the network into more manageable *areas*** — Many autonomous systems in the Internet are large and complicated to manage. OSPF allows them to be subdivided into smaller, more manageable networks or sets of contiguous networks called *areas*. You can think of an area as a generalization of an IP subnetworked network. The topology of an area is hidden from the rest of the AS, which significantly reduces routing traffic and also serves to lend the area protection from bad routing data. By partitioning the network into areas, OSPF limits the topology map required in each router. This limitation in turn conserves processing and memory requirements in each router, as well as reduces the amount of link state information being flooded onto the network.
- **Authentication for protocol exchanges** — All OSPF protocol exchanges are authenticated, which means that only known, trusted routers can participate in routing updates. OSPF supports a variety of authentication schemes, with a single scheme configured for each area. This partitioning allows some areas to use much stricter authentication than others.
- **Host-specific and network-specific route support** — OSPF supports traffic forwarding to single hosts or networks. Each network the router knows has both an IP destination address and a mask. The mask indicates the number of nodes on the network. A mask of all ones (0xffffffff) indicates a presence of a single node on the network (called a *stub network*).
- **Support for designated and back-up designated routers** — OSPF works by exchanging information between *adjacent* routers, not *neighboring* routers. To avoid the need for every router on a LAN or area to talk to every other router on a multiaccess network (a network that has at least two attached routers), one router is elected as the designated router. The designated router is considered adjacent to all other routers in the area and exchanges information with them. Routers that are not adjacent to each other do not exchange information. Therefore, instead of all routers on the network sending Link State Advertisements (LSAs), only the designated router sends LSAs. This feature significantly reduces data and routing traffic.

- **Support for virtual links to noncontiguous areas** — As discussed earlier, OSPF can partition large autonomous systems into smaller, more manageable subdivisions, called areas. An OSPF backbone is responsible for distributing routing information between the areas of an autonomous system. This backbone itself has all the properties of an area and consists of those networks that are not contained in any area. Although the backbone must be contiguous, backbone routers can also be connected by means of a virtual link. Virtual links can be configured between any two backbone routers that have an interface to a common nonbackbone area. OSPF treats two routers that are joined by a virtual link as if they were connected by an unnumbered point-to-point network. For more information, see "Virtual Links" later in this chapter.
- **Variable length subnet mask support** — OSPF considers both the IP address and subnet mask in determining the best route for a packet. An IP address mask is distributed with each advertised route. The mask indicates the range of addresses that are being described by the particular route. Including this mask enables the implementation of variable-length subnet masks (VLSMs), which means that a single IP network number can be *subnetworked* or broken up into many subnetworks of various sizes. When networks are subnetworked, OSPF forwards each IP packet to the network that is the best match for the packet's destination. It determines the best match by examining both the network address and the mask for each destination and finding the longest or most specific match. VLSM support is a key advantage for OSPF, especially when you consider the shortage of IP addresses. Another advantage is its flexibility.
- **Ability to import non-OSPF routing information** — Connectivity from one autonomous system to another is achieved through OSPF autonomous system boundary routers (ASBRs). ASBRs can import external link advertisements that contain information about external networks from other protocols like RIP and redistribute them as LSAs to the OSPF network. In this way, ASBRs flood information about external networks to routers within the OSPF network.
- **Assurance of loop-free networks** — The algorithm that dynamically calculates routing paths within the autonomous system does not generate a path with internal loops.

Key Concepts

Before you configure OSPF on your system, review the following key concepts and terms discussed in these sections:

- Autonomous Systems
- Areas
- Neighbors and Adjacency
- Router Types
- Protocol Packets
- How OSPF Routing Works

Autonomous Systems

An *autonomous system* consists of a set of OSPF routers that exchange routing information. The network shown in Figure 70 later in this chapter contains two autonomous systems.

Using identical topology databases, each router in an autonomous system calculates shortest-path routes from itself to every known destination in the autonomous system. The routers create their topology databases using the data in link state advertisements (LSAs) from other routers in the autonomous system.

Areas

Autonomous systems can be subdivided into smaller, more manageable, groups of contiguous networks called *areas*. Each OSPF router in an area must have identical topological link state databases. These databases may include area links, summarized links, and external links that depict the topology of the autonomous system.

Neighbors and Adjacency

Instead of each router sending routing information to every other router on the network, OSPF routers establish adjacencies among neighboring routers. Only adjacent routers exchange routing information. This information is exchanged using Database Description packets, which are used to describe the contents of each router's link state database.

Router Types OSPF routers serve several different, often overlapping, functions:

- **Internal routers** — Internal routers connect only to networks that belong to the same area. An internal router runs one copy of the OSPF algorithm and maintains routing data only for its area.

  In Figure 70, backbone area 0 and routers 1, 2, 3, and 4 are internal routers. In area 1, routers 5 and 6 are internal routers.

- **Backbone routers** — Backbone routers have an interface to the backbone area. Area border routers are always backbone routers because you must configure them as being within the backbone area or connected to it by a virtual link.

  In Figure 70, routers 1, 2, 3, and 4, and area border routers 1, 2, 3, and 4 are all backbone routers.

- **Area border routers (ABRs)** — Area border routers connect directly to networks in two or more areas. An area border router runs a separate copy of the OSPF algorithm and maintains separate routing data for each area that is connected to it (including the backbone area). Area border routers also send configuration summaries for their attached areas to the backbone area, which then distributes this information to other OSPF areas in the autonomous system.

  In Figure 70, four area border routers link the areas in autonomous system A.

- **Autonomous system boundary routers (ASBRs)** — Autonomous system boundary routers exchange their autonomous system topology data with boundary routers in other autonomous systems. Every router inside an autonomous system knows how to reach the boundary routers for its autonomous system.

  In Figure 70, two ASBRs control traffic between two autonomous systems.

- **Designated routers (DRs)** — Designated routers advertise network link states for attached network segments. A link state advertisement lists all routers that are connected to a segment.

  The DR is considered adjacent to all routers in its area. As a result, the DR exchanges routing data with all routers that are connected to its network segment.

- **Backup designated routers (BDRs)** — Backup designated routers are given a lower priority value than the DR. They take over DR functions if the DR fails.

Router IDs

The OSPF router ID identifies a router to other routers within an autonomous system. OSPF uses three types of router identifiers, which take the form of an IP address:

- **Default** — An arbitrary ID that the system generates and uses as the default router ID
- **Interface** — The address of an IP interface on the router
- **Address** — An arbitrary user-defined ID in the form of an IP address

You cannot set the router id to either 0.0.0.0 or 255.255.255.255.

Protocol Packets

The OSPF protocol uses these types of packets:

- **Hello packets** — Router interfaces periodically transmit hello packets to identify and maintain communications with their neighbors. In nonmulticast networks, routers find neighbors by sending unicast hello packets to other statically configured routers.
- **Database description packets** — Neighbor routers use database description packets to synchronize their link state summary databases.
- **Link state request packets** — To collect network topology data, routers transmit link state request packets to their neighbors on the segment.
- **Link state update packets** — On receiving a link state request packet, a router floods packets containing its LSA data into the area or autonomous system that it serves. The information contained in the packets depends on the router's location and function in the network.
- **Link state ack(nowledge) packets** — Routers use these packets to acknowledge receipt of link state update packets.

How OSPF Routing Works

This section summarizes how the OSPF algorithm works for a router that meets these characteristics:

- Lies within an autonomous system area (an interior router)
- Is attached to a multiaccess network
- Is configured to be the designated router for its network

Starting Up

When the router starts, the interfaces that are configured to run OSPF begin in the *down* state. When the lower-level IP protocols indicate that an interface is available, the interface moves to the *waiting* state. It remains in this state until the designated router and backup designated router are chosen.

Finding Neighbors

The router sends out hello packets to locate its network neighbors. These packets also list the routers from which the sending router has *received* hello packets. When a router detects its own address in another router's hello packet, the two routers establish two-way communications as neighbors.

Establishing Adjacencies

If neighboring OSPF routers succeed in exchanging and synchronizing their link state databases, they appear as *adjacent* in all router and network link advertisements.

Electing the Backup Designated Router

OSPF selects a backup designated router for the network segment. This router takes over as the designated router if the current designated router fails.

The OSPF algorithm first eliminates all routers that have an assigned priority of *0*. OSPF then selects the backup designated router from among the routers on the segment that have *not* declared themselves to be the designated router (based on their configuration settings). If some routers have already declared themselves to be the backup designated router, OSPF limits the selection to one of them.

OSPF selects the candidate router with the highest priority. If candidate routers have the same priority, OSPF selects the router that has the highest router ID.

Electing the Designated Router

OSPF selects a designated router, which originates LSAs on behalf of the network segment. These advertisements list all routers (including the designated router) that are attached to the segment. The designated router also floods LSA packets throughout the segment to allow its neighbors to update their databases.

The OSPF algorithm first eliminates all routers that have an assigned priority of *0*. OSPF then selects a designated router from among the routers that have declared themselves to be the designated router (based on their configuration settings). If no routers have declared their candidacy, the backup designated router becomes the designated router, and OSPF selects a new backup designated router.

OSPF selects the candidate router with the highest priority. If candidate routers have the same priority, OSPF selects the router that has the highest router ID.

The designated router then becomes adjacent to all other routers on the network segment by sending Hello packets to them.

Calculating Shortest Path Trees

OSPF routers collect raw topological data from the LSAs that they receive. Each router then prunes this data down to a tree of the shortest network paths centered on itself. In a series of iterations, the router examines the total cost to reach each router or network node in its domain. By discarding all but the lowest-cost path to each destination, the router builds a shortest path tree to each destination, which it uses until the network topology changes.

Routing Packets

A packet's source and destination determine the routers that move it:

- **Intraarea** — When a packet's source and destination are in the same area, the packet is routed using internal router databases. No routers are used outside the area.
- **Interarea** — When a packet's source and destination are in different areas, the topology databases in the backbone area dictate the paths that are taken between areas.

You can use virtual links to influence the routes that are taken for interarea traffic. See "Virtual Links" later in this chapter.

- **To a stub area** — When a packet's destination is in a stub area (an area that does not accept external route advertisements), OSPF uses the area's predefined default route. You configure default routing in area border routers that serve an OSPF stub area, such as area border router 1 in Figure 70. For more information, see "Stub Default Metrics" later in this chapter.
- **To a different autonomous system** — When a packet's source and destination are in different autonomous systems, ASBRs compute the routing paths using data obtained from another protocol, such as the Border Gateway Protocol. The boundary routers flood these external routes throughout all areas in the autonomous system except stub areas.

Key Guidelines for Implementing OSPF

Consider the following guidelines when you design a scalable and dependable OSPF internetwork:

These parameters must be consistent across all routers

The following OSPF interface parameters must be consistent across all routers on an attached network:

- Hello interval
- Dead interval
- Password

Addressing scheme

The addressing structure that you implement can affect both the scalability and the performance of your OSPF internetwork. Consider the following guidelines when you define an addressing structure to use for your OSPF internetwork:

- Make the range of subnets that are assigned within each OSPF area should be contiguous to allow optimal summarization by area border routers (ABRs).
- Define the address space so that you can easily add new areas, restructure existing ones, or add additional routers as your network grows.

Router placement and participation

When you populate an area with OSPF routers, consider the following guidelines:

- Because OSPF uses a CPU-intensive algorithm, keep the maximum number of routers participating in OSPF exchanges in any given area to around 50. This number decreases the likelihood of performance problems that may be associated with router recalculation. If the link is of high quality and the number of routes is minimal, you can increase the number of area routers.
- Keep the maximum number of neighbors for any one router to around 60. Each time that a topological change occurs, a router exchanges information only with those neighbors with which it has formed an adjacency. On a multiaccess network, this neighbor count is only of concern to the designated and backup-designated router of an area because, on a multiaccess network, area routers do not exchange link-state information with each other. Instead, they exchange link-state information with only the designated and backup designated routers.

Autonomous System Boundary Routers

Autonomous system boundary routers (ASBRs) are the links between the OSPF autonomous system and the outside network. They exchange their autonomous system topology data with boundary routers in other autonomous systems.

ASBRs can import external link advertisements that contain information about external networks from other protocols like RIP and redistribute them as LSAs to the OSPF network. In this way, ASBRs flood information about external networks to routers within the OSPF network.

Every router inside an autonomous system knows how to reach the boundary routers for its autonomous system.

In Figure 70, two ASBRs control traffic between two autonomous systems.

Configuring an ASBR

A router becomes an ASBR as a by-product of other settings. A router becomes an ASBR if any operational in-band IP interface on the router:

- Has both OSPF and RIP disabled on that interface. Or,
- Has RIP configured as learn, advertise, or enabled on that interface.

The ASBR then generates external link state advertisements for these IP interfaces.

A router also becomes an ASBR if you have configured either of the following on the box:

- A default route metric
- Any static routes, including configuring a default route

A router *never* becomes an ASBR if:

- All of the router's interfaces reside in a stub area.

This last rule overrides all other cases where a router can become an ASBR.

- You create IP interfaces with the `ip interface` option.
- You configure RIP on IP interfaces with the `ip rip` options.
- You configure OSPF on IP interfaces with the `ip ospf` options.
- You create default route metrics with the `ip ospf defaultRouteMetric` define option.
- You create static routes with the `ip ospf policy` options.

Areas

To reduce the amount of routing information that travels through a network, and the corresponding size of the OSPF routers' topology databases, subdivide OSPF autonomous systems into *areas*. Each area has the following configurable parameters:

- **Area ID** — A 32 bit number that identifies the area to the OSPF autonomous system. The Area ID is specified in the form n.n.n.n, where $0 \leq n \leq 255$. Subdividing the autonomous system (AS) into multiple areas requires the use of a backbone area, which connects all other areas in the AS and is always assigned an area ID of 0.0.0.0.

  Although an area ID has the same superficial form as an IP address, the area ID address space is its own distinct address space.

- **Stub area** — An OSPF area that does not accept or distribute external address advertisements. Instead, the area border router generates a default external route that is advertised into the stub area for destinations outside the autonomous system. Use the stub area designation to minimize topological data that is stored in the area's routers.

- **Range** — An address that covers a range of subnetwork addresses. A range address aggregates LSAs from all of its subnetwork addresses; this aggregation is also known as *route summarization*. For more information, see "Configuring Route Summarization in ABRs" later in this chapter.

- **Default route metric** — The network cost for an OSPF default route. If a default route metric is defined, the router advertises itself as the default router to the area. For more information, see "Default Route Metric" later in this chapter.

Types of Areas

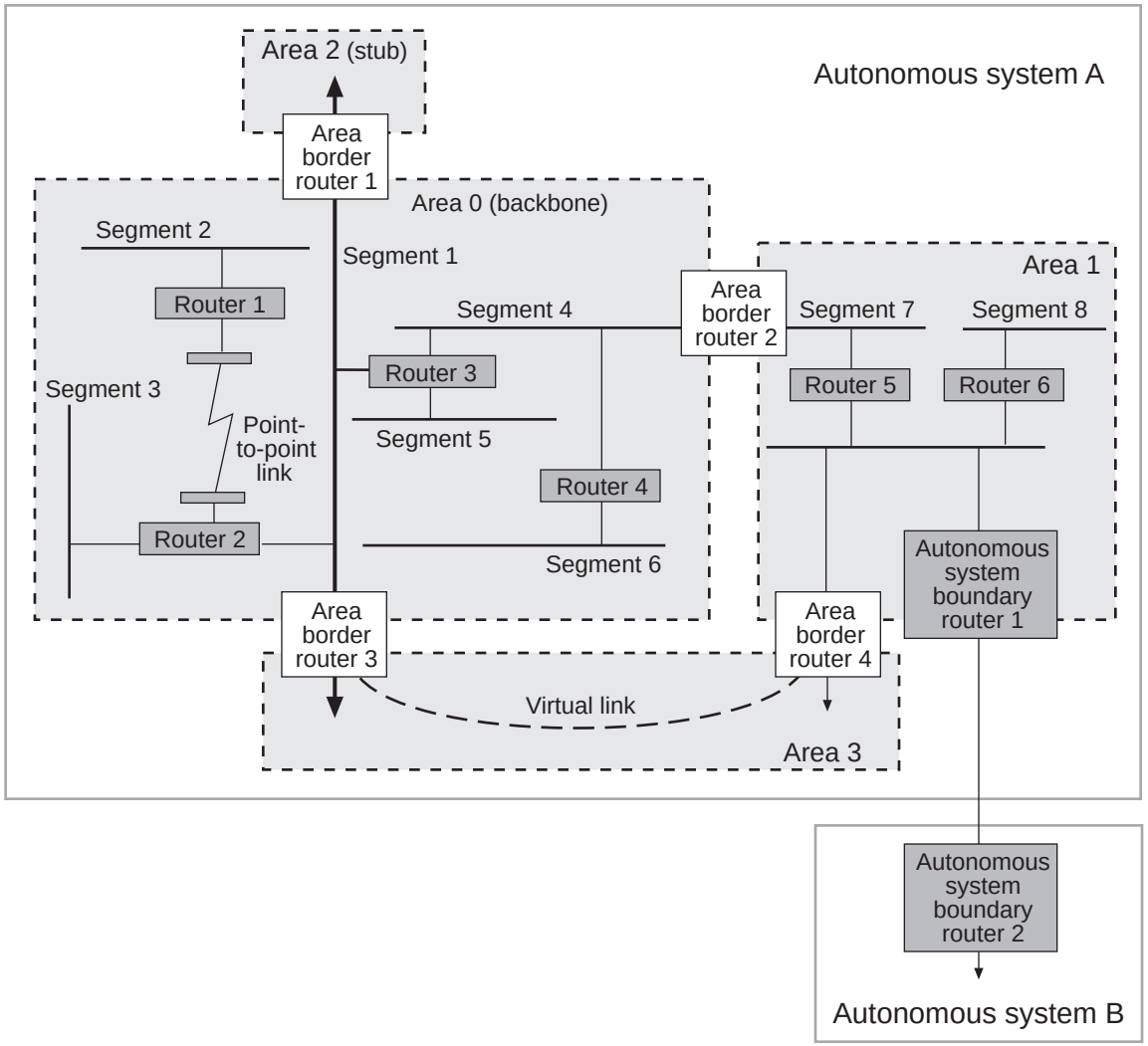
All routers within the same area maintain and use identical link state advertisement (LSA) databases. The network shown in Figure 70 later in this chapter contains four OSPF areas within autonomous system A. There are three types of OSPF areas:

- **Transit area** — An area through which network traffic can pass to reach other areas in the autonomous system. In Figure 70, the backbone area and areas 1 and 3 are transit areas.
- **Backbone area** — A contiguous area within an autonomous system that is divided into more than one area. The system defines the backbone area as 0.0.0.0. The backbone area distributes routing data between other areas in the autonomous system. By definition, the backbone area is also a transit area.
- **Stub area** — Generally, an area with only one entry or exit router. As a result, external routes are never flooded into stub areas. Instead, the area border router that is attached to the stub area advertises a single default external route into the area. This relationship conserves significant LSA database space that would otherwise be used to store external link state advertisements flooded into the area. In Figure 70, area 2 is a stub area that is reached only through area border router 1.

  It is possible to have a stub area with multiple area border routers and multiple exit points. However, all of the exit points and routers must contain the same external routing data so that the choice of an exit point does not need to be made for each external destination.

An area's network topology is visible only to systems inside that area; it is not visible to systems outside that area. Conversely, the systems within an area cannot see detailed network structures outside the area. Because of this restriction of topological information, you can control traffic flow between areas and reduce routing traffic to below the levels that occur when the entire autonomous system is a single routing domain.

Figure 70 Sample OSPF Routing Application



Area Border Routers

Each area (including the backbone area) includes all border routers that are connected to the area. In Figure 70, for example, you define:

- Area border routers 1, 2, and 3 as being in backbone area 0
- Area border routers 2 and 4 as being in area 1
- Area border router 1 as being in area 2
- Area border routers 3 and 4 as being in area 3

Routers must communicate with each other through interfaces that are defined as being in the same area. An area, however, may contain virtual links from area border routers to the backbone area. For example, in Figure 70, area border routers 3 and 4 terminate a virtual link between area 1 and the backbone area. For more information, see "Virtual Links" later in this chapter.

Routing Databases

All routers that are connected to an area maintain identical routing databases about the area. Routers that are connected to multiple areas maintain a separate routing database for each attached area. For example, in Figure 70:

- Routers 1, 2, 3, and 4 maintain identical routing databases about backbone area 0.
- Routers 5 and 6 maintain identical routing databases about area 1.
- Area border router 1 maintains separate routing databases about backbone area 0 and area 2.
- Area border router 2 maintains separate routing databases about backbone area 0 and area 1.
- Area border router 3 maintains separate routing databases about backbone area 0 and area 3.
- Area border router 4 maintains separate routing databases about areas 1 and 3. It also maintains a separate routing database about area 0 due to its virtual link to this backbone area through area border router 3.
- Autonomous system boundary routers 1 and 2 maintain separate routing databases about autonomous systems A and B.

Configuring Route Summarization in ABRs

The concept of route summarization is key in implementing a stable and scalable OSPF internetwork. *Route summarization* is the consolidating of advertised addresses by area border routers (ABRs). Instead of advertising routes to individual nodes within an area, you can configure an ABR to advertise a single summary route or "network range" that covers all the individual networks within its area that fall into the specified range.

Route summarization helps to control routing table sizes and prevents route flapping — that is, the constant changing of routes whenever an interface within an area comes online or goes offline. Using route summarization, routes within an area that change do not need to be changed in backbone ABRs and other area routers.

For optimal route summarization of an area, structure the area with a contiguous addressing scheme so that all routes within the area fall within a specified address range. This summary route or address range is defined by an IP address and mask combination. OSPF supports Variable Length Subnet Masks (VLSMs), so you can summarize a range of addresses on any bit boundary in a network or subnetwork address.

For example, an address range specified with an IP address of 142.194.0.0 with a mask of 255.255.0.0 describes a single route to a collection of destinations between 142.194.0.0 and 142.194.255.255.

Important Considerations

Consider the following guidelines when you design and configure areas:

- Define individual areas as contiguous, that is, any router that participates in OSPF exchanges must have a direct path to all other OSPF routers in the same area.
- Do not disconnect an area border router from the backbone area. This action may result in a loss of network topology information and routing capability.
- Define redundant links between area routers to help prevent area partitioning.
- If a portion of your internetwork consists of a particularly high number of nodes, consider creating an area specifically for that densely connected portion of your network.

- Whenever there is a change in network topology (such as when a link is lost or comes online), routers in all affected areas must converge on the new topology. If your internetwork consists of unstable links, you can partition the AS into smaller areas to minimize the number of areas that affected when the topology changes as a result of those unstable links.

Stub areas

- Because area border routers do not advertise external routes into stub areas, configuring an area as a stub area helps to minimize router table sizes, and therefore the memory requirements, of the routers within the area. Routers within a stub area need to store only a single default external route for destinations outside the autonomous system, as well as for intra-area and inter-area routes.
- If your network has no external routes, there is no advantage to configuring a stub area.
- Stub areas cannot contain autonomous system boundary routers (ASBRs)

Backbone area

A stable, fault-tolerant backbone is vital to your OSPF internetwork. It ensures communication between all areas within the AS. Consider the following guidelines when you design the backbone area:

- If you have only one area in your autonomous system, then you do not need to configure a backbone area (0.0.0.0).
- A backbone area must have the area ID of 0.0.0.0. Routers in the backbone area must be able to communicate with each other through interfaces that are configured in area 0.
- Connect all area border routers to the backbone area with either physical or virtual links.
- Backbones must be contiguous, meaning all area border routers (ABRs) that comprise the backbone must have a direct path to all other ABRs that are attached to the backbone.
- Configure ABRs with high redundancy so that no single link failure can cause a partition in the backbone.
- The more areas that the backbone interconnects, the greater the volume of traffic that must travel over the backbone. To increase stability, keep the size of the backbone reasonable.

- Because all routers connected to the backbone (ABRs) must recompute routes whenever the topology changes for any link in the AS, keeping the size of the backbone to a minimum is especially important in an autonomous system that may contain unstable links. At the very least, reducing the number of areas that connect a backbone directly reduces the likelihood of link-state change.
- Keep the maximum number of routers in the backbone area to about 50 or so, unless the link is of particularly high quality and the number of routes is minimal.
- Every ABR must connect to the backbone; this connection can be physical or virtual. If a router has an OSPF neighbor that is physically connected to the backbone, the router can use that neighbor to establish a virtual link to the backbone. Do not use too many virtual links to connect ABRs for the following reasons:
  - Stability of the virtual link depends on the stability of the underlying area that it spans.
  - This dependency on underlying areas can make troubleshooting difficult.
  - Virtual links cannot run across stub areas.
- Avoid placing hosts, such as workstations, servers, and other shared resources, within the backbone area.
- Having more than one ABR per area reduces the chance that the area will disconnect from the backbone.
- A single ABR can connect one or more areas to the backbone. To maximize stability, a single ABR should support no more than three areas because the router must run the link-state algorithm for each link-state change that occurs for every area to which the router connects.

Default Route Metric

An OSPF router always forwards an IP packet to the network that is the *best match* for the packet's destination; *best match* means the longest or most specific match. A router that fails to find a specific match for a packet's destination forwards the packet to the default router in the area.

To configure an OSPF router to advertise itself as the default router for an area, you define a default route metric. By default, the default route metric is not defined, which means that the router does not advertise itself as the area's default router.

When you remove the default route metric, the router no longer advertises itself as the default router.

OSPF Interfaces

You configure OSPF router interfaces by adding OSPF characteristics to existing IP VLAN interfaces. The OSPF interface has the following characteristics and statistics, which are discussed in the next sections:

- Mode
- Priority
- Area ID
- Cost
- Delay
- Hello Interval
- Retransmit Interval
- Dead Interval
- Password
- Statistics

Mode

The mode for an interface can be *off* or *active*. To run OSPF routing on the interface, set the mode to *active*.

Priority

You assign the interface priority to an OSPF router to determine its status as a designated router. A router can function in one of three ways:

- **Designated router (DR)** — The router that has the highest priority value, unless a designated router already exists on the network segment.
- **Backup designated router (BDR)** — The router that has a lower priority than the DR; the BDR takes over DR functions if the DR fails.
- **Not a designated router** — Any router that is given a priority of 0 or that is not elected DR or BDR. Priority 0 routers can never be elected as a DR or a BDR.

Using Priority to Select a Designated Router

Each OSPF area on a broadcast or nonbroadcast multiaccess network that has at least two attached routers requires a designated router and a backup designated router. The designated router (DR) forms adjacencies to all other routers in the area. If for any reason the DR fails, the backup designated router takes over as the designated router.

To configure a router to be chosen as a designated router, you must understand how the designated router is elected:

- The routing interface that has the highest routing priority within an area is elected as the designated router using the Hello protocol.
- In case of a tie — two or more routers having the same highest routing priority — the router with the highest router ID is chosen as designated router.
- After a designated router is chosen, the same process is used to elect a backup-designated router, with the existing designated router excluded from the election.

Therefore, to designate a router to be elected as the designated router for an area, configure the router with a higher router interface priority than the other routers within the same area. If you want to prevent certain routers within an area from serving as a designated router or backup designated router, configure those routers with a router priority of "0," because interfaces with a router priority of "0" are not eligible for designated router and backup designated router election.

When a router interface within an area first comes online, it determines if a designated router exists for the area. If one exists, the new router accepts the designated router regardless of its own router priority. Therefore, if you want to change the designated router for an area, configure the router that you want to serve as the new designated router to have a higher priority than other routers in the same area. The next time that the election process is initiated, the router that has the highest router priority is elected as the designated router for the area.

Area ID The area ID associates the router interface with an OSPF area. By default, all OSPF interfaces that you create on the system belong to the backbone area (0.0.0.0). If you want to change this association, specify a different area ID for any or all interfaces on the system.

Cost The interface cost parameter reflects the line speed of the port. Although the system calculates a default cost based on the module media type, you can set the cost manually to a different value. In most cases, you can accept the default value that the system sets.

Specifying Cost Metrics for Preferred Paths

In OSPF, the best path is the one that offers the least-cost metric. A cost is associated with each router output interface and each route as follows:

- Each output interface is assigned a default cost by the system based on the media bandwidth to which it is attached.
- Each route is assigned a metric that is equal to the sum of all metrics for all the links in the route.

You can configure area routers to use preferred paths by manually setting higher cost metrics for those paths that are not preferred.

For example, the fastest default media for OSPF is FDDI. All interfaces that are attached to a Fiber Distributed Data Interface (FDDI) media have a default cost metric of 1. All interfaces attached to faster media types, such as Gigabit Ethernet, are also assigned a default cost of 1. To ensure that a particular media interface is the preferred route, leave that link with its metric cost of 1 and manually configure the other links with a higher cost metric — for example, 2.

Delay The transmit delay is the estimated time (in seconds) that it takes for the system to transmit a link state update packet on the interface. The system increases the age of the link state advertisements (LSAs) that are contained in the update packets by the value that you specify for the delay.

This delay setting has more significance for interfaces that are connected to very low speed links because, on slower speed links, it is more probable that the router may send out back-to-back data packets more quickly than other routers and hosts can receive them. To avoid this situation, set the transmit delay to configure the router to wait a specified number or seconds between transmissions.

The delay value that you specify for an interface also increases the age of all LSAs that are transmitted over the interface by the same value. This setting may also affect how soon the LSA is flushed from an area router's database. Reasons that an LSA is flushed from a router's link state database include:

- **LSA is overwritten by a newer instance of the LSA** — For example, when a router receives similar LSAs (LSAs that have identical sequence and checksums), it then compares the ages of each LSA, and stores the LSA that has the least age value in the LSA database. This LSA is then used for routing table calculations.
- **LSA ages out** — When an LSA reaches the maximum age allowed by the system, the router first refloods the LSA onto the network. When it is no longer needed to ensure database synchronization (for example, when the LSA is no longer contained in neighbor LSAs), it is then flushed from the database.

Hello Interval

The Hello interval (in seconds) determines how often the interface transmits Hello packets to other routers. The hello interval value must be identical among all routers that are attached to a common network. *Hello packets* notify other routers that the sending router is still active on the network. If a router does not send Hello packets for the period of time that is specified by the dead interval, that router is considered inactive by its neighbors, and all participating OSPF routers within the affected areas converge on the new topology. Therefore, the smaller the Hello interval, the faster that topological changes are discovered; as a result, however, more routing traffic occurs. The default value for the Hello interval is 10 seconds.

Retransmit Interval

When a router sends a link state advertisement to its neighbor, it keeps a copy of the LSA until the neighbor acknowledges receipt of the LSA with a link state acknowledgment packet. If the sending router does not receive a link state acknowledgment from its neighbor, it then retransmits the LSA. The retransmit interval (in seconds) determines how long the sending router waits for an acknowledgement before retransmitting the LSA to its neighbors.

To prevent needless retransmissions, the value that you specify must be greater than the roundtrip delay between any two routers on the attached network.

Dead Interval The dead interval determines how long neighbor routers wait for a Hello packet before they determine that a neighbor is inactive. Each time that a router receives a Hello packet from a neighbor, the router resets the dead interval timer for that neighbor. The dead interval must be the same for all routers on the network. The default value for the dead interval is 4 times the default value for the Hello interval — 40 seconds.

Password OSPF supports simple password authentication. You can set security passwords for OSPF interfaces so that only routers that know the password participate in OSPF exchanges. Therefore, configure routers in the same area that want to participate in the routing domain with the same password.

When you configure a password on a router interface, the interface inserts the specified password in the OSPF header of every packet that it transmits and receives only those OSPF packets that contain the same password.

Simple password authentication prevents routers from inadvertently joining the area and helps ensure that only trusted routers participate in the routing domain.

By default, OSPF interfaces on your system do not have associated passwords. When no password is assigned to an interface, OSPF exchanges are not authenticated so that, although a password may exist in an OSPF packet header, it is not examined when it is received.

Statistics You can display interface statistics for diagnostic and network debugging purposes. Viewing the statistics for a particular interface can provide valuable information, such as whether the router is overburdened, and the number of Hello interval, dead interval, area ID, and password mismatches that the interface has seen on the network. For a complete listing of OSPF interface statistics, see the `ip ospf interface statistics` command in the *Command Reference Guide*.

Important Considerations

Consider the following guidelines when you configure router interfaces:

- To set the OSPF interface mode to active, enable IP routing.

Designated routers

- Because designated routers and backup designated routers have the most OSPF work to do within an area, select routers that are not already loaded with CPU-intensive activities to be the designated router and backup designated router.
- Because router priority is assigned on a per-interface basis, a single router with interfaces within several different areas can serve as designated router for those areas. But because a designated router has several CPU-intensive responsibilities, it is not a good idea to select the same router as designated router for many areas simultaneously.
- Routers that have an interface priority of 0 cannot serve as a designated router or backup designated router.
- On a broadcast network, if there is no designated router or backup designated router (such as when all routers have a priority of 0), routers do not form neighbor adjacencies, and routing information is not exchanged.

Area ID

- Set the area ID to the same value for all routers on the network segment. All routers in the same area *must* have the same area ID.
- The backbone area 0.0.0.0 is configured by default. The system associates all newly defined OSPF interfaces with the backbone area. You can change this association by changing the area ID for the selected interface.

Transmit delay

- The default value for the transmit delay is 1 second.
- Set the transmit delay to an integer value greater than 0.
- To set the transmit delay, take into account the transmission and propagation delays for the interface.
- Set the transmit delay according to the link speed; use a longer transmit delay for slower link speeds.
- The transmit delay is more effective on very low link speeds.

Hello interval

- The default value for the Hello interval is 10 seconds.
- The smaller the Hello interval, the faster that topological changes are detected, although more routing traffic ensues.
- Set the Hello interval to the same value for all routers on the same network segment.

Dead interval

- The default value for the dead interval is 40 seconds.
- Set the dead interval to 4 times the value specified for the hello timer.
- Set the dead interval to the same value for all routers on the same network segment.

Retransmit interval

- The default value for the retransmit interval is 5 seconds.
- Set the retransmit interval to greater than the expected round trip delay between any two routers on the attached network.
- Set the value that you specify for the retransmit interval conservatively, to avoid needless transmissions.
- Set the retransmit interval higher for serial lines and virtual links.

Password

- By default, an OSPF interface does not have an associated password.
- Use the same password for all routers on the same network segment.
- OSPF passwords are not encrypted. Therefore, a packet analyzer can be used to obtain the password by tapping the wire.
- If no password is defined for an interface, the interface does not verify the existence of a password on reception of a packet.

Link State Databases

OSPF routers use the information that is contained in the link state advertisements (LSAs) to build and maintain link state databases. Each link state database contains the link state advertisements from throughout the areas to which the router is attached. OSPF uses the following types of LSAs:

- Router Link State Advertisements
- Network Link State Advertisements
- Summary Link State Advertisements
- External Link State Advertisements

Router Link State Advertisements

All routers in an OSPF area originate router link state advertisements, also known as link state advertisements. Each link state advertisement describes the state and cost of the originating router's links (interfaces) to the area. Information contained in each link state advertisement includes:

- **LSID (Link State ID)** — The ID of the router that generated the LSA.
- **Router ID** — ID of the router that originated the LSA.
- **LS Seq (Link State Sequence) —** The sequence number of the advertisement. Used to detect old or duplicate link state advertisements.
- **LS age** — The time, in seconds, since the LSA was generated.
- **Flags** — Possible values:
  - **V** — Router is the endpoint of an active virtual link that is using the area as a transit area.
  - **ASBR** — Router is an autonomous system boundary router (ASBR).
  - **ABR** — Router is an area border router (ABR).
- **Link Type** — A description of the router link. Possible values:
  - **PTP** — Connection is point-to-point to another router.
  - **Transit** — Connection is to a transit network.
  - **Stub** — Connection to a stub network.
  - **Virtual link** — Connection is to a far-end router that is the endpoint of a virtual link.

- **Link ID** — Identifies the object to which this router link connects for each Link Type. Possible values:
  - **If Link Type is PTP**, then this is the neighboring router's router ID.
  - **If Link Type is Transit**, then this is the address of the designated router.
  - **If Link Type is Stub**, then this is the IP network or subnetwork number.
  - **If Link Type is Virtual Link**, then this is the neighboring router's router ID.
- **Link Data** — Provides additional link information. Possible values:
  - **If Link Type is PTP**, then this is the MIB II index value for an unnumbered point-to-point interface.
  - **If Link Type is Transit**, then this is the IP address of the advertising router's interface.
  - **If Link Type is Stub**, then this is the network's IP address mask.
  - **If Link Type is Virtual Link**, then this is the IP address mask of the neighboring router.
- **Metric** — Cost of using this outbound router link. With the exception of stub networks, this value must be other than 0.

Network Link State Advertisements

The designated router for each area originates a network link state advertisement for each transit network — a network that has more than one attached router. This advertisement describes all routers that are attached to the network, including the designated router itself. Each network link state advertisement (LSA) includes this information:

- **LSID (Link State ID)** — The ID of the router that generated the LSA.
- **Router ID** — ID of the router that originated the LSA.
- **LS Seq (Link State Sequence)** — The sequence number of the advertisement. Used to detect old or duplicate link state advertisements.
- **LS age** — The time, in seconds, since the LSA was generated.
- **Network Mask** — IP address mask for the network.
- **Attached Routers** — List of routers that are fully adjacent to the designated router (DR). The ID of the DR is also listed here.

Summary Link State Advertisements

Area border routers can generate two types of summary link state advertisements:

- Summary link state advertisements that report the cost to a single subnetwork number outside the area. These advertisements are identified as Type 3 in the link state advertisement header.
- Summary link state advertisements that report the cost to a single autonomous system boundary router (ASBR). These advertisements are identified as Type 4 in the link state advertisement header.

Each summary link state advertisement includes this information:

- **LSID (Link State ID)** — Possible values:
  - **For Type 3 summary link advertisements**, this is the IP network number.
  - **For Type 4 summary link advertisements**, this is the ASBR's router ID.
- **Router ID** — ID of the router that originated the LSA.
- **LS Seq (Link State Sequence)** — The sequence number of the advertisement. Used to detect old or duplicate link state advertisements.
- **LS age** — The time, in seconds, since the LSA was generated.
- **Network Mask** — For Type 3 summary link state advertisements, this is the destination network's IP address mask. For Type 4 summary link state advertisements, this parameter is set to 0.
- **Metric** — The cost of the specified route.

External Link State Advertisements

Each autonomous system boundary router generates an external link state advertisement for each network destination (known to the router) outside the AS. AS boundary routers use these external link state advertisements to describe routes to destinations outside the AS. For these advertisements, the Link State ID field in the advertisement header specifies an IP address.

In addition, OSPF also considers the following routes to be external routes. They are advertised using external link state advertisements:

- The default route
- Static routes
- Routes derived from other routing protocols, such as RIP
- Directly connected networks that are not running OSPF

All external routes are assigned a cost metric. External route cost is calculated based on one of these cost metric types:

- **Type 1** — Router adds the internal cost metric to the external route metric. For example, if an ABR is advertising Type 1 external route metrics, the cost of the route from any router within the AS is equal to the cost associated with reaching the advertising ABR, plus the cost of the external route.
- **Type 2** — Routers do not add the internal route metric to the external route metric. Therefore, the router that advertises the smallest external metric is chosen, regardless of the internal distance to the AS boundary router. For example, if an ABR is advertising Type 2 external route metrics, the cost of the route from any router within the AS is equal to the cost of the external route alone. The cost of reaching the advertising ABR is not considered in determining the cost of the external route. The internal cost is only used as a *tie-breaker* when several equal-cost Type 2 routes exist.

When both Type 1 and Type 2 metrics are present in an AS, Type 1 external metrics take precedence.

Each external link state advertisement includes this information:

- **LSID (Link State ID)** — An IP network address:
  - **For Type 3 summary link advertisements**, this is the IP network number.
  - **For Type 4 summary link advertisements**, this is the ASBR's router ID.
- **Router ID** — ID of the router that originated the LSA.
- **LS Seq (Link State Sequence)** — The sequence number of the advertisement. Used to detect old or duplicate link state advertisements.
- **LS age** — The time, in seconds, since the LSA was generated.

- **Network Mask** — The IP address mask for the advertised destination.
- **Fwd address (Forwarding Address)** — If the AS boundary router is advertising a destination that can be more optimally reached by a different router on the same LAN, then the advertising boundary router specifies that router's address in the forwarding address field. Otherwise, it leaves the field as 0.
- **Metric** — The cost to reach the advertised destination.
- **Type** — Possible values:
  - **Type 1** — Normal link state metric.
  - **Type 2** — The metric is larger than any local link state path. See the discussion of Type 1 and Type 2 external metrics earlier in this section.
- **Route Tag (External)** — Not used by OSPF. These 32-bits may be used to communicate other information between boundary routers.

Important Considerations

When you view the link state database, consider the following:

- An asterisk (*) after the router ID in a display indicates that the LSA originated locally.
- All routers within an area must maintain identical link state databases.
- Use the contents of the link state database for network configuration and troubleshooting purposes.

Neighbors

Neighbor routers are those that are physically attached to the same network segment. The OSPF Hello protocol establishes adjacencies among neighboring routers to facilitate the exchange of routing information. An *adjacency* describes the relationship between two routers that exchange network topology information. A router attached to multiple network segments may have different sets of neighbors on each segment.

For example, Figure 70 earlier in this chapter includes several sets of OSPF neighbor routers. In backbone area 0:

- Routers 2 and 3 and area border routers 1 and 3 are neighbors on segment 1 (the backbone network).
- Routers 1 and 2 are neighbors on a point-to-point link.
- Routers 3 and 4 and area border router 2 are neighbors on segment 4.
- No routers are neighbors on segments 2, 3, 5, and 6.

In area 1:

- Router 5 and area border router 2 are neighbors on segment 7.
- Routers 5 and 6, area border router 4, and autonomous system boundary router 1 are neighbors on segment 9.
- No routers are neighbors on segment 8.

In area 3, area border routers 3 and 4 are neighbors on a virtual link between the backbone area 0 and area 1.

Neighbor Information

Your system can display a list of all neighbors for all OSPF interfaces defined on the system. The list includes the following information:

- **Index** — The Index number that corresponds to the OSPF router interface for which neighbors have been discovered.
- **Neighbor Address** — The IP address of the neighboring router.
- **Router ID** — The router ID of the neighboring router.

- **State** — The state of the adjacency. You can also think of this as the state of the conversation that is held with the neighboring router. Possible neighbor state values:
  - **Down** — The initial state of a neighbor conversation. It indicates that no recent information has been received from this neighbor.
  - **Attempt** — Only used on nonbroadcast networks. This value indicates that no recent information has been received from this neighbor, but the router tries to contact the neighbor by sending Hello packets.
  - **Init** — A Hello packet has recently been seen by a neighbor, but two-way communication has not been established.
  - **Two-way** — Bidirectional communication has been established. Two-way is the most advanced state of a neighbor relationship before beginning to establish an adjacency. In fact, the designated router and backup designated router are selected from the set of neighbors that are in a state of two-way communication or greater.
  - **Exstart** — The initial step in creating an adjacency between two routers. Adjacencies involve a master/slave relationship between two routers, which is when that relationship is determined. The master sends the first information describing its link state database in the form of database description packets. The slave can only respond to the database description packets.
  - **Exchange** — The router is describing its link state database by sending database description packets to the neighbor. All adjacencies in the exchange state are used by the flooding procedure. Adjacencies in this state are capable of transmitting and receiving all types of OSPF protocol packets.
  - **Loading** — The router is sending requests for link state advertisements (LSAs) that were discovered in the exchange state but not yet received.
  - **Full** — The neighbor is now fully adjacent. This adjacency is now advertised in router LSAs and network LSAs.
- **Priority** — The priority of this neighbor in terms of designated router election. A value of 0 indicates that the neighbor is not eligible to become a designated router.

- **RxQ (Retransmit Queue)** — The number of LSAs in the local retransmit queue to the neighbor. These LSAs have been flooded but not acknowledged on this adjacency. The LSAs in the queue are flooded until they are acknowledged by the neighbor or until the adjacency is destroyed.
- **SumQ (Summary Queue)** — The number of LSAs that make up the area link state database at the moment that the neighbor goes into the database exchange state. These LSAs are sent to the neighbor in database description packets.
- **ReqQ (Request Queue)** — The number of LSAs that are required from the neighbor in order to synchronize the neighboring routers' link state databases. The router requests these LSAs by sending link state request packets to the neighbor. The neighbor then responds with link state update packets containing the requested LSAs. As the appropriate LSAs are received from the neighbor, they are removed from the request queue.
- **Flags** — The type of neighbor. Possible values:
  - **D** — The neighbor was dynamically discovered.
  - **S** — The neighbor was statically defined.
  - **BDR** — The neighbor is the backup designated router for the area.
  - **DR** — The neighbor is the designated router for the area.
- **Examples**
  - `S+BDR` indicates that the neighbor was statically defined and serves as the backup designated router for the area.
  - `D+DR` indicates that the neighbor was dynamically discovered and serves as the designated router for the area.

Static Neighbors

On broadcast networks such as Ethernet, the OSPF Hello protocol uses the broadcast capability to dynamically discover neighbors. On nonbroadcast networks, such as X.25 Public Data Network, however, you may need to assist in neighbor discovery by statically defining neighbors on each interface. OSPF then uses the Hello protocol to maintain the neighbors that you statically define.

When you statically define a neighbor on the system, you specify both the router interface to which you want to add the neighbor and the IP address of the neighboring router that you want to associate with the specified interface. The Hello protocol then dynamically retrieves the additional neighbor information, as described in "Neighbor Information" in the previous section.

Important Considerations

Consider the following guidelines when you configure neighbors:

- Routers use OSPF hello packets to learn neighbor addresses dynamically on broadcast networks.
- Define static neighbors only on nonbroadcast interfaces, because neighbors are not learned dynamically on nonbroadcast networks.
- Hello packets are the only OSPF packet type that is processed by all routers within an area. All other packet types are sent and received only on adjacencies.
- Neighbor adjacencies cannot be formed if two routers have different Hello intervals, Dead intervals, or passwords.

Router IDs

Each router that is configured for OSPF has an OSPF router ID. The OSPF router ID uniquely identifies the router to other routers within an autonomous system.

The router ID determines the designated router in a broadcast network if the priority values of the routers involved in the designated router election are equal. In the event of a *priority* tie, the router with the highest router ID is elected designated router for the area.

Three types of router identifiers, in the form of an IP address, are available:

- **Default** — A unique ID that the system generates and uses as the default router ID.
- **Interface** — The index of an IP interface on the router.
- **Address** — An ID that you define in the form of an IP address.

OSPF routing must be inactive before you can add or modify an OSPF router ID. To deactivate OSPF routing, set the OSPF mode to `disabled`. See the *Command Reference Guide* for details. After you add the router ID, you can set the OSPF mode to `enabled` on the interface.

Important Considerations

Consider the following guidelines when you configure OSPF router IDs:

- For OSPF to operate correctly, the router ID must be unique for every router.
- Choose the default setting to ensure unique router IDs.
- You cannot set the router ID to either 0.0.0.0 or 255.255.255.255.

OSPF Memory Partition

There are three choices for OSPF memory allocation:

- Have the system intelligently determine the maximum OSPF memory partition size (partition size = `1`). This is the default.
- Have OSPF be part of system memory, growing as needed and without limit (partition size = `0`).
- Configure the maximum OSPF memory partition size manually (partition size = `4096` - `<maximum available memory>`).

You use the `ip ospf partition modify` option to control memory allocation, as described in the *Command Reference Guide*.

Default Memory Allocation

You typically do not have to modify the OSPF memory allocation. By default, the system manages memory by partitioning the total memory available for applications among the various protocols. This functionality ensures that the router has enough memory for to perform all of its functions and enable most features. Under this option, OSPF always has a partition of memory available for its use.

Under the default OSPF memory allocation scheme, two values have meaning:

- Current partition maximum size
- Allocated memory size

Current Partition Maximum Size

The *current partition maximum size* is the maximum amount of memory that OSPF can allocate. It is calculated at system startup as a function of the maximum routing table size and available memory by the following formula:

(((externalLSAsize * maxRoutingTableSize + 100000) / 100000) + 1) * 100000

Because most of the routes are going to be external to OSPF, the formula bases the OSPF memory partition maximum size on the amount of memory that is required to store an external link state advertisement (LSA), 80 bytes, times an estimate of the maximum number of routing table entries that the system can hold (maxRoutingTableSize). It then rounds to the nearest 100000 bytes and adds an additional 100000 bytes as a buffer.

The estimate (maxRoutingTableSize) of the maximum number of routing table entries the system can hold for a given memory size is a hardcoded value. On extended memory systems this value is 51200. On systems without extended memory this value is only 1024.

Applying the formula to extended memory systems yields a default OSPF current partition maximum size of 4,200,000. (Due to memory overhead, the actual number of routing table entries possible is somewhat different than the 51200 maximum.)

Even though currently unallocated, this memory is not available to other protocols.

Allocated Memory Size

The *allocated memory size* is the size of the memory that is currently allocated to OSPF. The minimum size this allocated memory partition can default to is 100000.

The system allocates more memory as required in 100000-byte chunks until the current partition maximum size is reached.

Running Out of Memory — Soft Restarts

An attempt to allocate memory past the OSPF current partition maximum size generates a soft restart condition that momentarily causes the router to go down. This may occur, for example, because:

- The routing table grew suddenly because it received a large number of external link state advertisements (LSAs), such as RIP routes learned from an ASBR, that had to be added to the internal database.
- The router is an area border router (ABR) for multiple large subareas and thus has a much larger than usual routing table.

The `ip ospf statistics` *option displays the number of soft restarts.*

After the soft restart, the system frees all of its OSPF memory, disables its interfaces, reenables them, and reconstructs the router tables from scratch. This process attempts to free and defragment enough unused memory so that OSPF has sufficient memory to continue. If the soft restart does not free enough memory, the soft restart condition repeats — and the router continues to thrash for memory.

If the softRestarts statistic shows that the default memory allocation scheme is too small for your router, then you must use one of the other two memory allocation options described next.

Manual Memory Allocation

You can manually control the OSPF current partition maximum size. You can enter any value between 4096 and the maximum memory available on your system, as shown in the `ip ospf partition modify` command prompt.

You can also use manual memory allocation control to *lower* the OSPF current maximum partition size to be less than the 4,200,000 default minimum on extended memory systems. As noted previously, memory reserved under the OSPF current maximum partition size is not available to other protocols even if it is not allocated. If you must carefully apportion memory among competing protocols, then you might want to decrease the memory available to OSPF. A router located in a stub area has no external link state advertisements (LSAs), for example, and might require less memory.

System Memory Allocation

You can also have OSPF use the *system* memory partition. There will be no specific OSPF memory partition and no current maximum partition size. OSPF will grow as it finds necessary, possibly encroaching upon the space available to other protocols.

Stub Default Metrics

Generally, a stub area is a network that is connected to an OSPF routing domain by a single area border router (ABR). External link state advertisements are not advertised into stub areas. Instead, the ABR injects a Type 3 summary link state advertisement that contains a single external default route into the stub area. The routers within the stub area use this single external route to reach all destinations outside the stub area. This arrangement saves routing table space and system resources because stub area routers do not have to learn a multitude of external routes for the greater network; they need only store a single external route.

The stub default metric determines whether an ABR generates the default route into the stub area to which it is connected, and the cost associated with that route.

For example, in Figure 70 earlier in this chapter, you would configure area border router 1 to generate a default route into stub area 2. If you define a stub default metric of 4, area border router 1 will generate a default route with an associated cost of 4 into stub area 2.

If you remove the stub default metric, the ABR does not advertise a default route into the stub area.

A stub area can have multiple ABRs and multiple exit points. However, all of the exit points and routers must contain the same external routing data so that the choice of an exit point does not need to be made for each external destination.

Important Considerations

Consider the following guidelines when you define stub default metrics:

- By default, area border routers advertise a stub default metric of 1.
- Stub default metrics are relevant only for area border routers (ABRs) that are attached to stub networks.
- If your network does not have external routes, you do not need to configure the stub default metric; and you do not need a stub area.
- If you remove the stub default metric, the ABR does not advertise a default route into the stub area.

Virtual Links

The backbone area (0.0.0.0) must link to all areas. If any areas are disconnected from the backbone, some areas of the autonomous system (AS) become unreachable. In the rare case that it is impossible to physically connect an area to the backbone, you can use a virtual link. The virtual link provides a logical path to the backbone for the disconnected area.

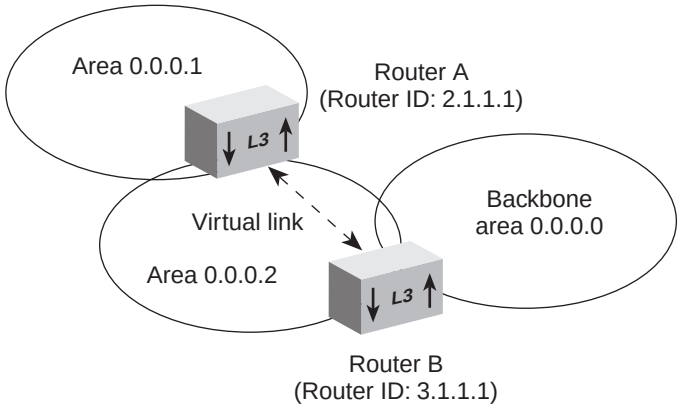
Virtual links are used to ensure that the OSPF backbone is contiguous. You can use virtual links to:

- Introduce new areas that do not have physical access to the backbone.
- Add redundancy to the backbone to help prevent partitioning.
- Patch the backbone when discontinuity occurs.
- Connect area backbones. For example, you can merge two existing OSPF networks into one network sharing a common backbone.

A virtual link must be established between two ABRs that share a common nonbackbone area, with one of those ABRs directly connected to the backbone. The nonbackbone area through which the virtual link runs is called a *transit area*.

The endpoints of a virtual link must be area border routers. You must configure the virtual link on both routers. Each router's virtual link definition includes the other router's router ID and the transit area through which the routers connect. Figure 71 illustrates a virtual link between two area border routers.

Figure 71 Virtual Link



In Figure 71, area 0.0.0.1 cannot be physically connected to the backbone area. Instead, connectivity to the backbone is achieved using a virtual link, configured between router A and router B. Area 0.0.0.2 is the transit area, and router B is the entry point into backbone area 0.0.0.0. The virtual link in Figure 71 provides area 0.0.0.1 and router A with a logical connection to the backbone. Here is the virtual link configuration for both routers shown in Figure 71:

- Router A:
  - Transit area: 0.0.0.2
  - Target router: 3.1.1.1
- Router B:
  - Transit area: 0.0.0.2
  - Target router: 2.1.1.1

Important Considerations

Consider the following guidelines when you configure virtual links:

- You must configure a virtual link for any area border router that has an interface connected to a location outside the backbone area.
- You can define up to 32 virtual links per router.
- You cannot configure a virtual link through a stub area.
- Use virtual links sparingly for the following reasons:
  - Stability of the virtual link depends on the stability of the underlying area that it spans.
  - This dependency on underlying areas can make troubleshooting difficult.

OSPF Routing Policies

Routing policies are rules that define criteria to control the flow of routes to and from the routing table. Your system supports two types of OSPF routing policies: *import* policies that dictate which routes are added to the routing table and *export* polices that dictate which routes are advertised to other routers. You can use routing policies to:

- **Increase security** — For security reasons, you may not want the router to advertise certain routes. For example, Organization A may have defined one of its ASBRs with a direct connection to Organization B that they use for direct communication. For security or performance reasons, A may not want to give other groups access to that connection. To prevent this direct connection from being known to other organizations, A can define an export policy that prohibits its ASBR from advertising the direct connection that it uses to communicate with B.
- **Conserve routing table space** — The selective nature of routing policies can minimize routing table sizes and increase network stability. For example, you may want to limit the number of hosts and gateways from which routing information is accepted, in which case you can define an import policy to selectively rule out, or reject, unnecessary routing table entries.
- **Isolate suspect networks** — Misconfigured hosts can sometimes send inappropriate routing information, which can compromise network integrity. In such a case, you can define an import policy on an ASBR that rejects all routes from the suspect network.
- **Adjust route cost** — Both import and export policies let you change the cost that is associated with routes without physically changing the cost of an interface. For example, router A may advertise a route with one cost, but router B may use an import policy to write the same route to its routing table with a different, or adjusted, cost. Similarly, router A may have a route in its routing table with one cost but choose to advertise the route to other routers with a different cost.

Important Considerations

Consider the following guidelines when you work with OSPF routing policies:

- You can only apply OSPF policies against external routes. External routes refer to routes that are advertised over the network using external link state advertisements (LSAs). These routes include:
  - **Directly connected non-OSPF interfaces** — Physical interfaces on the router itself that are not running OSPF and that are directly connected to the network.
  - **RIP routes** — Routes to destinations outside the autonomous system learned via the Routing Information Protocol (RIP) and imported by autonomous system boundary routers.
  - **Static routes** — IP routes statically defined by the user.
- You cannot apply export policies against directly connected OSPF interfaces, because all routers in the area must maintain identical link state databases.
- With the ability to wildcard policy parameters (such as 0.0.0.0 to indicate all routers or all routes), occasions may arise when several policies match a route. In such cases, routers use the following procedure to determine which policy to apply to the route:
  - If multiple policies apply to the route, the router uses the policy that has the highest administrative weight.
  - If multiple matches still exist, the router uses the policy that matches the specified source (excluding wildcards).
  - If multiple matches still exist, the router compares route address bits and uses the policy that best matches the route address (excluding wildcards).
  - If multiple matches still exist, the router uses the policy that matches the origin protocol.
  - If multiple matches still exist, then the router uses the policy that has the lowest index number.

- You can set up an IP RIP or OSPF import or export policy to accept or advertise the default route, as long as the default route exists in the routing table. When you define a policy, you are always prompted for the route subnet mask after the route address, even though you specify the wildcard route address of 0.0.0.0.

  Specify a route subnet mask as follows:

  - If you want the wildcard subnet mask for all routes, enter 0.0.0.0 for the subnet mask. This is the default subnet mask.
  - If you want the default route (not all routes), enter 255.255.255.255.

- For more information about IP routing policies, see Chapter 11.

Implementing Import Policies

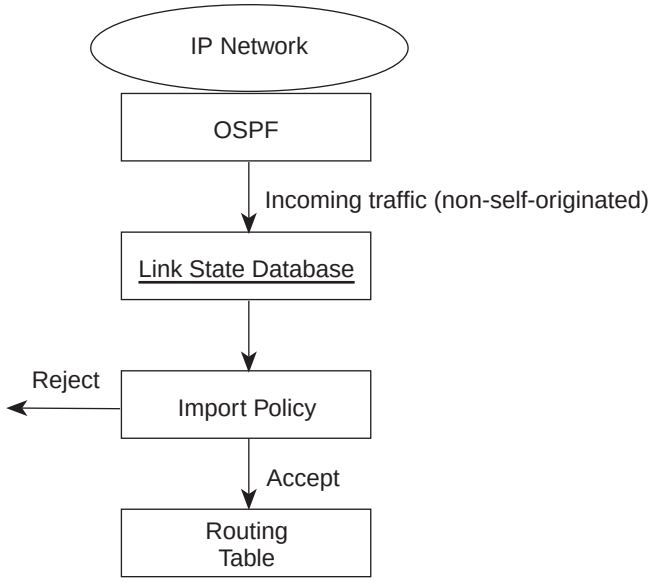
Import policies control which non-self-originated (external routes being RIP) are accepted and stored in the routing table. *Non-self-originated* means that the router itself did not originate the route; it learned it by means of an external link state advertisement. You can also adjust the cost of each route that is accepted into the routing table. Using RIP, you can define which *external routes* (RIP) a router advertises. They are self-originated but must be external to OSPF.

Because all routers within the same OSPF area must maintain similar databases, all routers must receive all link state advertisements that are sent over the network, and store those advertisements in their link state databases. By defining an import policy, however, you can control what routes from a router's external link state database are migrated to its routing table.

When you define an import policy, you specify a set of criteria. When a the router receives an external link state advertisement, it consults its routing policies to determine if the route specified in that advertisement matches any of the defined policies criteria, and, if so, applies to the route the actions that are defined by the policy.

Figure 72 illustrates the import policy process.

Figure 72 Import Policy Process



Information that you define for an import policy includes:

- The route or routes to which you want the policy to apply, specified by a network address and subnet mask.
- The action that you want the router to take — accept or reject. `Accept` configures the router to add the route to its routing table. `Reject` prevents the router from adding the route to its routing table.

- For routes that are accepted into the routing table as defined by the policy, you can define a new cost metric value for the route, or you can adjust the existing cost metric using one of these operators:
  - + adds the specified number to the existing cost metric
  - - subtracts the specified number from the existing cost metric
  - * multiplies the specified number by the existing cost metric
  - / divides the existing cost metric by the specified number
  - % modulo divides the existing cost metric by the specified number and returns the remainder

  The routes are then accepted into the routing table with the cost metric that has been defined by the import policy.
- In case multiple policies match the same route, you can also assign an administrative weight to define an order of precedence.

Import Policies at a Glance

Table 47 lists the possible import policy configurations.

Table 47 OSPF Import Policies

Route Address	Route Subnet Mask	Policy Action	Metric Adjustment	Description
0.0.0.0	N/A	Accept	C	Adds all external routes to routing table and assigns a cost of C to each
A	Subnet mask for route A	Accept	C	Adds Type 1 and Type 2 Route A to the routing table with an associated cost of C
0.0.0.0	N/A	Reject	N/A	Does not add any external routes to the routing table
A	Subnet mask for route A	Reject	N/A	Does not add Type 1 or Type 2 Route A to the routing table

Import Example 1: Accept Route

The policy defined in Table 48 imports route 243.140.28.0 into the routing table and assigns a cost of 10 to the route.

Table 48 Import Policy Example

Policy Field	Definition
Policy type	import
Route address	243.140.28.0
Route subnet mask	255.255.255.0
Policy action	accept
Metric adjustment	10
Administrative weight	16

Import Example 2: Reject Route

The policy defined in Table 49 prohibits the router from adding route 243.140.28.0 to its routing table.

Table 49 Export Policy Example

Policy Field	Definition
Policy type	import
Route address	243.140.28.0
Route subnet mask	255.255.255.0
Policy action	reject
Administrative weight	15

Implementing Export Policies

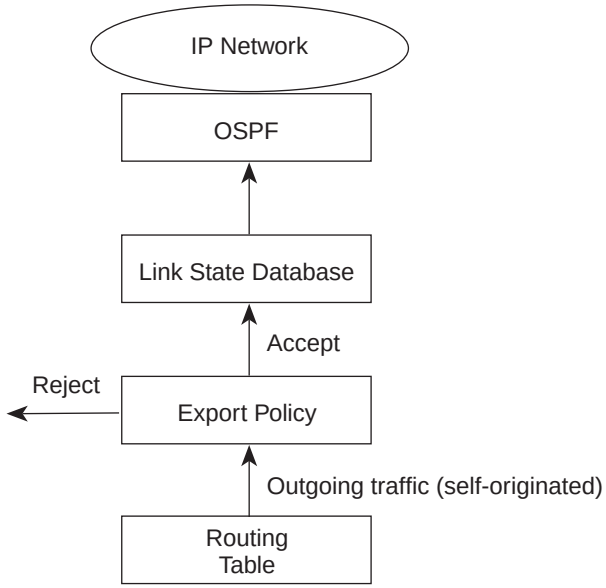
Using export policies, you can define which self-originated (external routes being RIP) a router advertises. *Self-originated* refers to routes that are originated by the router itself. You can also adjust the cost and external metric type of each route that you allow the router to advertise. Using RIP, you can define which *external routes* (RIP) a router advertises. External routes are self-originated, but must be external to OSPF.

See the discussion about Type 1 and Type 2 metrics in "External Link State Advertisements" *earlier in this chapter for more information about external metric types.*

When you define an export policy, you can configure the router to accept or reject routes. An *accept* export policy configures the router to place the specified route in external link state advertisements for propagation over the network. The routes are advertised with the cost and the external metric type defined by the policy. A *reject* export policy prevents the router from placing the specified route in external link state advertisements, thereby prohibiting propagation of the route over the network.

Figure 73 illustrates the export policy process.

Figure 73 Export Policy Process



You define these criteria as part of an export policy:

- The method by which the route was learned by the router. Possible origins include directly connected interfaces and static routes, as well as RIP routes imported by autonomous system boundary routers.
- When you define an export policy against a directly connected interface, you can specify one or all of the physical router interfaces that are directly connected to the network against which you want the export policy to be applied. For example, if you define an export policy that rejects a direct interface, the router does not advertise the specified interface over the network.

- When you specify RIP or static as the origin protocol, you can specify the source address of the router that originated the RIP or static route. For example, you can define an export policy to reject (that is, not advertise) all statically defined routes, in which case you specify the local router's ID as the source address.
- The route or routes to which you want the policy to apply, specified by a network address and subnet mask.
- The action that you want the router to take:
  - **Accept** — The specified route is placed in external link state advertisements and propagated over the network.
  - **Reject** — The specified route is not placed in external link state advertisements and as a result is not propagated over the network.
- For export policies that define routes to be advertised in external LSAs, you can define a new cost metric value for the route, or you can adjust the existing cost metric using one of these operators:
  - + adds the specified number to the existing cost metric
  - - subtracts the specified number from the existing cost metric
  - * multiplies the specified number by the existing cost metric
  - / divides the existing cost metric by the specified number
  - % modulo divides the existing cost metric by the specified number and returns the remainder

  The routes are then advertised with the cost metric as defined by the export policy.
- You can choose to advertise the route as a Type 1 or a Type 2 external cost metric.
- In case multiple policies match the same route, you can also assign an administrative weight to define an order of precedence.

Export Policies for RIP and Static Routes

Table 50 shows the export policies that can be applied to RIP and statically defined routes.

Table 50 OSPF Export Policies for RIP and Static Routes

Origin Protocol	Source Router	Route	Policy Action	Metric Adjustment	External Metric Type	Description
RIP or Static	A	B	Accept	C	Type 1, Type 2	RIP or Static Route B originating from Router A is advertised as the specified metric type with a cost of C.
RIP or Static	A	0.0.0.0	Accept	C	Type 1, Type 2	All RIP or Static routes originating from Router A are advertised as the specified metric type with a cost of C.
RIP or Static	0.0.0.0	B	Accept	C	Type 1, Type 2	RIP or Static Route B originating from any router is advertised as the specified metric type with a cost of C.
RIP or Static	0.0.0.0	0.0.0.0	Accept	C	Type 1, Type 2	RIP or Static routes originating from any router are advertised as the specified metric type with a cost of C.
RIP or Static	A	B	Reject	N/A	N/A	RIP or Static Route B originating from Router A is not advertised.
RIP or Static	A	0.0.0.0	Reject	N/A	N/A	All RIP or Static routes originating from Router A are not advertised.
RIP or Static	0.0.0.0	B	Reject	N/A	N/A	RIP or Static Route B originating from any router is not advertised.
RIP or Static	0.0.0.0	0.0.0.0	Reject	N/A	N/A	RIP or Static routes originating from any router are not advertised.

Export Policies for Direct Interfaces

Table 51 shows the possible export policies that can be applied to directly connected router interfaces.

Table 51 OSPF Export Policies for Directly Connected Interfaces

Origin Protocol	Interface	Policy Action	Metric Adjustment	External Metric Type	Description
Direct	Specific non-OSPF interface or All non-OSPF interfaces	Accept	C	Type 1, Type 2	The specified interfaces are advertised as a Type 1 or Type 2 metric with a cost of C.
Direct	Specific non-OSPF interface or All non-OSPF interfaces	Reject	N/A	N/A	Do not advertise the specified interfaces.

Export Example 1: Prohibit Advertisement of non-OSPF Interfaces

The policy defined in Table 52 prohibits an autonomous system boundary router from advertising any directly connected non-OSPF interfaces.

Table 52 Export Policy to Reject Direct Interfaces

Policy Field	Definition
Policy type	export
Origin Protocol	dir
IP interfaces	all
Policy action	reject
Administrative weight	1

The router prohibits the address of any of its directly connected RIP interfaces from being placed in external link advertisements. As a result, the interfaces are not advertised over the network.

Because all OSPF routers must maintain similar link state databases and shortest path trees, you cannot define an export policy to restrict the advertisement of directly connected OSPF interfaces.

Export Example 2: Prohibit Advertisement of Static Address

The policy defined in Table 53 prohibits a router from advertising any static route originating from router 131.141.127.7.

Table 53 Export Policy to Reject Static Routes

Policy Field	Definition
Policy type	export
Origin protocol	sta
Source address	131.141.127.7
Route address	0.0.0.0
Policy action	reject
Administrative weight	1

Although the router can learn all static routes that originate from router 131.141.127.7, this policy prohibits any of those routes from being placed in external link advertisements.

Export Example 3: Prohibit Advertisement of RIP Routes

The policy defined in Table 54 prohibits an autonomous system boundary router from advertising imported RIP route 138.140.9.0 originates from router 131.141.126.9.

Table 54 Export Policy to Reject RIP Routes

Policy Field	Definition
Policy type	export
Origin protocol	rip
Source address	131.141.126.9
Route address	138.140.9.0
Route subnet mask	255.255.255.0
Policy action	reject
Administrative weight	1

Although the router can add the 138.140.9.0 route to its routing table, this policy prohibits the boundary router from migrating the route from its routing table to its link state database. As a result, the route is not propagated over the network.

Export Example 4: Advertisement of Direct Interfaces

The policy defined in Table 55 configures a router to advertise direct interface 8 as a Type 2 external metric with a cost increase of 2.

Table 55 Export Policy to Accept a Direct Interface

Policy Field	Definition
Policy type	export
Origin protocol	dir
IP interfaces	8
Policy action	accept
Metric adjustment	+2
ASE Type	Type 2
Administrative weight	1

Suppose a routing table entry exists for interface 8 that identifies the route as a Type 1 external metric with an associated cost of 10. This policy configures the router to export direct interface 8 from its routing table and write the routing interface information to its link state database as a Type 2 external metric, with an associated cost of plus 2. As a result the router advertises the interface over the network as a Type 2 external metric with an associated cost of 12, overriding the external metric type and cost that are defined for the interface in the system's routing table.

Export Example 5: Advertisement of Static Routes

The policy defined in Table 56 configures a router to advertise all static routes as Type 1 external metrics with a cost of 1.

Table 56 Export Policy to Accept Static Routes

Policy Field	Definition
Policy type	export
Origin protocol	sta
Source address	0.0.0.0
Route address	0.0.0.0
Policy action	accept
Metric adjustment	1
ASE Type	Type 1
Administrative weight	1

Export Example 6: Advertisement of RIP Routes

The policy defined in Table 57 configures an autonomous system boundary router to advertise all routes that are imported from a RIP network as Type 2 external metrics with associated costs of 10.

Table 57 Export Policy to Accept RIP Routes

Policy Field	Definition
Policy type	export
Origin protocol	rip
Source address	0.0.0.0
Route address	0.0.0.0
Policy action	accept
Metric adjustment	10
ASE Type	Type 2
Administrative weight	1

OSPF Statistics

From the Administration Console and the Web Management interface, you can display general statistics for specific OSPF interfaces. These statistics provide valuable information useful in troubleshooting network and system issues. For example, the number of SPF computations directly corresponds to the number of topological changes that the interface had to converge on. An excessive number of soft restarts may be an indication that the router is overburdened because of resource limitations.

The following is a list of OSPF statistics:

- **SPF computations** — Number of shortest-path-first computations made. Each time that a router comes online, or each time there is a change in topology, the router must perform SPF computations.
- **Memory failures** — Number of nonfatal memory allocation failures.
- **LSAs transmitted** — Number of link state advertisements transmitted.
- **LSAs received** — Number of link state advertisements received.
- **Route update errors** — Number of nonfatal routing table update failures.
- **Receive errors** — Number of general receive errors.
- **External LSA changes** — Number of external LSA changes made to database.
- **Soft restarts** — Number of OSPF router soft restarts due to insufficient memory resources (implies a fatal memory allocation failure). To fix this problem, change the OSPF memory partition with the ip ospf partition modify option, add memory, or reconfigure the network topology to generate smaller OSPF databases.

Standards, Protocols, and Related Reading

OSPF as implemented on this system is described in the following Internet Engineering Task Force (IETF) Request for Comment (RFC) documents:

- RFC 1583, Moy, J., *OSPF Version 2*, March 1994.
- RFC 1850, Baker, F., and Coltrun, R., *OSPF Version 2 Management Information Base*, November 1995.

Other useful reading includes:

- Moy, John, *OSPF: Anatomy of an Internet Routing Protocol*, Reading, MA., Addison-Wesley/Longman, ISBN 0201634724, 1997.
- RFC 1245, Moy, J., *OSPF Protocol Analysis*, July 1991.
- RFC 1586, DeSouza, O., and Rodriguez, M., *Guidelines for Running OSPF Over Frame Relay Networks*, March 1994.

15 IPX ROUTING

This chapter provides an overview, key concepts, guidelines, and other key information about using the Internet Packet Exchange (IPX) protocol to route packets to and from your system.

- IPX Routing Overview
- Key Concepts
- Key Guidelines for Implementation
- IPX Interfaces
- IPX Routes
- IPX Servers
- IPX Forwarding
- IPX RIP Mode
- IPX SAP Mode
- IPX Statistics
- Standards, Protocols, and Related Reading

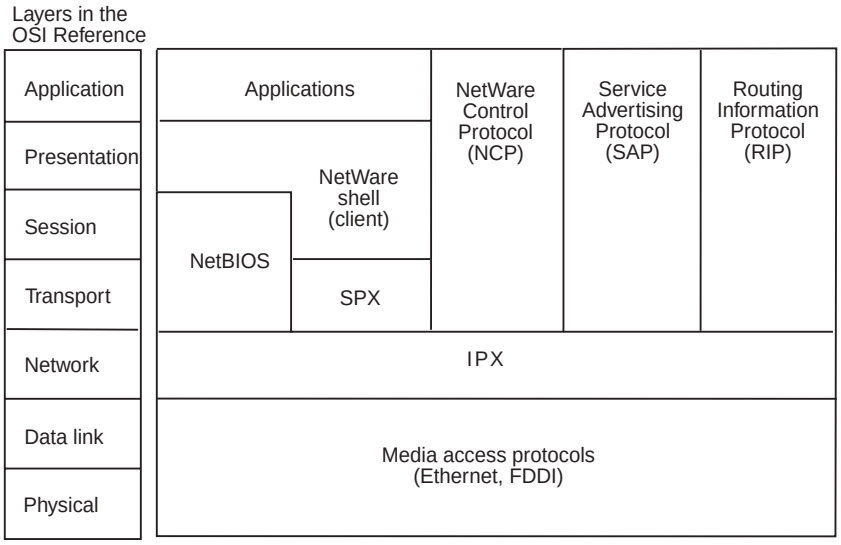
You can manage IPX routing from the `ipx` *top-level menu of the Administration Console. See the* Command Reference Guide*.*

IPX Routing Overview

You can route packets from your system to an external destination using the Internet Packet Exchange (IPX) protocol. The IPX protocol is a NetWare LAN communications protocol that moves data between servers and workstation programs that are running on various network nodes. IPX is a User Datagram Protocol (UDP), which is used for connectionless communications. IPX packets are encapsulated and carried by Ethernet packets and Token Ring frames.

Figure 74 shows the relationship of the IPX protocol to the Open System Interconnection (OSI) reference model.

Figure 74 IPX Protocol in the OSI Reference Model



Features Using the IPX protocol to route packets, you can create and support:

- IPX interfaces.
- IPX routes (primary and secondary).
- IPX servers (primary and secondary).
- IPX forwarding.
- IPX RIP mode.
- IPX SAP mode.

Benefits You can use IPX routing to:

- Provide services for connectionless communications.
- Reduce the cost of equipment moves, upgrades, and other changes and simplify network administration.
- Create VLAN-to-IPX interfaces to create virtual workgroups with most of the traffic staying in the same IPX interface broadcast domain.
- Help avoid flooding and minimize broadcast and multicast traffic.

Key Concepts

This section explains how IPX routing works and provides a glossary of IPX routing terms.

How IPX Routing Works

To route packets using the IPX protocol, take these general steps:

1. Define an IPX routing interface.
2. Decide which IPX routing and server options you want to use.
3. Enable IPX forwarding.

The IPX routing interface defines the relationship between an IPX VLAN and the subnetworks in the IPX network.

Each IPX VLAN interface is associated with a VLAN that supports IPX. The system has one interface defined for each subnetwork to which it directly connected.

A router operates at the Network layer of the OSI Reference Model. The router receives instructions to route packets from one segment to another from the network-layer protocol. IPX, with the help of the Routing Information Protocol (RIP), performs network-layer tasks, including:

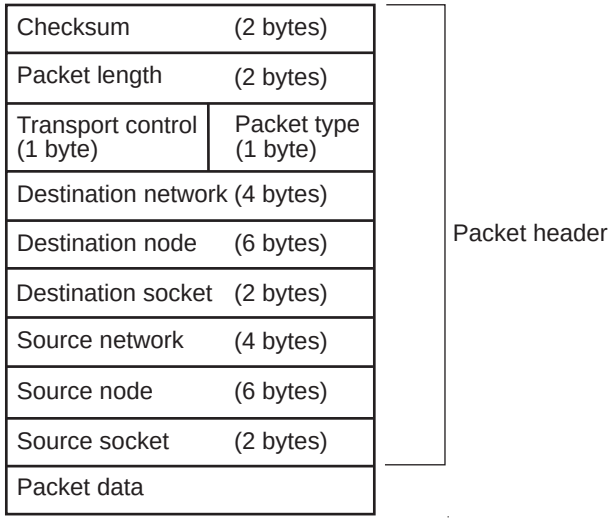
- Addressing packets
- Routing packets
- Switching packets

IPX Packet Format

An IPX packet consists of a 30-byte header followed by packet data. The packet header contains network, node, and socket addresses for both the destination and the source.

Figure 75 shows the IPX packet format.

Figure 75 IPX Packet Format



The IPX packet contains the following elements:

- **Checksum** — A 16-bit checksum that is set to 1s.
- **Packet length** — A 2-byte field that indicates the packet's length in bytes. This length, which includes both header and data, must be at least 30 bytes.
- **Transport control** — A 1-byte field that indicates how many routers a packet has passed through on its way to its destination. Packets are discarded when this value reaches 16. A network node sets this field to 0 before sending the IPX packet.
- **Packet type** — A 1-byte field that specifies the upper-layer protocol that receives the packet.
- **Destination network** — A 4-byte field that contains the network number of the destination node. When a sending node sets this field to 0, the system routes the packet as if the sending and destination nodes were on the same local segment.

- **Destination node** — A 6-byte field that contains the physical address of the destination node.
- **Destination socket** — A 2-byte field that contains the socket address of the packet's destination process.
- **Source network** — A 4-byte field that contains the source node network number. If a sending node sets this field to 0, the source's local network number is unknown.
- **Source node** — A 6-byte field that contains the source node physical address. Broadcast addresses are not allowed.
- **Source socket** — A 2-byte field that contains the socket address of the process that transmitted the packet.
- **Packet data** — A field that contains information for upper-layer network processes.

IPX Packet Delivery

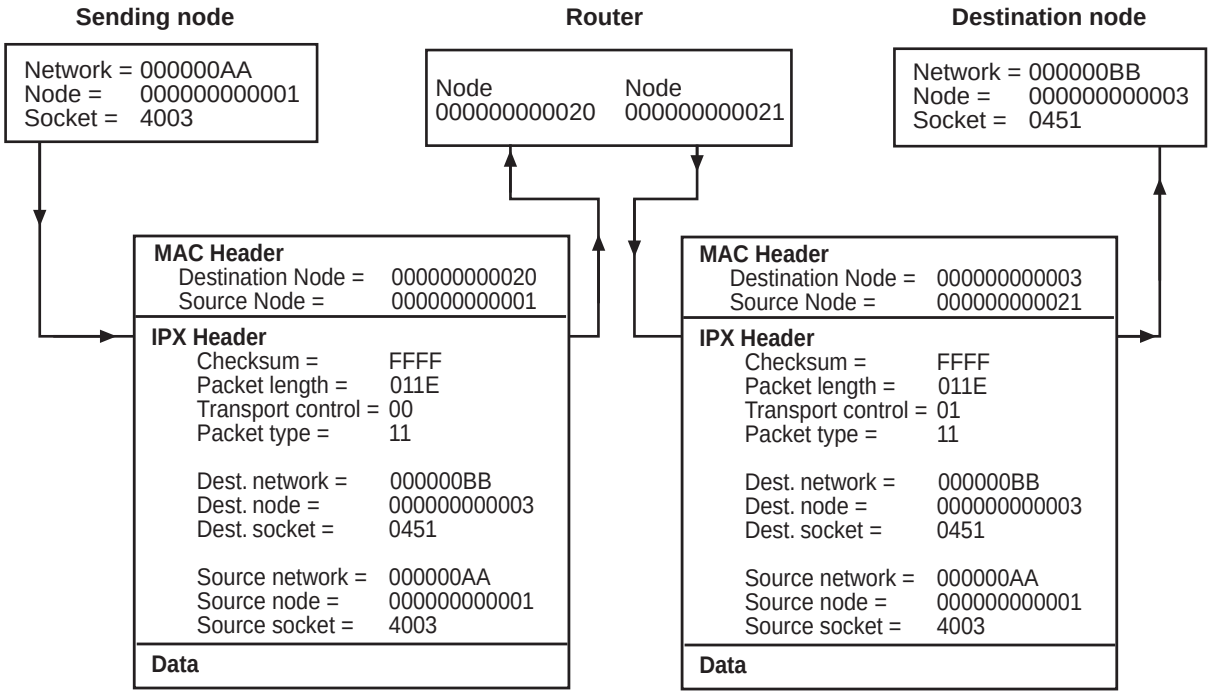
Successful packet delivery depends both on proper addressing and on the network configuration. The packet's Media Access Control (MAC) protocol header and IPX header address handle packet addressing.

The sending node must have the destination's complete network address, including the destination network, node, and socket. After the sending node has the destination address, it can address the packet.

However, the way the MAC header of the packet is addressed depends on whether a router separates the sending and destination nodes.

Figure 76 shows an example of IPX format routing.

Figure 76 IPX Packet Routing



Sending Node's Responsibility

When sending and destination nodes have the same network number, the sending node addresses and sends packets directly to the destination node. If sending and destination nodes have different network numbers, as in Figure 76, the sending node must find a router on its own network segment that can forward packets to the destination node's network segment.

To find this router, the sending node broadcasts a RIP packet, requesting the best route to the destination node's network number. The router on the sending node's segment that has the shortest path to the destination segment responds to the RIP request. The router's response includes its network and node address in the IPX header. After the sending node determines the intermediate router's address, it can send packets to the destination node.

If the sending node is a router rather than a workstation, the node's internal routing tables supply the destination's network location. The destination router does not need to broadcast a RIP request.

Router's Responsibility

A router handles a received IPX packet in one of two ways:

- If the packet is destined for a network number to which the router is directly connected, the sending router:
  - Places the destination node address from the IPX header in the destination address field of the packet's MAC header
  - Places its own node address in the source address field of the packet's MAC header
  - Increases the transport control field of the IPX header by 1 and transmits the packet on the destination node segment
- If the packet is destined for a network number to which the router is not directly connected, the router sends the packet to the next router along the path to the destination node. The sending router:
  - Looks up the node address in the routing information table of the next router and places the address in the destination address field of the packet's MAC header

- Places its own node address in the source address field of the packet's MAC header
- Increments the transport control field in the IPX header and sends the packet to the next router

Terminology

Review the following IPX routing terms that are used extensively throughout this chapter:

- **Address** — Unique 4-byte network address of a segment that is located in the routing table.
- **Age** — The time in seconds since the network's last update.
- **Cost** — A number between `1` and `65534` that the system uses to calculate route tiks. Assign a cost of 1 to each IPX interface unless your network has special requirements like the need for redundant paths.
- **Frame Formats** — Frame encapsulation format.
- **Hops** — The number of routers that must be crossed to reach a network segment.
- **Interface** — The system-assigned number for an IPX interface.
- **NetBIOS Protocol** — Network Basic Input Output System protocol. An application programming interface (API) that adds special functions for PC-based LANs.
- **Node** — The node address of the router that can forward packets to each network segment (when this is set to all `0`s, the router is directly connected).
- **RIP** — Routing Information Protocol. Allows the exchange of routing information on a NetWare network. IPX routers use RIP to create and maintain their dynamic routing tables.
- **SAP** — Service Advertisement Protocol. Provides routers and servers that contain SAP mode agents with a means of dynamically exchanging network service information.
- **Tiks** — An estimate of the time in seconds that is necessary to reach a network segment.
- **VLAN Interfaces** — Your system's point of attachment to a given VLAN. A VLAN interface exists entirely within a given IPX interface.

Key Guidelines for Implementation

Consider the guidelines in this section when you configure your system for IPX routing.

Procedural Guidelines

Complete the following steps to set up IPX routing on your system:

1. Set up your VLAN interfaces.
2. Define the IPX interfaces before you define the routes and servers.
3. Define routes.
4. Define servers.
5. Select RIP or SAP, if you plan to use them.
6. Define IPX forwarding.

General Guidelines

Consider the following general guidelines before you configure IPX routing on your system:

- Every IPX interface has one IPX VLAN and other associated information.
- The IPX router has one IPX interface defined for each network to which it is directly connected.
- Before you define an associated IPX interface for a network, you must first define a VLAN. See Chapter 9.
- The IPX router has one IPX interface defined for each network to which it is directly connected.

IPX Interfaces

An IPX interface has the following information associated with it:

- **IPX network address** — You must set this 4-byte address. Make each address unique within the network.
- **Cost** — A number between 1 and 65534 that the system uses to calculate route tiks. A tic is an estimate of how long it takes a packet to reach a network segment. One tic is approximately 55. Assign a cost of `1` to each IPX interface unless your network has special requirements like the need for redundant paths.
- **Encapsulation format** — IPX routing uses all four encapsulation formats: Ethernet Type II, Novell 802.3 Raw, 802.2 LLC, and 802.3 SNAP
- **State** — The status of the IPX interface. The IPX interface status can be up (available for communication) or down (unavailable for communication).
- **VLAN interface index (VLAN index)** — Identifies the VLAN that is associated with a IPX interface. When the system prompts you for this option, it indicates the available VLAN indexes.

Important Considerations

Consider the following guidelines when you set up an IPX interface:

- The first line in an interface display indicates whether:
  - IPX forwarding is enabled.
  - IPX RIP mode is active.
  - IPX RIP mode triggered updates are enabled.
  - IPX SAP mode is active.
  - IPX SAP triggered updates are enabled.
  - The secondary route/server option is enabled on the system.
- An IPX interface defines the relationships among an IPX VLAN, the IPX router, and the IPX network. The IPX router has one IPX interface defined for each network to which it is directly connected.
- When you define an IPX interface, you define the interface's:
  - IPX address.
  - Cost.
  - Format.
  - Associated IPX VLAN index.

- Before you define the IPX (routing) interface, you must define a VLAN and select IPX, IPX-II, IPX-802.2, IPX-802.2 LLC, or IPX-802.3-SNAP as the protocol to be supported by the VLAN. See Chapter 9.
- Unless your network has special requirements, such as the need for redundant paths, assign a cost of 1 to each interface and do not modify this setting.
- The three FDDI encapsulation formats correspond to the Ethernet 802.2 LLC, 802.3 SNAP, and RAW encapsulation formats. If you select either of these Ethernet encapsulation formats, the corresponding FDDI encapsulation format is automatically selected for shared Ethernet and FDDI ports.
- When you modify an IPX interface, you define the interface's:
  - IPX address.
  - Cost.
  - Format.
  - Associated IPX VLAN index.
- If you use the OddLengthPadding feature, make sure that you select only those interfaces that require odd-length padding. If you enable this option for every interface, network performance slows.

To create an IPX interface, see the Administering IPX Routing chapter in the *Command Reference Guide*.

Per-Interface Options

You set the NetBIOS and OddLengthPadding options on each interface.

NetBIOS Option

This option determines whether the system handles IPX Type 20 packet forwarding on each interface. For details about how to use this option, see the Administering IPX Routing chapter in the *Command Reference Guide*.

OddLengthPadding Option

This option provides compatibility with older network interface cards (NICs). This option enables an interface to pad IPX packets that have an odd number of bytes, so that older NICs do not discard the packets. To use this option, see the chapter about IPX routing in the *Command Reference Guide*.

IPX Routes

Your system maintains a table of routes to other IPX networks. You can:

- Use RIP mode to exchange routing information dynamically.
- Use the Administration Console to make static entries in the table.

Important Considerations

Consider the following guidelines when you set up an IPX route:

- The first line in the output (the status line) indicates whether:
  - IPX forwarding is enabled.
  - IPX RIP mode is active.
  - IPX RIP mode triggered updates are enabled.
  - IPX SAP mode is active.
  - IPX SAP triggered updates are enabled.
  - The secondary route/server option is enabled on the system.
- The route table display shows the range for routing table *Primary* entries in the *n-m* format, where *n* is the current number of entries and *m* is the maximum number of Primary entries.
- If your system has extended memory, the route table display includes a range for the routing table *Secondary* entries in the *N-M* format, where *N* is the minimum number of entries and M is the maximum number of secondary entries. If no range is displayed, the system does not have extended memory, so the number of route table entries is a fixed number.
- A Secondary route entry can replace a Primary route entry when the Primary route is removed from the routing table for any reason (for example, if the route reaches its age limit).
- To view entries for any Secondary routes:
  - Establish alternate paths to the same IPX network.
  - Enable the IPX Secondary route/server option.
- The maximum number of hops, or routers that a packet can cross, is 15, except for NetBIOS packets which can cross no more than 7 routers.
- Before you define static routes on your system, you must define at least one IPX interface.
- Static routes remain in the routing table until you remove them or until you remove the corresponding interface.

- If an interface goes down, routes are temporarily removed from the routing table until the interface comes back up.
- Static routes take precedence over dynamically learned routes to the same destination. You can have a maximum of 32 static routes.
- When you use the IPX route remove option to remove a route, that route is immediately removed. All servers that depend on the removed route are also removed from the Server Information Table, including all static servers.
- When you use the IPX route flush option to remove dynamically learned routes from the IPX routing table, all dynamically learned routes are immediately removed. All dynamic servers that depend on these routes are also removed from the Server Information Table.

Primary and Secondary Routes

You can set up both Primary and Secondary routes in the routing table.

To set up routes in the routing table, see the IPX chapter in the *Command Reference Guide*.

Static Routes

You manually configure a static route. Static routes are useful in environments in which no routing protocol is used or when you want to override a routing protocol's generated route.

Static routes do not change until you change them, and they do not time out. Because static routes do not change in response to network topology changes, manually configure only a small number of reasonably stable routes.

Dynamic Routes Using RIP

A router uses RIP to exchange its routing table with other routers at regular intervals. This automatic method of learning routes helps you keep up with a changing network environment and allows you to reconfigure routes quickly and reliably. Interior Gateway Protocols (IGPs), which operate within intranetworks, provide this automated learning. The system uses RIP (one of the most widely used IGPs) to dynamically build routing tables.

RIP operates with active and passive network devices:

- **Active devices** — Usually routers, they broadcast their RIP messages to all devices in a network; they update their own routing tables when they receive a RIP message.
- **Passive devices** — Usually hosts, they listen for RIP messages and update their routing tables; they do not send RIP messages.

On your system, you select a RIP mode to determine how RIP operates, as described in "IPX RIP Mode" later in this chapter.

An active router sends a RIP message every 60 seconds. This message contains both the network number for each destination network and the number of hops to reach it. In RIP, each router through which a packet must travel through to reach a destination counts as one network *hop*.

Routing Tables

A routing table collects information about all intranetwork segments. This table allows a router to send packets toward their destinations over the best possible routes.

The table contains an entry for every network number that the router knows about. The router uses this information when the router is not directly connected to a packet's destination network. The routing information table provides the address of another router that *can* forward the packet toward its destination.

The routing table consists of the following elements:

- **Interface** — The interface number of the router that is used to reach a network segment
- **Address** — The network segments that the router knows about
- **Hops to network** — The number of routers that must be crossed to reach a network segment
- **Tiks to network** — An estimate of the time in seconds that is necessary to reach a network segment
- **Node** — The node address of the router that can forward packets to each network segment. When the node is set to all 0s, the router is directly connected.
- **Aging timer** — The time in seconds since the network's last update

Figure 77 shows an example of a typical routing information table.

Figure 77 Sample Routing Table

Routing table

Interface	Address	Hops	Tics	Node	Age
1	1	1	1	00-00-00-00-00-00	0
2	45469f30	1	1	00-00-00-00-00-00	0
2	45469f33	2	3	08-00-17-04-33-45	40

The routing information table is updated statically or dynamically.

Selecting the Best Route

Large networks contain many possible routes to each destination. A router performs the following steps to find the best route toward a destination:

- If one route requires the lowest number of tiks, the router selects it as the best route.
- If multiple routes require the same lowest number of tiks, the router selects the route that requires the lowest number of hops as the best route.
- If multiple routes require the same lowest number of tiks and hops, the router may select any of them as the best route.

IPX Servers

Your system creates and maintains a server information table that lists all the servers that reside on other IPX networks. You can:

- Use SAP to exchange server information dynamically.
- Make static entries in the server table.

Important Considerations

Consider the following guidelines when you set up an IPX server:

- The first line in the output (status line) indicates whether:
  - IPX forwarding is enabled.
  - IPX RIP mode is active.
  - IPX RIP mode triggered updates are enabled.
  - IPX SAP mode is active.
  - IPX SAP triggered updates are enabled.
  - The secondary route/server option is enabled on the system.
- The route table display shows the range for the server table primary entries in the format n-m, where *n* is the current number of entries and *m* is the maximum number of entries.
- Static servers remain in the table until you:
  - Remove them.
  - Remove the corresponding interface.
  - Remove the route to the corresponding network address.
- A static server must have an IPX network address that corresponds to a configured interface or to a static route. If an interface goes down, any static servers on that interface are permanently removed from the server table until the interface comes back up.
- Static servers take precedence over dynamically learned servers to the same destination. You can have a maximum of 32 static servers.
- Before you define static servers on the system, first define at least one IPX interface.
- When you use the ipx server remove option to remove a server, that server is immediately removed from the Server Information Table.
- When you use the `ipx server flush` command to remove all dynamically learned servers, all dynamically learned servers are immediately removed from the Server Information Table.

Primary and Secondary Servers

You can set up both Primary and Secondary servers in the server table. You can set up Secondary servers to serve as a backup to the Primary server set up on the same IPX server.

To set up Secondary servers on your system, see the IPX chapter in the *Command Reference Guide*.

Static Servers

Static servers are useful in environments in which no routing protocol is used or when you want to override some of the servers that were generated with a routing or server protocol. Because static servers do not change in response to network topology changes, manually configure only a small number of relatively stable servers.

Dynamic Servers Using SAP

Servers are automatically added to and removed from the information table through SAP. This automatic SAP update helps you keep up with changing network environments and allows servers to advertise their services and addresses quickly and reliably.

As servers boot up, they advertise their services. When servers are brought down, they use SAP to broadcast that their services are no longer available.

Client systems do not use this server information directly. Instead, SAP agents within each router on the server's network segment collect this information. The SAP agents store information in their server information tables. Client systems then contact the nearest router or file server SAP agent to obtain server and service information.

On your system, you select a SAP mode to determine how SAP operates, as described in "IPX SAP Mode" later in this chapter.

Maintaining Server Information

When a router's SAP agent receives a SAP broadcast response indicating a change in a server's configuration, the agent updates its server information table and informs other SAP agents. Examples of such a change are when a server is disconnected or becomes accessible through a better route.

The SAP agent immediately sends an update broadcast to all directly connected network segments except the segment from which the information was received. All future periodic broadcasts contain the change information.

SAP Aging

Router SAP agents use a special aging mechanism to deal with a SAP agent that goes down suddenly without sending a DOWN broadcast. A hardware failure, power interruption, or power surge can cause this situation.

Each SAP agent maintains a timer for each entry in its server information tables. The timer tracks the elapsed time since this entry has been updated. This information is either new or changed, and the SAP agent immediately passes it on. Changes are quickly captured and stored throughout the intranetwork.

SAP Request Handling

When a SAP agent receives a general request, it notifies the sending source about all servers known to the agent. This response includes the same information that is sent out in periodic SAP broadcasts. When the request is specific, the SAP agent notifies the sending source about all servers of the requested type.

Server Tables

Server information tables contain data about all active servers on the intranetwork. SAP agents use these tables to store information received in SAP broadcasts. Server tables are dynamically and statically created.

Figure 78 shows an example of a Server Information Table.

Figure 78 Sample Server Information Table

Server information table

Interface	Name	Type	Network	Node	Socket	Hops	Age
1	LPX1102	4	45469f33	00-00-00-00-00-01	451	2	102
1	LPX1103	4	45469f44	00-00-00-00-00-01	451	5	65
2	LPX2001	4	45470001	00-00-00-00-00-01	451	4	33

This table contains the following data:

- **Interface** — The interface from which server information is received
- **Server name** — The name of the server
- **Server type** — The type of service the server provides
- **Network address** — The address of the network that contains the server
- **Node address** — The server's node address
- **Socket address** — The socket number through which the server receives service requests
- **Hops to server** — The number of intermediate networks that must be crossed to reach the server
- **Age of server** — The time in seconds since the server's last table update

IPX Forwarding

You can control whether the system forwards or discards IPX packets with the `ipx forwarding` option.

Important Considerations

Consider the following guidelines before you use the `ipx forwarding` option:

- When you enable ipx forwarding, the system acts as a normal IPX router. It forwards IPX packets from one network to another when required.
- When you disable ipx forwarding, the system discards all IPX packets.

IPX RIP Mode

You can exchange routing information on a NetWare network using the `ipx rip mode` option. This option selects the IPX RIP mode that is appropriate for your network and selects the routers that use RIP mode to create and maintain their dynamic routing tables.

In IPX RIP mode, one router exchanges routing information with a neighboring router. When a router discovers any changes in the network layout, it broadcasts this information to any neighboring routers. IPX routers also send periodic RIP broadcast packets that contain all routing information. These broadcasts synchronize all routers on the network and age those networks that might become inaccessible if a router is disconnected abnormally from the network.

Important Considerations

Consider the following guidelines before you use the `ipx rip mode` option:

- The system has three RIP modes:
  - **Off** — The system processes no incoming RIP packets and generates no RIP packets of its own.
  - **Passive** — The system processes all incoming RIP packets and responds to RIP requests, but it does not broadcast periodic or triggered RIP updates.
  - **Active** — The system processes all incoming RIP packets, responds to explicit requests for routing information, and broadcasts periodic and triggered RIP updates.
- The system has two RIP triggered modes:
  - **disabled** — Broadcasts IPX routes 3 seconds after learning them.
  - **enabled** — Broadcasts IPX routes immediately after learning them.

RIP Policies

Each router maintains a table of current routing information (the routing table). The routing protocols receive or advertise routes from the network. RIP policies control the flow of routing information among the network, the protocols, and the routing table manager.

Routing policies allow you to define:

- The import policies that specify which routes the router places into the routing table.
- The export policies that specify the routes that the router propagates to the network.
- The import and export policies for each peer.

RIP Import Policies

Before the router adds a route to the routing table, it follows these steps:

- The protocol receiving the route forwards the route to the routing table manager.
- The routing table manager compares the route to the import policy to determine whether to accept or drop the route.
- If the routing table manager accepts the route, it stores the route in the routing table.

The default import policy is none; that is, the router places all routes into the routing table.

RIP Import Policies

At certain times, such as when the routing table changes, the protocol asks the routing table manager for routes to advertise to other routers. The routing table manager follows these steps:

- It compares the route to the export policy to determine whether to advertise the route to the network.
- If it accepts the route, the manager propagates it to the network.

The default import policy is none; that is, the router advertises all routes in the routing table.

RIP Policy Parameters

These parameters define SAP policies:

- **Policy type** — Import (apply the policy to received services) or Export (apply the policy to advertised services).
- **Route origin** — The origin of the route for this policy if it is an export policy: static, RIP, or all.
- **Route** — An IPX network address that specifies the route that applies to this policy.
- **Interface** — One or more IP interfaces on this router that are associated with the RIP policy.
- **Source Node Address** — The MAC address of the router that can forward packets to the network.
- **Action** — Whether this router accepts or rejects a route that matches the policy.
- **Metric** — Increase or decrease a route metric by a value that you specify. This parameter is valid only if the Policy Action is set to Accept (import policies).

To change the route metric of an export policy, you must adjust the metric of the import policy on the receiving router.

- **Weight** — The metric value of this policy. This parameter specifies the order of precedence for policies that match the same route. A higher value takes precedence over a lower value.

IPX SAP Mode

IPX SAP provides routers and servers that contain SAP mode agents with a means of exchanging network service information. Through SAP, servers advertise their services and addresses. Routers gather this information and share it with other routers. With this process, routers dynamically create and maintain a database (server table) of network service information. Clients on the network determine what services are available and obtain the network address of the nodes (servers) where they can access those services. Clients require this information to initiate a session with a file server.

You determine how SAP operates on your system with the `ipx sap mode` option.

Important Considerations

Consider the following guidelines before you use the `ipx sap mode` option:

- The system has three SAP modes:
  - **Off** — The system does not process any incoming SAP packets and does not generate any SAP packets of its own.
  - **Passive** — The system processes all incoming SAP packets and responds to SAP requests, but it does not broadcast periodic or triggered SAP updates.
  - **Active** — The system processes all incoming SAP packets, responds to explicit requests for routing information, and broadcasts periodic and triggered SAP updates.
- The system has two SAP triggered modes for updates:
  - **Disabled** — Broadcasts IPX SAP server addresses 3 seconds after learning them.
  - **Enabled** — Broadcasts IPX SAP server addresses immediately after learning them.

SAP Policies

Each router maintains a table of current configured services (the service table). SAP receives information and advertises information about the network nodes that provide these services. SAP policies control which services the router places in the service table and advertises to the network.

SAP Import Polices

Each time that the router receives an advertised service, it compares the service to the import polices to decide whether to add the service to the service table or drop it. If the router accepts the service, the router adds it to the service table.

The default import policy is none; that is, the router places all services into the service table.

SAP Export Policies

At certain times, such as when a router is started up or shut down, SAP advertises services to other routers. Each time the router prepares to advertise the service, it compares it to the export policies to decide whether to advertise the service. If the export policy does not prohibit the service, the router sends it out.

The default export policy is none; that is, the router advertises all services.

SAP Policy Parameters

These parameters define SAP policies:

- **Policy type** — Import (apply the policy to received services) or Export (apply the policy to advertised services).
- **Route origin** — The origin of the service for this policy, if it is an export policy: static, SAP, or all.
- **Service type** — The Novell standard 6-digit hexadecimal number that represents the type of service offered by the server. For details, consult your Novell documentation. See the *Command Reference Guide* for a list of common service types.
- **Server name** — The name of the server providing the services.
- **Network address** — The IPX network address of the network on which the server resides.

- **Node address** — The 6-byte MAC address of the router that can forward packets to the network.
- **Interfaces** — One or more IP interface index numbers associated with this policy.
- **Action** — Whether this router accepts or rejects a service that matches the policy.
- **Weight** — The metric value that is associated with this policy. This parameter specifies the order of precedence for policies that match the same service. A higher value takes precedence over a lower value.

IPX Statistics

You can view the following IPX statistics on your system:

- IPX summary statistics
- IPX RIP statistics
- IPX SAP statistics
- IPX forwarding statistics
- IPX interface statistics

In the display, the status line indicates whether:

- IPX forwarding is enabled.
- RIP mode is active.
- RIP mode triggered updates are enabled.
- SAP mode is active.
- SAP mode triggered updates are enabled.
- The secondary route/server option is enabled.

See the IPX chapter in the *Command Reference Guide* for more information about IPX statistics.

Standards, Protocols, and Related Reading

The following standards and protocols apply when you use IPX to route packets on your system:

- IEEE 802.2
- IEEE 802.2 LLC
- IEEE 802.3
- IEEE 802.3-RAW
- IEEE 802.3-SNAP
- Internet Packet eXchange (IPX) — RFC 1234, RFC 1552
- Routing Information Protocol (RIP) — RFC 1058
- Service Advertisement Protocol (SAP) — NetWare Protocol

16 APPLETALK

This chapter provides guidelines, limitations, and other key information about routing with AppleTalk technology. This information includes:

- AppleTalk Overview
- Key Concepts
- Key Implementation Guidelines
- AppleTalk Interfaces
- AppleTalk Routes
- AppleTalk Address Resolution Protocol (AARP) Cache
- AppleTalk Zones
- Forwarding AppleTalk Traffic
- Checksum Error Detection
- AppleTalk Echo Protocol (AEP)
- AppleTalk Statistics
- Standards, Protocols, and Related Reading

You can manage AppleTalk from the `appletalk` *top-level menu of the Administration Console. See the* Command Reference Guide.

AppleTalk Overview

AppleTalk is a suite of protocols defined by Apple Computer, Inc., for connecting computers, peripherals devices, and other equipment to a network. AppleTalk protocols support most of the functions offered by the Open Systems Interconnection (OSI) Reference Model.

The AppleTalk protocols work together to provide file sharing and printer sharing, as well as applications like electronic mail and database access. All Macintosh computers have AppleTalk connectivity options built into them, which makes it the *de facto* standard for Apple networks.

AppleTalk transport and application services operate over a best-effort Delivery Datagram Protocol (DDP). The AppleTalk Data Steam Protocol (ADSP) ensures reliable transmission of AppleTalk information.

Your system supports AppleTalk version 2, which runs the AppleTalk Routing Table Maintenance Protocol (RTMP). As a distance-vector based routing protocol, RTMP constructs the best paths based on hop-count information that is propagated by neighbors.

Features AppleTalk routing includes these features:

- **AppleTalk interfaces** — An AppleTalk interface is an interface that can send and receive AppleTalk traffic. When you configure an AppleTalk interface, you define the behavior and role of the interface within the AppleTalk routing domain. For example, seed interfaces propagate network configuration information, while nonseed interfaces listen for it. See "AppleTalk Interfaces" later in this chapter for more information.
- **AppleTalk routes** — Your system maintains a table of reachable AppleTalk networks. You may want to view the contents of this table for administrative purposes. See "AppleTalk Routes" later in this chapter for more information.
- **AppleTalk Address Resolution Protocol (AARP) cache** — The AARP cache contains a listing that maps each known AppleTalk address to a corresponding MAC address. Your system lets you view this listing, as well as remove entries from it. See "AppleTalk Address Resolution Protocol (AARP) Cache" later in this chapter for more information.
- **AppleTalk zones** — All resources on an AppleTalk network are grouped into zones. Zones make AppleTalk resources easier to identify and locate. Your system maintains a zone table which maps network numbers to zones, and lets you display this zone table indexed by network numbers, or by zones. See "AppleTalk Zones" later in this chapter for more information.
- **Forwarding AppleTalk traffic** — You can disable or enable the forwarding of AppleTalk traffic on a system-wide basis. See "Forwarding AppleTalk Traffic" later in this chapter for more information.

- **Checksum error detection** — AppleTalk uses checksums to detect errors in data transmissions. Your system allows you to enable or disable checksum generation and verification. See "Checksum Error Detection" later in this chapter for more information.
- **AppleTalk Echo Protocol (AEP)** — Your system supports AppleTalk Echo Protocol, which you can use to test the connectivity and response of an AppleTalk device. See "AppleTalk Echo Protocol (AEP)" later in this chapter for more information.
- **AppleTalk statistics** — You can also display AppleTalk statistics for a number of AppleTalk protocols. These statistics can help you diagnose and troubleshoot network issues and performance problems. See "AppleTalk Statistics" later in this chapter for more information.

Benefits The benefits of AppleTalk include:

- AppleTalk is built into all Apple devices, making them automatically network capable. This makes AppleTalk an extremely easy network system to install and operate.
- The naming mechanism AppleTalk uses frees users from having to understand anything about how AppleTalk works.
- AppleTalk supports peer-to-peer networking, so no dedicated servers or centralized network control is required.
- AppleTalk is plug-and-play, or auto-configuring. This allows users to plug an AppleTalk device into an AppleTalk network and use it immediately.
- No configuration of network information or assigning of network addresses is required when you add a device to an AppleTalk network.
- In theory, AppleTalk networks can support millions of nodes.
- AppleTalk supports zones, which makes it easier for network administrators to define workgroups consisting of users and services that can span multiple network segments.

Key Concepts

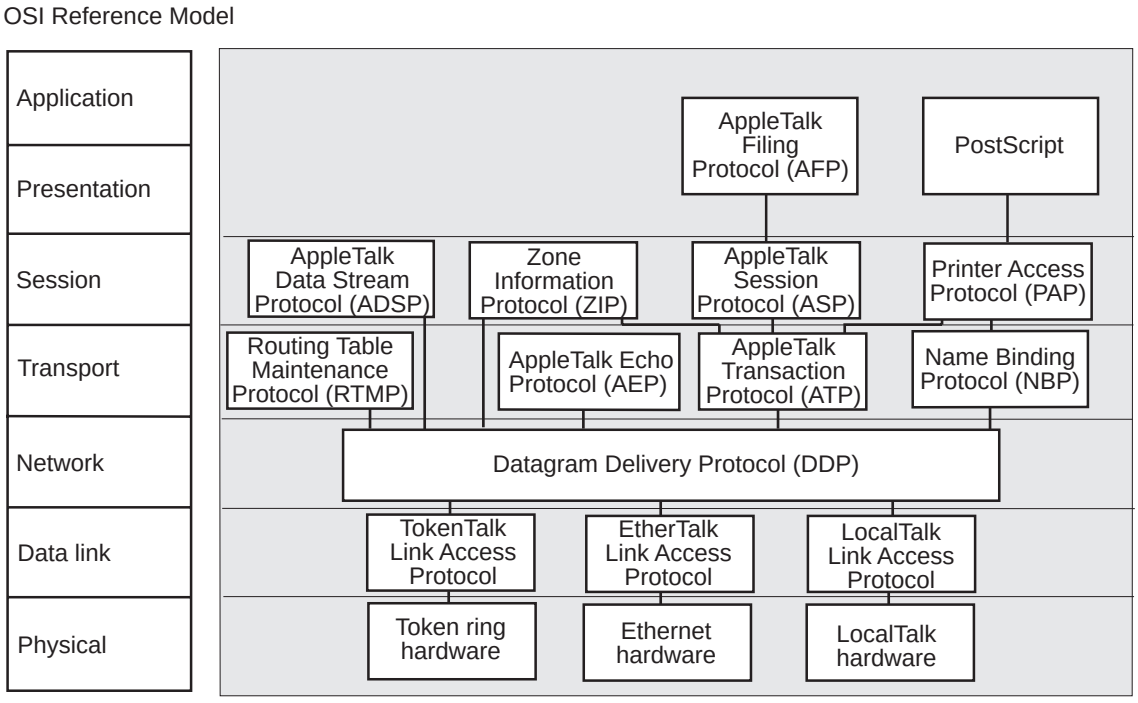
Before configuring AppleTalk, review the following key concepts and terms discussed in these sections:

- AppleTalk Protocols
- AppleTalk Network Elements
- Terminology

AppleTalk Protocols

AppleTalk protocols ensure the flow of information through AppleTalk networks. Figure 79 shows a simplified view of AppleTalk protocols and their relationship to the OSI Reference Model. These protocols provide physical connectivity, end-to-end network services, and data delivery.

Figure 79 AppleTalk Protocols and the OSI Reference Model



The AppleTalk six-layer protocol suite does not fully comply with the OSI seven-layer model. However, AppleTalk provides many of the functions and services of OSI. AppleTalk has no specific protocols for the Application layer because the lower levels provide printer and file service.

Physical Layer Protocols

The Physical layer of the OSI protocol stack defines the connection with network hardware. With AppleTalk, you can use standard network hardware, such as that designed for Ethernet and token ring networks. Apple has also defined its own network hardware, called LocalTalk, which uses a synchronous RS-422A bus for communications.

Link Layer Protocols

The data link layer provides the interface between the network hardware and the upper layers of the protocol stack. The AppleTalk data link layer includes three link access protocols (LAPs):

- TokenTalk LAP (TLAP)
- Ethernet LAP (ELAP)
- LocalTalk LAP (LLAP).

The AppleTalk Address Resolution Protocol (AARP), which translates hardware addresses to AppleTalk addresses, also exists at the data link layer because it is closely related to the Ethernet and token ring LAPs. AARP is usually included in the definition of each LAP, so it does not appear in the reference model. See "AppleTalk Address Resolution Protocol (AARP) Cache" later in this chapter for more information about this protocol.

Network Layer Protocols

The network layer accepts data from the layers above it and divides the data into packets to send over the network through the layers below it. The Datagram Delivery Protocol (DDP) transfers data in packets called *datagrams*.

Datagram delivery is the basis for building other AppleTalk services such as electronic mail. With DDP, AppleTalk runs as a process-to-process, best-effort delivery system in which the processes running in the nodes of interconnected networks exchange packets with each other.

Transport Layer Protocols

The Transport layer and the Session layer provide end-to-end services in the AppleTalk network. These services ensure that routers transmit data accurately between one another. Each layer includes four protocols that work together to support these services. This section describes these protocols and provides more detail for the protocols that you can view using the Administration Console.

An AppleTalk intranet has four transport layer protocols:

- Routing Table Maintenance Protocol (RTMP)
- AppleTalk Echo Protocol (AEP)
- AppleTalk Transaction Protocol (ATP)
- Name Binding Protocol (NBP)

Routing Table Maintenance Protocol (RTMP) This protocol maintains information about AppleTalk addresses and connections between different networks. It specifies that each router:

- Learns new routes from other routers.
- Deletes a route if the local router has not broadcast the route to the network for a certain period of time.

Each router builds a routing table for dynamic routing operations in an AppleTalk intranet. Every 10 seconds, each router sends an RTMP data packet to the network. Routers use the information that they receive in the RTMP broadcasts to build their routing tables. Each entry in the routing table contains these items:

- The network range
- The distance in hops to the destination network
- The interface number of the destination network
- The state of each port (*good*, *suspect*, *bad*, or *really bad*)

A router uses these items to determine the best path along which to forward a data packet to its destination. The routing table contains an entry for each network that a router's datagram can reach within 15 hops. The table is aged at set intervals as follows:

1. After a specified period of time, the RTMP changes the status of an entry from *good* to *suspect*.
2. After an additional period of time, the RTMP changes the status of an entry from *suspect* to *bad*.
3. After an additional period of time, the RTMP changes the status of an entry from *bad* to *really bad*.
4. The router removes the entry of a nonresponding router with a *really bad* status.

The data in the routing table is cross-referenced to the Zone Information Table (ZIT). This table maps networks into zones. See "Session Layer Protocols" later in this chapter for more information about the ZIT.

Figure 80 illustrates a simple AppleTalk network, and Table 58 shows the corresponding routing table.

Figure 80 A Simple AppleTalk Network

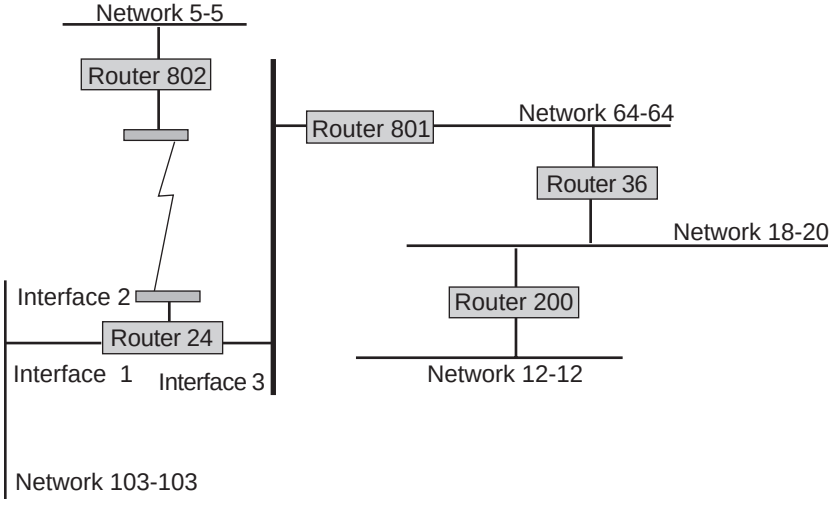


Table 58 Routing Table for Router 24 in Figure 80

Network Range	Distance (hops)	Interface	State
5-5	1	2	Good
12-12	3	3	Good
18-20	2	3	Good
103-103	0	1	Good
64-64	1	3	Good

You view the AppleTalk routing tables in your network through the Administration Console.

AppleTalk Echo Protocol (AEP) AppleTalk nodes use the AEP to send datagrams to other nodes in the network. The AEP datagram transmitted causes the destination node to return, or *echo*, the datagram to the sending node. This protocol determines whether a node is accessible before any sessions are started, and it enables users to estimate the round-trip delay time between nodes.

AppleTalk Transaction Protocol (ATP) This protocol, along with the AppleTalk Data Stream Protocol (ADSP), ensures delivery of DDP packets to a destination without any losses or corruption.

Name Binding Protocol (NBP) This protocol translates alphanumeric entity names to AppleTalk addresses. NBP maintains a table of node addresses and named entities within each node. Because each node also maintains its own list of named entities, the names directory within an AppleTalk network is not centralized. The names directory database is distributed among all nodes on the intranet.

Session Layer Protocols

An AppleTalk intranet has four session-layer protocols:

- AppleTalk Data Stream Protocol (ADSP)
- Zone Information Protocol (ZIP)
- AppleTalk Session Protocol (ASP)
- Printer Access Protocol (PAP)

AppleTalk Data Stream Protocol (ADSP) The ADSP works with the ATP to ensure reliable data transmission. Unlike ATP, however, ADSP provides full-duplex byte-stream delivery. Therefore, two nodes can communicate simultaneously. ASDP also includes flow control, so that a fast sender does not overwhelm a slow receiver.

Zone Information Protocol (ZIP) ZIP works with RTMP to map network numbers to network zones for the entire AppleTalk intranet. Network zones are the logical groupings of AppleTalk networks. The table created by ZIP is called the *Zone Information Table (ZIT)*. You view the ZIT by network number or network zone from the Administration Console.

ZIP creates a zone information table in each router. Each entry in the ZIT is a *tuple*, or pair, that includes a network number and a network zone name. When an NBP packet arrives at the router, the router compares the zone name in the packet with zone names in the ZIT entries. The router then compares the network number in the matching ZIT entry with the network number in the RTMP table, to find the interface for routing the packet.

AppleTalk Session Protocol (ASP) The ASP passes commands between a workstation and a server after they connect to each other. ASP ensures that the commands are delivered in the same order that they were sent and returns the results of these commands to the workstation.

Printer Access Protocol (PAP) The PAP maintains communications between a workstation and a printer or print service. The PAP functions include setting up and maintaining a connection, transferring the data, and tearing down the connection on completion of the job. Like other protocols at the session layer, PAP relies on NBP to find the addresses of named entities. PAP also depends on ATP for sending data.

Presentation Layer Protocols

The presentation layer maintains information about files, formats, and translations between formats. An AppleTalk intranet has two protocols at the presentation layer: the AppleTalk Filing Protocol (AFP) and PostScript. AFP provides remote access to files on the network. PostScript is a graphic page description language used by many printers.

AppleTalk Network Elements

An AppleTalk network consists of different nodes and groups of networks. Nodes can include workstations, routers, printers, and servers that provide services for other computers, called *clients*.

This section describes the elements of an AppleTalk network:

- AppleTalk Networks
- AppleTalk Nodes
- Named Entities
- AppleTalk Zones
- Seed Routers

AppleTalk Networks

A subnetwork in an AppleTalk intranet is a cable segment attached to a router. Each subnetwork is identified by a network number or range of network numbers. You assign these numbers from a range of valid network numbers.

Two AppleTalk network numbering systems are currently in use: nonextended (Phase 1) and extended (Phase 2). 3Com routers support extended network numbers. While the CoreBuilder® 3500 system does not translate Phase 1 packets to Phase 2 packets, it does route packets to a Phase 1 network. The system anticipates that a gateway exists between the two networks to translate the packets.

An extended intranet can span a range of logical networks. Network numbers in an extended network consist of a range, such as network 15 through 20. This numbering scheme allows as many as 16,580,608 nodes, although the actual cables do not support this many nodes.

AppleTalk Nodes

A node in a AppleTalk network is any addressable device, including a workstation, printer, or router. Nodes are physically attached to a network. At initialization, each node in an AppleTalk network selects a unique AppleTalk address. The address consists of the node's network number and a unique node number.

Named Entities

When a device on the network provides a service for other users, you can give the device a name. The name appears on the *Chooser* menu of the Macintosh with an associated icon. For example, the Chooser of the Macintosh can include a printer icon. When the user selects the printer icon, several printer names can appear in a list, such as `Laser1` or `Laser2`. The Name Binding Protocol (NBP), described later in this chapter, translates these device names into AppleTalk addresses.

AppleTalk Zones

An AppleTalk zone is a logical collection of nodes on an AppleTalk intranet. Zones make it easier to locate devices. Because your system supports AppleTalk, Phase 2, you can associate a list of zones for each network. Nodes on the network may belong to any of the zones associated with the network, and you can associate the same zone name with multiple networks. For more information about zones, see "AppleTalk Zones" later in this chapter.

Seed Routers

A seed router initializes the intranet with AppleTalk configuration information, including network numbers and zone names. The seed router broadcasts this information so that nonseed routers can learn it. You designate a seed router through the Administration Console.

A nonseed router listens for a seed router and takes configuration information from the first one it detects. A nonseed router that obtains configuration data participates in the network as if it is a seed router.

Terminology

If you are unfamiliar with AppleTalk routing, you may want to review the following terms:

- **Seed router** — A router that initializes the AppleTalk network with the network range and zone list information that you configure.
- **Non-seed router** — A router that listens for a seed router and obtains its network range and zone information from the seed interface that it detects.
- **Zone** — A logical subset of the systems on an AppleTalk internetwork. For example, a logical group of AppleTalk networks. For more information, see "AppleTalk Zones" later in this chapter.
- **Network range** — The range of network numbers assigned to an AppleTalk extended (Phase 2) network.

- **Phase 1 network** — Also known as a *nonextended network*, AppleTalk networks that contain a single network number (such as network 2). Phase 1 networks do not allow two nodes on a single network segment to belong to different zones.
- **Phase 2 network** — Also known as an *extended network*, AppleTalk networks that contain multiple consecutive network numbers (such as network 3-20), though it can also contain a single network number (such as network 3-3).
- **AppleTalk Address Resolution Protocol (AARP)** — An AppleTalk support protocol that maps the hardware address of an AppleTalk node to an AppleTalk protocol address.
- **Hop Count** — The number of routers a packet must cross to reach a destination network.
- **AppleTalk Echo Protocol (AEP)** — An AppleTalk support protocol used to test the accessibility of a system and make an estimate of the route-trip transmission time required to reach the system.
- **Checksum** — A method providing error detection for AppleTalk packets, calculated by summing a set of values.

Key Implementation Guidelines

Consider the following guidelines when designing a dependable and scalable AppleTalk network:

- All AppleTalk routers on the same network segment must have the same configuration. This means all seed routers must be configured with matching:
  - Network ranges.
  - Default zones.
  - Zone lists.

  If a configuration mismatch occurs between routers on the same segment, then unpredictable behavior may result. For example, zones may fail to show up in Chooser, and AppleTalk services may become inaccessible.
- If you are connecting your system's AppleTalk Phase 2 routing interface to an AppleTalk Phase 1 network, follow these guidelines:
  - Specify a network range of 1 (for example, `22-22`).
  - The network can belong to only one zone.

AppleTalk Interfaces

On the CoreBuilder 3500, an AppleTalk interface defines the relationship between a virtual LAN (VLAN) and an AppleTalk network. An AppleTalk interface has these elements associated with it:

- **Seed Interface** — You can configure the interface to be a seed or nonseed interface:
  - *A seed interface* initializes ("seeds") the network with your configuration information. This information includes the network range and zone name list.
  - *A nonseed interface* listens for a seed router and then takes the zone and network range information from the first seed interface that it detects. After a nonseed interface obtains this information, it can participate in AppleTalk routing.
- **Network Range** — The contiguous range of numbers assigned to the interface (for example, 20301 through 20310). Each router attached to the network selects a network number from within this range.
- **Address** — The AppleTalk interface address, which is based on the network range and a unique network node number (*1* through *253*) and expressed in the format *network.node*. The network number identifies the network. The node number uniquely identifies the AppleTalk node on the network. The router selects the network number from the range of numbers assigned to the network; then selects an available node number. Sample interface address: *20301.7.*
- **Zone List**— The zone or zones to which the interface belongs. You specify the default zone name and up to 15 additional zones; for a maximum of 16 zones per interface.
- **State** — The status of the AppleTalk interface, which indicates whether the interface is available (*enabled*) or unavailable (*down*).
- **VLAN interface index (VLAN index)** — The VLAN that is associated with the AppleTalk interface. When the system prompts you for a VLAN interface index, it indicates the available VLANs that you can associate with a new AppleTalk interface. For information on creating VLANs, see Chapter 9.

Important Considerations

Before configuring AppleTalk interfaces, review the following guidelines and considerations:

- Your system can support up to 32 AppleTalk interfaces.
- Each seed interface supports up to 16 zones.
- Your system supports a maximum of 1 AppleTalk interface per VLAN; overlapping AppleTalk interfaces on a bridge VLAN is not allowed.
- A seed router interface will maintain its configuration (local zone and local network information) even if the information conflicts with other routers on the same network.
- The network range is a contiguous range of numbers between 1 and 65,279.
- The network node number that a router dynamically assigns to itself is a value between 1 and 253, inclusive.
- Node numbers 0, 254, and 255 are reserved by the AppleTalk protocol.
- The maximum number of active AppleTalk devices on a network is equal to the number of network numbers multiplied by the number of possible node numbers.
- All seed routers on a network must have the same value for both the start and end of the network number range. For example, if you have a segment to which multiple routers are attached and you have assigned a network range of 4–9, then all seed router ports attached to the segment must be configured with a network range of 4–9.
- All seed routers on a network must be configured with the same zone names. For example, if you have a segment to which multiple routers are attached and you have assigned the zone names *Sales* and *Marketing* to the segment, then all seed routers attached to the segment must be configured with zone names *Sales* and *Marketing*.
- A router will not advertise its routing table through an interface until that interface has an associated network number range.
- An interface is not added to the routing table until it has an associated network number range.

Changing the zone association for an existing network number involves the deletion of the existing zone association for that network from all routers on the segment. For details, see "Changing Zone Names" later in this chapter.

AppleTalk Routes

Your system maintains a table of local and remote routes to all reachable AppleTalk networks. The Routing Table Maintenance Protocol (RTMP) automatically generates the routing table. RTMP defines rules for:

- **Information contained within each routing table** — Routers use the information within this table to determine how to forward data on the basis of its destination network number.
- **Exchanging information between routers so that the routers can maintain their routing tables** — All AppleTalk routers periodically exchange routing tables by broadcasting RTMP packets onto the network every 10 seconds; each packet containing a router's routing table entries. When a router receives the routing table of another router, it compares its own table to the one it received, then updates its table with the shortest path to each destination network.

Each routing table entry contains the following information:

- **Network Range** — A range of 16 bit numbers that identifies a network. Each device on the network selects from this range the network number it will use to identify itself on the network.
- **Distance** — Number of hops to the destination network.
- **Interface** — Interface used to reach the destination network.
- **State** — Status (*good, suspect, bad,* or r*eally bad*) of each route.
- **Next Hop** — The next-hop Internet router to which the packet must be sent.

Important Considerations

Before administering AppleTalk routes, review the following guidelines and considerations:

- The RTMP table supports a maximum of 514 entries.
- AppleTalk supports a maximum distance of 15 hops.
- A hop count of 0 represents a network directly connected to a router.
- When an AppleTalk router starts up on the network, the first entries in its routing table are the network numbers to which it is directly attached.
- Node numbers are dynamically assigned and often change when the router restarts.
- Each 16 bit number within a network range is capable of supporting 253 network nodes.

- When a router receives an RTMP packet that contains a routing entry currently not in it's table, the router adds the entry to its routing table, and increments the route's distance (hop count) by 1.
- When a network is removed from the RTMP table (whether manually, or though the aging process), the router also scans the Zone Information Table (ZIT), and removes ZIT entries that contain the deleted network number.
- If the zone Information Table contains an entry whose network number range is not in the RTMP table, it then concludes that the network is no longer on the Internet, and deletes the network's ZIT entry.
- An overburdened network with many routers can prevent some routers from sending RTMP updates every 10 seconds. Because routers begin to age out routes after the loss of 2 successive RTMP updates, the failure for RTMP packets to arrive may result in unnecessary route changes, known as route flapping. For this reason, network segments should be kept to a reasonable size.

AppleTalk Address Resolution Protocol (AARP) Cache

The AppleTalk Address Resolution Protocol (AARP) maps the hardware address of an AppleTalk node to an AppleTalk protocol address. AARP maps for both extended and nonextended networks.

Your system uses AppleTalk Address Resolution Protocol (AARP) to map hardware addresses to AppleTalk protocol addresses. AppleTalk protocol uses dynamically assigned 24-bit addresses.

AppleTalk addresses are 24 bits long and consist of a 16-bit network number and a unique 8-bit node number. AppleTalk networks support a hierarchal addressing scheme in the form of a network range, with each 16-bit network number within that range capable of supporting up to 254 nodes.

All AppleTalk nodes, including router interfaces, dynamically acquire a unique AppleTalk address using a feature provided by the AppleTalk Address Resolution protocol, called Probe.

When a node on the network initializes, it randomly selects an AppleTalk address for itself. At the same time, the node sends 10 AARP probe packets. The probe packets determine whether any other nodes on the network are using the selected address. If the address already exists, the initializing node randomly selects another address and sends another set of probe packets.

The AARP maintains an Address Mapping Table (AMT) with the most recently used hardware addresses and their corresponding AARP addresses. If an address is not in this table, the router broadcasts AARP requests to all other AppleTalk nodes on the link to determine the MAC address mapping for the specified AARP address. It then creates a corresponding AMT entry to reflect the new mapping when the destination node replies. You view this table, called the *AARP cache*, through the Administration Console.

AARP uses an Address Mapping Table (AMT), which contains the most recently used addresses. If an address is not in the AMT, the system sends an AARP request to the designated protocol address and then adds the node's destination hardware address to the table when the node replies.

AARP also registers a node's dynamically assigned address on the network, as follows:

- AARP randomly assigns an address.
- To determine whether another node is already using the address, the system broadcasts AARP probe packets containing the address.
  - If the system receives no reply, the address becomes the node's address.
  - If the system receives a reply, it repeats the process until it discovers an available address.

AARP entries include the following information:

- **AARP Address** — AARP address of the node in *network.node* format
- **MAC Address** — MAC layer address of the node
- **Interface** — Interface through which the node can be reached
- **Age** — Number of seconds before the system ages out the cache entry

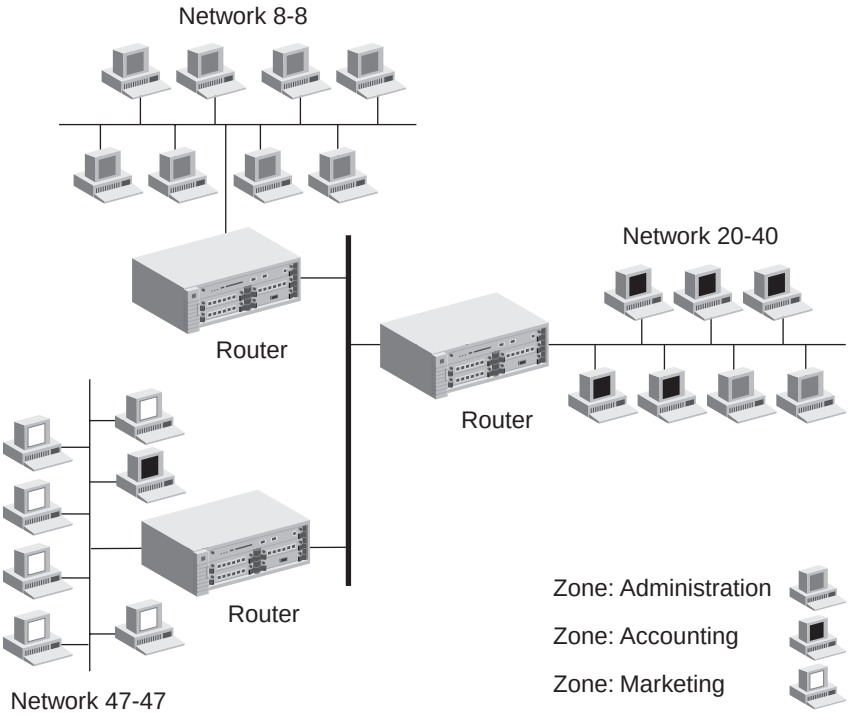
If there is no space available in the AARP cache for a new entry, the least-recently-used entry is purged to make room for the new entry.

AppleTalk Zones

An AppleTalk zone is a logical collection of nodes on an AppleTalk intranet. A zone can include all nodes in a single network or a collection of nodes in different networks. You assign a unique name to each zone to identify it in the intranet.

Figure 81 illustrates the relationship between physical AppleTalk networks and logical AppleTalk zones.

Figure 81 AppleTalk Networks and Zones



This example shows an AppleTalk intranet with three subnetworks: 47-47, 20-40, and 8-8. Three AppleTalk zones span these networks: Administration, Accounting, and Marketing. Network 20-40 includes two nodes in the Administration zone and five nodes in the Accounting zone. Network 47-47 includes a node from the Accounting zone and all nodes in the Marketing zone. Network 8-8 consists of nodes in the Administration zone only.

AppleTalk routers use the Zone Information Protocol (ZIP) to map network numbers to Zones. Each AppleTalk router maintains a Zone Information Table (ZIT), which lists the zone-to-network mapping information.

Creating zones within a network reduces the amount of searching that a router must do to find a resource on the network. For example, to gain access to a printer on the network, instead of searching the whole network when you want to print a file to a certain printer, the router searches for it within a particular zone. You gain access to the printer more quickly within the zone because the zone includes fewer devices than the entire intranet.

Important Considerations

Before administering zones, review the following guidelines and considerations:

- Whenever a router discovers a new network, it adds the network to its RTMP table. It then creates a corresponding ZIT entry with a zone list of NIL. The ZIP process then requests from the originating router the corresponding Zones associated with the newly discovered network. When it receives the associated zones, it then updates the ZIT entry.
- If the Zone information table contains an entry whose network number range is not in the RTMP table, it then concludes that the network is no longer on the Internet, and deletes the networks ZIT entry. This means, whenever a network is removed from the RTMP table (whether manually, or though the aging process), the router also removes ZIT entries that contain the deleted network number.
- At the time of initialization, the zone information table contains an entry for each seed interface directly connected to an AppleTalk network.
- On a stable AppleTalk network, the ZIP process only occurs when a new router (new network number) is introduced to the network. ZIP traffic during any other time can be an indication of network instability.
- Whenever a network is aged-out and removed from the routing table, the corresponding zone information for that network is removed from the router's zone information table.
- Assign zone names for the convenience of end users.

Changing Zone Names

When you change the zone information for a network, all routers on the segment must update their zone information tables with the new information. Although no AppleTalk mechanism forces routers to update zone lists, you can successfully change the zones associated with a network segment by:

- **Aging out the network range** — Use this method to change the zone information for a network segment without changing its existing network range.
- **Changing the network range** — Use this method to propagate a new network range throughout the network with new zone information.

You change the network range and zone information for an AppleTalk seed interface by using the `appletalk interface modify` *command. For more information on this command, see the* Command Reference Guide*. For more information about AppleTalk interfaces, see "AppleTalk Interfaces" earlier in this chapter.*

Aging Out the Network Range

If you want to change the zone information for a segment and retain the existing network range, you must age out the range from all routers on the network. This ensures that all routers query for the new zone information. This is because after a zone has been acquired, routers do not query for zone information until the network has been aged out of their routing tables.

If you do not age out the network range, some routers may not remove the network from their routing tables. Devices attached to these networks will then be unaware of the new zone information. This can result in some users seeing the new zones in their Choosers, while others see the old zones.

To age out the network range, you must prevent routers on the network from sending RTMP messages containing the network range for a minimum amount of time. This known as the ZIP bringback time. ZIP defines a bringback time of 10 minutes.

During this time, all AppleTalk interfaces on the segment are brought down, and cannot send or receive RTMP packets to confirm the existence of the network in their RTMP tables. The unconfirmed network ranges are then aged out of their routing tables; the associated zone information for the network range is removed as well.

To change the associated zones for a network segment without changing the segment's network range:

1 For any seed interfaces on the segment, use the `appletalk interface modify` command to enter the new zone list for the existing network range. When prompted, enter the number of minutes to bring the seed interface down, so that the interface does not send out RTMP updates. ZIP defines a minimum bringback time of 10 minutes.

Although ZIP defines a minimum down time of 10 minutes, the exact time required to ensure that the network range is aged from all routers depends on the complexity and size of the network to which your AppleTalk segment is attached.

2 For any nonseed interfaces attached to the segment for which you are changing zone information, remove the interface, wait a minimum of 10 minutes (or a period of time appropriate to the size and complexity of your network), then redefine the interface.

Changing the Network Range

This method involves assigning a new network range to the segment for which you are changing zone information. This forces all routers on the segment to query for the zone information when they receive the new network range. The old information is removed from the ZIT when the old network range is aged out of the RTMP tables.

To change the zone information for a network by assigning a new network range to the segment, do the following:

1 Reconfigure any seed router interfaces connected to the segment with the new network range and zone information. Because you are specifying a new network range, you do not have to bring the seed interfaces down.

2 For any nonseed interfaces attached to the segment for which you are changing zone information, remove the interface, wait a minimum of 10 minutes (or a period of time appropriate to the size and complexity of your network), then redefine the interface.

Although ZIP defines a minimum down time of 10 minutes, the exact time required to ensure that the network range is aged from all routers depends on the complexity and size of the network to which your AppleTalk segment is attached.

The new network range is propagated throughout the network forcing all routers to query for the new network's zone information.

Forwarding AppleTalk Traffic

You can choose to enable or disable AppleTalk forwarding on your system.

Enabling Forwarding

When you enable AppleTalk forwarding, you enable the forwarding of Datagram Delivery Protocol (DDP) packets. Because AppleTalk uses this network layer protocol, this also enables the routing of AppleTalk packets. You enable routing of AppleTalk traffic on a system-wide basis. This means all AppleTalk interfaces defined on the system forward routable AppleTalk traffic. All non-routable protocols, or protocols not yet configured for routing, are dropped.

Disabling Forwarding

When you disable AppleTalk forwarding, you disable the forwarding of Datagram Delivery Protocol (DDP) packets. Because AppleTalk uses this network layer protocol, this also disables the routing of AppleTalk packets. You disable routing of AppleTalk traffic on a system-wide basis. This means all AppleTalk interfaces defined on the system will not forward routable AppleTalk traffic. All AppleTalk traffic is dropped. In addition, all traffic from non-routable protocols, or protocols not yet configured for routing, are dropped.

Important Considerations

Consider the following when enabling or disabling AppleTalk forwarding:

- AppleTalk forwarding is `disabled` by default.
- Requiring you to specifically enable AppleTalk forwarding system-wide allows you to:
  - Verify that you have correctly set all necessary AppleTalk configuration parameters before activating AppleTalk routing.
  - Age network ranges from routing tables to facilitate the changing of network zone information, as described in the previous section.

Checksum Error Detection

You can enable or disable checksum generation and verification. The AppleTalk protocol uses checksums to detect errors in data transmissions. A *checksum* totals all data bytes and adds the sum to the checksum field of the data packet. The receiving station computes a verification checksum from the incoming data and compares the new checksum with the value sent with the data. If the values do not match, the transmission contains an error.

Important Considerations

Before configuring checksum error detection, review the following guidelines and considerations:

- By default, checksum generation and verification is *disabled*.
- *Disabled* is the preferred setting. Enabling the checksum generation or verification significantly impacts the router's performance.
- You may want to disable checksum generation and verification if you have older devices that cannot receive packets with checksums.

AppleTalk Echo Protocol (AEP)

The system supports the AppleTalk Echo Protocol (AEP), which sends a datagram (an Echo Request) to a specified node. The destination node returns, or *echoes*, the datagram to the sender (using an Echo Reply). This process allows you to determine whether a node is accessible. Your system's `appletalk ping` command is equivalent to an IP ping, except that you specify an AppleTalk address instead of an IP address. You can use this command to verify whether or not an AppleTalk node is reachable from the router.

AppleTalk Statistics

You can view statistics for the following AppleTalk protocols:

- Datagram Delivery Protocol
- Routing Table Maintenance Protocol
- Zone Information Protocol
- Name Binding Protocol

Datagram Delivery Protocol

AppleTalk extends the normal node-to-node delivery of packets to a process-to-process delivery. The processes running on AppleTalk nodes exchange data packets through logical sockets assigned by the Datagram Delivery Protocol (DDP). DDP provides a best-effort, socket-to-socket delivery of datagrams — packets exchanged using DDP — over the AppleTalk network.

Datagram delivery is the key service on which other AppleTalk services are built. All other AppleTalk services, such as RTMP, NBP, and ZIP rely on DDP for packet delivery, as illustrated in Figure 79 earlier in this chapter.

Your system allows you to view a variety of DDP statistics, including:

- **inBcastErrors** — Number of dropped DDP datagrams for which the system was not their final destination and they were sent to the broadcast MAC address
- **inCsumErrors** — Number of DDP datagrams that were dropped because of a checksum error
- **inDiscards** — Number of DDP Datagrams that were discarded during routing
- **inForwards** — Total number of packets that were forwarded, including those with errors
- **inLocals** — Number of DDP datagrams for which an attempt was made to forward them to their final destination
- **inNoClients** — Number of DDP datagrams that were dropped for unknown DDP types
- **inNoRoutes** — Number of DDP datagrams that were dropped for unknown routes
- **inReceives** — Total number of packets that were received, including those with errors

- **inShortDdps** — Number of input DDP datagrams that were dropped because the system was not their final destination and their type was short DDP
- **inTooFars** — Number of input datagrams that were dropped because the system was not their final destination and their hop count would exceed 15
- **inTooLongs** — Number of input DDP datagrams that were dropped because they exceeded the maximum DDP datagram size
- **inTooShorts** — Number of input DDP datagrams that were dropped because the received data length was less than the data length specified in the DDP header, or the received data length was less than the length of the expected DDP header
- **outLocals** — Number of host-generated DDP datagrams

Routing Table Maintenance Protocol

AppleTalk uses the Routing Table Maintenance Protocol (RTMP) to build and maintain routing tables. Your system allows you to view a variety of RTMP statistics, including:

- **inDatas** — Number of good RTMP data packets that were received
- **inOtherErrs** — Number of RTMP packets received that were rejected for an error other than a version mismatch
- **inRequests** — Number of good RTMP request packets that were received
- **inVersionErrs** — Number of RTMP packets received that were rejected due to a version mismatch
- **outDatas** — Number of RTMP data packets that were sent
- **outRequests** — Number of RTMP request packets that were sent
- **routeDeletes** — Number of times that RTMP deleted a route that was aged out of the table

- **routeEqChgs** — Number of times that RTMP changed the Next Internet Router in a routing entry because the hop count advertised in a routing table was equal to the current hop count for a particular network
- **routeLessChgs** — Number of times that RTMP changed the Next Internet Router in a routing entry because the hop count advertised in a routing table was less than the current hop count for a particular network
- **routeOverflows** — Number of times that RTMP attempted to add a route to the RTMP table but failed because of lack of space

Zone Information Protocol

AppleTalk uses the Zone Information Protocol (ZIP) to maintain a mapping between networks and zone names. This network-to-zone mapping is used to facilitate the name-lookup process performed by the Name Binding Protocol. Your system allows you to view a variety of ZIP statistics, including:

- **inErrors** — Number of ZIP packets received that were rejected for any error
- **inExReplies** — Number of ZIP extended replies received
- **inGniReplies** — Number of ZIP GetNetInfo reply packets received
- **inGniRequests** — Number of ZIP GetNetInfo request packets received
- **inLocalZones** — Number of Zip GetLocalZones requests packets received
- **inObsoletes** — Number of ZIP Takedown or ZIP Bringup packets received
- **inQueries** — Number of ZIP queries received
- **inReplies** — Number of ZIP replies received
- **inZoneCons** — Number of times that a conflict has been detected between this system's zone information and another entity's zone information
- **inZoneInvs** — Number of times that this system has received a ZIP GetNetInfo reply with the zone invalid bit set because the corresponding GetNetInfo request had an invalid zone name
- **inZoneLists** — Number of Zip GetZoneLists requests packets received

- **outAddrInvs** — Number of times that this system had to broadcast a ZIP GetNetInfo reply because the GetNetInfo request had an invalid address
- **outExReplies** — Number of ZIP extended replies sent
- **outGniReplies** — Number of ZIP GetNetInfo reply packets sent out of this port
- **outGniRequests** — Number of ZIP GetNetInfo packets sent
- **outLocalZones** — Number of transmitted ZIP GetLocalZones reply packets
- **outQueries** — Number of ZIP queries sent
- **outReplies** — Number of ZIP replies sent
- **outZoneInvs** — Number of times that this system has sent a ZIP GetNetInfo reply with the zone invalid bit set in response to a GetNetInfo request with an invalid zone name
- **outZoneLists** — Number of transmitted ZIP GetZoneList reply packets

Name Binding Protocol

AppleTalk uses the Name Binding Protocol (NBP) to convert user-friendly entity names (which are user-defined and change infrequently) into AppleTalk network addresses (which are dynamically-assigned and change frequently). Your system allows you to view a variety of NBP statistics, including:

- **inBcastReqs** — Number of NBP Broadcast Requests received
- **inErrors** — Number of NBP packets received that were rejected for any error
- **inFwdReqs** — Number of NBP Forward Requests received
- **inLkupReplies** — Number of NBP Lookup Replies received
- **inLkupReqs** — Number of NBP Lookup Requests received

Standards, Protocols, and Related Reading

For more information about AppleTalk technology, see the following publications:

- Gursharan S. Sidhu, Richard F. Andrews, and Alan B. Oppenheimer, *Inside AppleTalk, Second Addition* (Addison-Wesley Publishing Company, 1990).
- RFC 1742, AppleTalk Management Information Base II

17 QoS AND RSVP

This chapter provides guidelines and other key information about how to use Quality of Service (QoS) and the Resource Reservation Protocol (RSVP) on your system.

- QoS Overview
- Key Concepts
- Key Guidelines for Implementation
- QoS Classifiers
- QoS Controls
- Examples of Classifiers and Controls
- Modifying and Removing Classifiers and Controls
- QoS Excess Tagging
- Transmit Queues and QoS Bandwidth
- LDAP
- RSVP

You can manage QoS and RSVP in either of these ways:

- *From the* `qos` *menu of the Administration Console. See the* Command Reference Guide.
- *From the Traffic Policy (QoS) folder of the Web Management Software. See the* Web Management User Guide.

QoS Overview

Quality of Service (QoS) is an advanced feature that allows you to establish control over network traffic. QoS provides *policy-based services*, which establish various grades of network service to accommodate different types of traffic, such as multimedia, video, protocol-specific, time-critical, and file-backup traffic. Although QoS and Class of Service (CoS) are closely related, QoS has more features and addresses bandwidth, delay, loss, and jitter control. (CoS focuses on differentiating traffic into classes and prioritizing those classes.)

Features

Your system supports the following QoS features:

- **QoS Classifiers** — Define how your system groups packets in order to schedule them with the appropriate service level.
- **QoS Controls** — *A*ssign rate limits and IEEE 802.1p priorities, and/or prioritize packets that are associated with one or more classifiers. Using the QoS Excess Tagging feature, you can also select an IEEE 802.1p priority for packets that exceed the control's rate limit.
- **Settable QoS Bandwidth** — Controls the weighting of high priority and best effort traffic.
- **Resource Reservation Protocol (RSVP)** — A building block of QoS that implements QoS characteristics in your LAN environment. RSVP is an end-to-end signaling IP protocol that allows an end station to request the reservation of bandwidth across the network. RSVP provides admission control. QoS can operate at Layer 2 and Layer 3; RSVP operates at Layer 3 only.

Benefits

You can use QoS on your system to provide the following benefits:

- Control a wide variety of Ethernet or FDDI network traffic by:
  - Classifying traffic based on packet attributes such as protocol type, class type (802.1p), IP address, and/or TCP/UDP socket.
  - Assigning priorities to traffic (for example, to set higher priorities to time-critical or business-critical applications).
  - Applying security policy through traffic filtering.
  - Using the connection-oriented RSVP for bandwidth reservation (reserving and policing an RSVP session to make sure the session uses only as much bandwidth as it needs).
- Provide constant delay/jitter for multimedia applications such as video conferencing or voice over IP.

- Improve performance for specific types of traffic and preserve performance as the volume of traffic grows.
- Reduce the need to constantly add bandwidth to the network.
- Manage network congestion.

Methods of Using QoS

Your system's implementation of QoS focuses on traffic classification, policy-based management, and bandwidth. It provides multiple service levels (mapped to several transmit queues), classification of traffic types, and weighted fair queueing of priority-queued traffic.

If you use QoS and simply opt to broadly classify traffic, you are using a subset of QoS called *network class of service*. To simplify your classification of traffic, the system provides a set of predefined traffic classes. You can also specify your own classes of traffic with applied controls to:

- Create a to/from classifier with address/port patterns that isolate traffic based on source and destination.
- Block traffic (for example, prevent certain traffic from one workgroup from seeing another workgroup).
- Assign priorities to traffic.

See "Examples of Classifiers and Controls" later in this chapter.

If you use QoS with RSVP, you are opting for a more complex type of end-to-end QoS that aims for a "guaranteed" quality of service. To use RSVP, you must be routing. In addition, RSVP is required at the desktop, which may present issues of desktop control and upgrade issues concerning the resident operating-system and applications.

Key Concepts

Before configuring QoS, review the following standards and terms.

Related Standards and Protocols

The system supports IEEE 802.1Q, IEEE 802.1p, and the RSVP protocol.

IEEE 802.1p

This standard, which is part of the IEEE 802.1D MAC Bridges base standard, focuses on traffic class prioritization as well as dynamic multicast filtering services in bridged LANs. It uses the same tag format as the proposed IEEE 802.1Q standard, but it uses three additional bits of the tag control information for setting a user priority level (for policy-based services such as QoS). You can classify traffic according to a specific IEEE 802.1p priority tag value (or several tag values). You can also define a control that inserts a priority tag value in forwarded frames.

The IEEE 802.1p priority tag values are 0 to 7 decimal. Table 59 shows the IEEE 802.1p (user-priority) values and the corresponding traffic types. The value 7 (Network Control) is considered the highest priority and 1 (Background Traffic) is the lowest priority. Note that the value 0 (the default, Best Effort) has a higher priority than value 2 (Standard).

Table 59 IEEE 802.1p Priority Tag Values

Tag Value	Traffic Type (order of priority)
1	Background
2	Standard (spare)
0 (the default)	Best Effort
3	Excellent Effort (Business Critical)
4	Controlled Load (Streaming Multimedia)
5	Video (Interactive Media), less than 100 milliseconds latency and jitter
6	Voice (Interactive Voice), less than 10 milliseconds latency and jitter
7	Network Control (Reserved Traffic)

The IEEE 802.1p standard addresses separate queuing of time-critical frames to reduce jitter.

The Resource Reservation Protocol (RSVP)

This connection-oriented IP protocol handles bandwidth reservation. The request for comments document RFC 2205 describes the details of RSVP.

RSVP aims to meet the demands of real-time voice and video applications by using a QoS flow specification that mandates parameters such as the maximum frame transmission rate, long-term average frame transmission rate, maximum frame jitter, and maximum end-to-end delay. It supports the QoS flow specifications by managing *resource reservations* across the network.

With RSVP, all devices in the path from the source to the destination must agree to observe the RSVP call request parameters before traffic can flow.

Terminology

The following terms apply to the QoS implementation on your system:

- **Classifiers** — Two types of classifiers define how your system groups packets in order to schedule them with the appropriate service level:
  - **Flow classifiers** — Apply to *routed* IP unicast and IP multicast traffic only (not bridged traffic). (When the system is bridging, you cannot classify to the IP address or socket level.) These classifiers are numbered in a range of from *1* to *399.* You can define filtering parameters for a flow classifier by setting the source IP address, source IP address mask, the destination IP address, destination IP address mask, and the TCP or UDP port range. Because these classifiers have lower class numbers, they take precedence over nonflow classifiers. When a packet falls into more than one controlled classifier, the system uses the lower-numbered classifier to classify the packet. The system predefines two flow classifiers for you: TELNET and FTP.
  - **Nonflow classifiers** — Apply to *both switched and routed* traffic. You define this type of classifier to handle specific link-level protocols *(IP, TCP/IP, IPX,* or *AppleTalk)*, a "cast" type *(broadcast, unicast,* or *multicast)*, and/or one or more IEEE 802.1p priority tag values. Nonflow classifiers are numbered in a range of from *400* to 499. The system automatically defines a number of nonflow classifiers for you. The predefined nonflow classifiers *(401* through *407)* employ IEEE 802.1p tagging by default for received frames. (Before you define an IP, AppleTalk, or IPX classifier, you must have an IP, AppleTalk, or IPX VLAN for the ports that will come under jurisdiction of the classifier and applied control.)

- **Controls** — *Define the following parameters to a*ssign rate limits and priorities to the packets that are associated with one or more classifiers:
  - **Rate limit** — Limits the amount of input bandwidth used by incoming classified traffic (optionally, on a per-port basis). When you define a control, you can specify one of three rate limits: *none* (no rate limit), *receivePort* (a separate limit on each specified receive port), *aggregate* (limits on groups of receive ports)
  - **Service levels** — Specify a transmit priority and map to a specific transmit queue. If you specify receivePort or aggregate for a rate limit, you can assign a service level of *high*, *best*, or *low* to both conforming packets (packets that are below the rate-limit parameters) and nonconforming excess packets (excess packets that exceed the rate-limit parameters). If you set the rate limit to none, you can specify a service level of high, best, low, or *drop* for conforming classified packets.

Drop causes the system to drop all packets on all ports associated with the control and its classifier. If you want to drop conforming packets for only a subset of ports, specify the receivePort or aggregate rate limit, set the rate limit to 0, and specify the group of ports.

  - **Loss-eligible status** — *Loss-eligible packets* are conforming packets that are discarded instead of queued when transmit queues back up beyond a threshold. You can specify whether conforming packets (as well as nonconforming excess packets) are loss eligible when you define a control. Marking packets loss eligible is useful for an intelligent discard of traffic in a congestion situation. When the system is congested, you can decide which traffic can be discarded and mark that traffic loss eligible.
  - **Burst size** — The maximum amount of data that you can transmit at the line rate before the transmission is policed. This value accommodates variations in speeds and allows you to occasionally exceed the configured rate.
  - **TCP drop control** — TCP drop control lets you create QoS Flow Classifiers that allow traffic going from "source" IP addresses to "destination" IP addresses to be dropped or otherwise controlled using one-way TCP flow filtering. This control can only be used for flow classifiers that use the TCP/IP protocol.

- **Timer option —** The QoS Timer option lets you configure a QoS session to take effect during a predefined time period by setting the start and end times for the specific control.
- **IEEE 802.1Q priority tag** — When you define a control for a classifier, you can select an IEEE 802.1p priority tag value to insert into forwarded frames. Make sure that this priority tag is applied to ports that are configured for IEEE 802.1Q tagging.

- **QoS bandwidth** — Specifies the weighting of the high priority and best effort transmit queues. The bandwidth for the control queue is set via RSVP. By default, 75 percent of the bandwidth is used for high priority traffic and 25 percent is used for best effort packets (that is, three high priority packets are sent for each best effort packet). Low priority packets do not have bandwidth allocated.
- **QoS excess tagging** — Enables you to select an IEEE 802.1p priority tag value for nonconforming excess packets (packets that exceed the rate limit). This option refers to any packets marked as excess that you want to tag. If you enable this option, you can select an IEEE 802.1p priority tag value in the range of from *0 to 7*, with *0* as the default. Specifying *1* means that nonconforming excess become background traffic. (See Table 59 earlier in this chapter.)

Key Guidelines for Implementation

Consider the following guidelines when you configure QoS on your system.

Procedural Guidelines

Configure classifiers and controls in the following order:

1. Define a classifier, or choose a predefined classifier. Identify a particular type of traffic that you want to regulate and define a classifier for this traffic via the Administration Console or the Web Management Traffic Policy Wizard. The rules for defining classifiers are different for flow versus nonflow classifiers.
2. Create a control to apply to the classifier(s) you defined. The control enables the system to direct the traffic to one of the available transmit queues or drop the traffic. As part of defining a control, you can:
   - **a** Assign a rate limit to the incoming classified traffic (optionally, on a per-port basis).
   - **b** If you specify a rate limit, define what should be done with the nonconforming excess (traffic that exceeds the rate-limit parameters).
   - **c** Apply an IEEE 802.1p priority tag value to forwarded traffic.

General Guidelines

- You must define a classifier before you can assign a control to it.
- A classifier does not affect traffic scheduling until you configure a control for that classifier.
- Traffic that is not classified and controlled is treated with a transmit priority of best (best effort) using the default classifier (499) and control (1). In this case, all packets are conforming packets.
- You cannot remove or modify the default classifier *(499)*.
- You cannot remove the default control *(1)*, but you can modify it.
- When you specify a TCP or UDP port range for a flow classifier, limit the range as much as possible (for example, to a single TCP or UDP port or a small range of ports). If the classifier applies to a wide range of TCP or UDP ports, you increase the amount of classified traffic on the system and consume valuable QoS resources (cache entries).
- If you have defined a control and you want to remove or modify the associated classifier, you must remove the control before you can remove or modify the classifier.

QoS Classifiers

You define classifiers to distinguish certain types of traffic from other types of traffic. A classifier tells the system how to identify a certain type of traffic; after defining a classifier, you must apply a control to the classifier.

Important Considerations

Review the following before configuring classifiers:

- You can classify bridged or routed traffic (such as AppleTalk or IPX) based on protocol type, cast type, and IEEE 802.1p priority. For routed IP traffic, you can also classify traffic by IP source addresses, destination addresses, or TCP or UDP sockets.
- Before you define a classifier, determine whether you can use one of the system's predefined classifiers (classifiers that come with your system). If you decide to define your own classifier, you need to decide which type of classifier to define, *flow* (IP routed traffic only) or *nonflow* (bridged or routed traffic).
- You can define up to *100* flow classifiers and up to *16 nonflow* classifiers. Because the system predefines 16 nonflow classifiers, you must delete one of the existing nonflow classifiers (except the default classifier) before you can add your own. See "Modifying and Removing Classifiers and Controls" later in this chapter for information on changing or deleting a classifier.
- When you configure a classifier, the system prompts you for different information based on your choice of defining a flow or nonflow classifier.

Using Predefined Classifiers

Figure 82 shows a QoS classifier summary from the Administration Console with the 2 predefined flow classifiers (FTP and Telnet) and 16 predefined nonflow classifiers, along with their associated controls. (You can use your configuration tool to display summary and detail information for your classifiers.)

The system provides a default classifier (499). You cannot remove or modify this classifier. *If you want to* modify one of the predefined nonflow classifiers with controls, you must remove the control first.

In Figure 82, U refers to unicast, M refers to multicast, and B refers to broadcast. Also, the range 0–7 implies that a nonflow classifier recognizes *all* IEEE 802.1p priority tags. (See Table 59 earlier in this guide.)

Figure 82 Predefined Classifiers and Associated Controls

	Classifier	Name	Control	Cast	Protocol	802.1p
Flow	20	FTP	none	UM	TCP	--
	23	Telnet Traffic	none	UM	TCP	--
Nonflow	401	Background	2	UMB	any	1
	402	Standard	2	UMB	any	2
	403	Business Critical	3	UMB	any	3
	404	Streaming Multimedia	4	UMB	any	4
	405	Interactive Multimedia	4	UMB	any	5
	406	Interactive Voice	4	UMB	any	6
	407	Network Control	4	UMB	any	7
	420	TCP/IP	none	U	TCP/IP	0-7
	430	IP Unicast	none	U	IP	0-7
	440	IP Multicast	none	M	IP	0-7
	450	IP Broadcast	none	B	IP	0-7
	460	IPX Unicast	none	U	IPX	0-7
	470	IPX Multicast/Broadcast	none	MB	IPX	0-7
	480	Appletalk Unicast	none	U	Appletalk	0-7
	490	Appletalk Multicast/Broadcast	none	MB	Appletalk	0-7
	499	Default	1	UMB	any	0-7

Assigning Flow and Nonflow Classifier Numbers

Each classifier requires a unique number in the range 1 to 498. When you define a classifier, the first information you supply is the classifier number. The number you specify dictates which type of classifier you are defining.

Default

The default classifier number is *499,* which you cannot remove or modify. This is because all traffic passes through the QoS engine and the system needs a classifier to handle all packets.

- If you want to define a flow classifier (for routed IP packets only), specify a value in the range of from *1* to *399*. This allows you to specify IP source and/or destination addresses as well as TCP or UDP socket information.
- If you want to define a nonflow classifier (for bridged or routed packets), specify a value in the range of from *400* to *498*. (See the list of predefined nonflow classifiers in Figure 82.) For nonflow classifiers, you cannot classify to the IP address or socket level.

The classifier number indicates precedence. The classifier with the *lowest* number takes precedence if a packet meets the criteria for more than one classifier.

For example, you might use two classifiers as follows:

- You define a flow classifier with classifier number 6 that recognizes all TCP or UDP traffic from IP address 3.3.3.3. The control you assign to this classifier (control 5) gives this traffic a *low* priority service level.
- You use the predefined nonflow classifier 420, which recognizes all TCP traffic, and create a control for this classifier to give the TCP traffic a *high* priority service level. (By default, this classifier has no control.)

With these classifiers in place, if 3.3.3.3 sends TCP traffic, this traffic receives *low* priority, since classifier number 6 is lower than classifier 420 and has a higher precedence. Table 60 shows the basic information for these classifiers.

Table 60 Classifier Number Precedence for Two Classifiers

Classifier	Name	Control	Cast	Protocol	802.1p
6 (user-defined)	from_3.3.3.3	5 (for low priority)	UM	all (TCP, UDP)	–
420 (predefined)	TCP/IP	6 (for high priority)	U	TCP/IP	0-7

Defining Flow Classifiers

You can define up to 100 flow classifiers per system for routed IP traffic. When you define a flow classifier (using a unique classifier number), you can create one or more address/port patterns (filters) for that classifier.

Each address/port pattern counts toward the flow classifier limit. Therefore, if you define a flow classifier with 10 address/port patterns, you can have up to 90 additional flow classifiers.

Because a flow classifier handles IP routed traffic only, it is expected that you have an IP VLAN, an IP routing interface, and IP routing enabled.

Flow Classifier Information

You supply the following information when defining a flow classifier:

- A classifier number in the range 1 to 399 (20 and 23 are predefined)
- A classifier name (a unique name of up to 32 characters long)
- A cast type (unicast, multicast, or both). If you create a classifier to block all IP unicast traffic, the system will block TCP and UDP unicast traffic only, not ICMP traffic.
- The IP protocol type (TCP, UDP, or all)
- Source IP address (in standard dot notation, such as 192.101.10.0)
- Source IP address mask (not a subnet mask; see "Specifying Addresses and Address Masks")
- Destination IP address
- Destination IP address mask
- Start and end of a TCP or UDP source port range as a number from 0 to 65535
- Start and end of a TCP or UDP destination port range as a number from 0 to 65535
- Whether you want to define another address/port pattern (filter) for this classifier

Specifying Addresses and Address Masks

You can classify traffic using source and destination IP addresses and their associated source and destination IP address masks. For a classifier aimed at filtering traffic to a specific destination from a particular source, you could define a single address/port pattern that specifies the source address and the destination address. Alternatively, if classified traffic to and from certain locations is going to be controlled at the same service level, you may decide to use two address/port patterns, one that covers IP address A as the source and IP address B as the destination, and a second pattern that covers IP address B as the source and IP address A as the destination.

You specify a source or destination IP address in standard dot notation, such as 192.101.10.0. An address of all zeroes is a wildcard match for any source or destination address. You can use the 0 as a wildcard in any portion of the address.

For the source or destination IP address mask, you specify how many parts of the IP address you want to match. Place a 255 in each portion of the mask that you want the software to recognize; place a 0 in any portion of the mask that you want the software to ignore.

The following examples show different ways of specifying IP addresses and IP address masks:

- An IP address of 192.101.20.254 with a mask of 255.255.255.255 requests an exact match for the host IP address 192.101.20.254.
- An IP address of 192.101.20.0 with a mask of 255.255.255.0 requests a match for any node on the subnet 192.101.20.0.
- An IP address of 192.101.20.40 with a mask of 255.255.0.0 requests a match for any node on the 192.101.0.0 network.
- A destination IP address of 0.0.0.254 (or 192.101.20. 254) with a mask of 0.0.0.255 requests a match on any node that ends in 254.
- A mask of 0.0.0.0 is a wildcard match.

Specifying Ports and Port Ranges

Many common applications are associated with well-known port numbers. For example, FTP (which uses TCP) uses ports 20 and 21, Telnet (which also uses TCP) uses port 23, and SNMP (which uses UDP) uses port 161. You can consult the services database file typically associated with TCP/IP hosts for a list of the well-known applications (services) and port numbers. For other applications, you may have to determine the appropriate port number.

When you specify the start and end range of a TCP or UDP port, specify as small as range as possible, such as 1 port (for example, port 2049 as both the start range and the end range). Also if possible, apply this small range to both source and destination port ranges. If the classifier applies to a wide range of TCP or UDP ports, you increase the amount of classified traffic on the system and consume valuable QoS resources.

To define flow classifiers and their associated controls for specific scenarios, see "Examples of Classifiers and Controls" later in this chapter.

Defining Nonflow Classifiers

Nonflow classifiers enable you to classify bridged or routed frames according to protocol, cast type, and/or IEEE 802.1p priority tag values. You can define up to 16 nonflow classifiers per system. The system predefines 16 nonflow classifiers for you. Therefore, if you want to define your own nonflow classifier, you must first delete one of the predefined nonflow classifiers.

NonFlow Classifier Information

You supply the following information when defining a nonflow classifier:

- A classifier number in the range 400 to 498 (401 to 407, 420, 430, 440, 450, 460, 470, 480, and 490 are predefined)
- A classifier name (a unique name of up to 32 characters long)
- A cast type (unicast, multicast, broadcast, or all).
- A protocol type (TCP/IP, IP, IPX, AppleTalk, any or custom)

  If you choose `custom,` select the protocol type (ethernet or DSAP/SSAP).

  - For Ethernet type enter the hexidecimal value
  - For DSAP/SSAP type enter the DSAP and SSAP hexidecimal values
- An IEEE 802.1p tag value in the range 0 to 7 or all. You can tell the system to recognize any IEEE 802.1p tagged frames with any combination of the priority tags in the range 0 to 7. The tag value is automatically used by the associated control when forwarding frames. See "IEEE 802.1p" earlier in this chapter for more information on the tag values.

For example, you may create a nonflow classifier for your bridged AppleTalk traffic, assign it a cast type of broadcast, a protocol type of AppleTalk, and an IEEE 802.1p tag value of all. You can then apply a control to this classifier to assign a rate limit and/or service level, and/or IEEE 802.1p tag to apply to forwarded frames. The classifier/control begins to work only if you have a VLAN for the Appletalk traffic or an unspecified protocol VLAN, such as the default VLAN.

For examples of how to define nonflow classifiers and their associated controls for specific scenarios, see "Examples of Classifiers and Controls" later in this chapter.

QoS Controls

After you define a classifier, you assign it a control to apply one or more of the following:

- A rate limit (to limit the amount of input bandwidth the classifier uses)
- A service level for conforming packets (a transmit priority that maps to a particular transmit queue)
- Whether packets conforming to the rate limit are loss eligible (that is, discarded instead of queued when transmit queues back up beyond a threshold)
- An IEEE 802.1p priority tag value to apply to forwarded frames
- A one-way filter to drop packets used to establish TCP connections
- A time range that the QoS control has on the classified traffic

Important Considerations

Review the following before configuring controls:

- The system predefines controls 1 through 4 for some of the predefined nonflow classifiers. You can also modify one of these predefined controls.
- There are several ways to create controls for classifiers. You can:
  - Apply one control to only one classifier.
  - Apply one control to multiple classifiers.
  - Assign a rate limit of none to a control and thereby emphasize the service level and priority tag.
  - Assign a rate limit type of receivePort or aggregate to the control and define multiple rate-limit values for different subsets of ports.
- Each classifier can have only *one* control. Therefore, although you can apply a control to a classifier that has multiple rate-limit values for subsets of ports, that control can have only one priority specification (for forwarded frames). You would have to use multiple classifiers to use different priority levels.

For examples of how controls can be applied to classifiers, see "Examples of Classifiers and Controls" later in this chapter. For information on modifying or removing controls, see "Modifying and Removing Classifiers and Controls" later in this chapter.

Assigning Control Numbers

Each control must have a unique control number. When you define a control, the system provides the next-available control number, but you can specify any unreserved control number.

The system supports control numbers in the range 1 to 50 and predefines controls 1 through 4 for some of the predefined nonflow classifiers. Control 1 is associated with the default classifier and can be modified but not removed. You can modify the other predefined controls as well (2 through 4). For example, may want to redefine the way Business Critical traffic is handled by modifying predefined control 3.

Table 61 lists the predefined controls.

Table 61 Predefined Controls

Control Number/Name	Service Level	Classifiers Controlled	Other Characteristics
1 Default/Best Effort	best	499 (default)	No rate limit, not loss eligible, no priority
2 Background	low	401, 402	No rate limit, not loss eligible, no priority
3 Business Critical	best	403	No rate limit, not loss eligible, no priority
4 Controlled Load	high	404, 405, 406, 407	No rate limit, not loss eligible, no priority

Use your configuration tool (such as the Administration Console) to display summary and detail information for your controls.

When you define a control, you supply the following information:

- A control number in the range 5 to 50 (unless you remove the predefined controls from prefined classifiers)
- A control name (a unique name of up to 32 characters long)
- The rate-limit type for the control (none, receivePort, or aggregate)
- A service level (transmit priority) for conforming packets.
- Whether the conforming packets are loss eligible. The default is no.

- For the rate limit type receivePort or aggregate, the following:
  - Service level for nonconforming excess (packets exceeding the rate limit)
  - Whether nonconforming excess are loss eligible. The default is yes.
  - How the rate limit is expressed (percentage of port bandwidth or KBytes/sec)
  - Rate-limit value (0 to 65434 Kbytes or 0 to 100 percent).
  - A burst size in KBytes (*16* through *8192*, with the default value dependent on your specified rate limit).
  - Bridge ports for which you want to enable the specified rate limit value (specific bridge ports or all bridge ports). If you specify a subset of available ports, you can enter another rate-limit value for another set of ports.
- For any type of rate limit (and a service level other than drop), any combination of IEEE 802.1p priority tag values in the range *0* to *7* or *none* to apply to forwarded frames. By default, no tags are applied, unless the associated classifier defines a tag value. In that case, the tag value from the associated classifier is used for the forwarded frames.
- Whether to drop packets used to establish TCP connections. This is a form of one-way filtering for flow classifiers only. The default is no.
- Enable control start and stop times. Similar to how a VCR operates, this timer allows you to set the desired beginning and ending period for a control. The default is no.

  If you select yes, you set the following:
  - Input time type such as daily, weekdays, or weekends. You can also choose a "specific" type that lets you choose an exact day.
  - One or more classifiers (classifier numbers) that are subject to this control.

Specifying Rate Limits

A rate limit restricts the amount of input bandwidth used by incoming classified traffic (optionally, on a per-port basis). When you define a control, you can specify one of three rate limits:

- **None** — No rate limit
- **ReceivePort** — Imposes a separate limit on each receive port
- **Aggregate** — Imposes limits on groups of receive ports. This rate limit type can only be applied to flow classifiers.

Your choice of rate limit determines how much additional information you need to supply. The default rate limit is *none*. If you specify a rate limit of none, there is no rate limit applied to the classifier. With this rate limit type, you then have a small subset of options to specify. You select a service level and loss-eligibility status for conforming packets (packets that are below the rate limit), decide if you want to apply an IEEE 802.1p priority tag value to forwarded frames (for service levels other than drop), and specify the classifiers you want to associate with the control.

If you specify a rate limit of *receivePort* or *aggregate*, you have many additional options. After you specify a service level and loss-eligibility status for conforming packets, you can also specify a service level for nonconforming excess packets (packets that exceed the specified rate limit), whether the nonconforming excess are loss eligible, how the rate limit for receive ports should be expressed, the rate-limit value, a burst size, and the receive ports for which you want to enable the rate limit. (The rate limit sets a bandwidth limit for a specific set of ports. You can specify multiple rate-limit values for different subsets of ports. As with any rate limit type, you can additionally specify an IEEE 802.1p priority tag value on forwarded frames.)

When you specify how a receivePort or aggregate rate limit is expressed, you can select a percentage of port bandwidth or KBytes/sec:

- For KBytes/sec as a rate limit (the default), specify the value for the rate limit in KBytes/sec (*0* through *65434*).
- For a percentage for the rate limit, specify the percentage in the range of from *0* to *100* percent. These numbers are rounded to the nearest 16 KBytes. A value of *0* makes all packets nonconforming excess packets. The system drops these packets only if the service level for excess packets is set to *drop*.

After specifying how the rate limit is expressed, you can specify a burst size. The *burst size* is the maximum amount of data that you can transmit at the line rate before the transmission is policed. This value accommodates variations in speeds and allows you to occasionally exceed the configured rate.

Specifying Service Levels

When you define a control, you specify a service level (a transmit priority). Most of the service levels you can specify represent a specific transmit queue. You can assign service levels to conforming packets (packets that are within the rate limit) and nonconforming excess packets (packets that exceed the rate limit).

For information on assigning an IEEE 802.1p priority to nonconforming excess packets, see "QoS Excess Tagging" later in this chapter. For information on the transmit queues and QoS bandwidth, see "Transmit Queues and QoS Bandwidth" later in this chapter.

Service levels also define the loss-eligibility status for conforming and nonconforming excess. By default, conforming packets are *not* loss-eligible; nonconforming excess *are* loss-eligible.

The system supports these service levels:

- **High** — For any type of rate limit, transmits the packet first (top priority).
- **Best** — For any type of rate limit, transmits the packet on a best-effort basis (the default for conforming and nonconforming excess packets).
- **Low** — For any type of rate limit, transmits the packet on a low-priority basis.
- **Drop** — For a rate limit of none only, drops *all* conforming packets on *all* ports associated with the classifier(s). For a rate limit of receivePort or aggregate, drops all nonconforming excess packets.

  If you want to drop conforming packets for only a subset of ports, use the receivePort or aggregate rate limit, set the rate limit to 0, and specify the group of ports.

  If you specify drop for the service level for conforming packets (you are using a rate limit of none), the system does not give you the option of specifying an IEEE 802.1p tag.

Specifying TCP Drop Control

The TCP drop control option lets you create a control for packets used to establish TCP connections. This control affects QoS Flow Classifiers that have TCP traffic going from "source" IP addresses to "destination" IP addresses.

TCP drop control does not function with nonflow classifiers or UDP. It is only available for flow classifiers that include TCP.

Figure 83 illustrates how TCP handshaking works between the source and destination to establish a connection. By dropping only the *initial* TCP packet used to establish TCP connections (those packets containing a signature of SYN=1, ACK=0), you can establish one-way TCP flow filtering.

Figure 83 3-way TCP Handshake

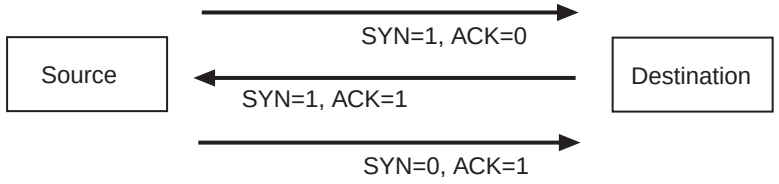
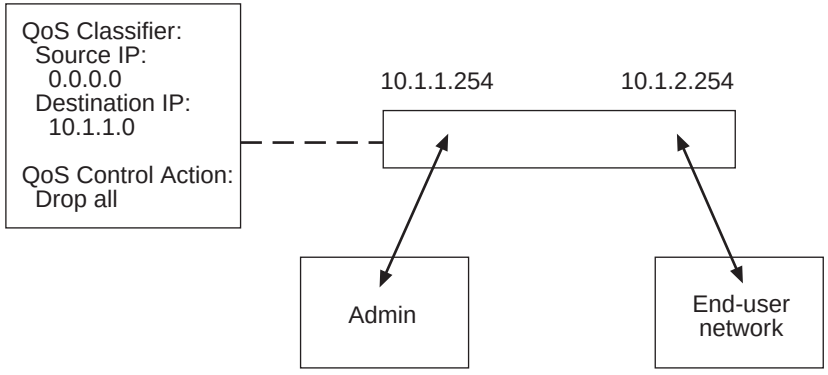


Figure 84 shows an example with TCP drop control disabled.

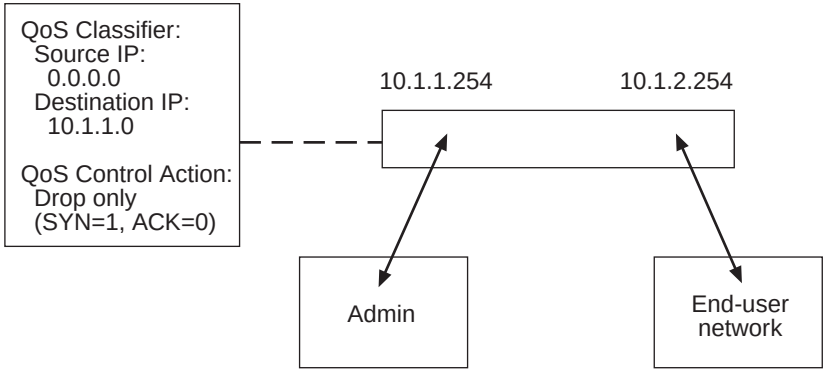
Figure 84 QoS Control Action (Drop Control Disabled)



With the QoS Classifier and QoS Control definition shown in Figure 84 (TCP control is not enabled), any attempt by a client on the End-user network to establish a TCP connection to a server on the Admin network fails.

This next example illustrates how TCP one-way-filtering can be effective. Figure 85 shows the same situation, but with TCP drop control enabled to filter only those packets with the SYN=1 and ACK=0 signature.

Figure 85 QoS Control Action (Drop Control Enabled)



In this example, any attempt by a client on an End User network to establish a TCP connection to a server on the Admin network still fails, but it is now possible for clients on the Admin network to establish TCP connections to servers on any network without restriction.

Setting the QoS Timer Control

The QoS Timer option lets you configure a QoS session to take effect during a predefined time period by setting the start and end times for the specific control. Setting the start and end times is similar to using a VCR to record programs.

The default setting for the timer control is no (no timer control). QoS controlled classifiers are in effect all the time when timer control is not enabled.

- Starting and ending days use the following syntax: `mm–dd`
  For example, to enter a date of May 20 you would enter: 05–20
- Starting and ending times use the following syntax: `hh:mm`
  For example, to enter a time of 10 o'clock in the morning you would enter: 10:00

- Days of the week use the following syntax: `1-7` (Monday=1, Tuesday=2, Wednesday=3, Thursday=4, Friday=5, Saturday=6, Sunday=7). For example, to enter Monday as the day of the week, you would type: 1
- You can check the timer control options using the "qos control detail" command. The detail displays the type of time control, the start and end times, and the classifiers associated with the control.

The time is checked every minute.

Timer Options

The following options are available for the timer control:

- **Specific Day** — Select the specific start day and time, and the specific end day and time. The control is removed once the end time is reached.
- **Daily** — Select a starting day and then a start and end time. The control is activated between the start and end time everyday.
- **Day of the Week** — Select a day and then a start and end time. The control is removed once the end time is reached.
- **Every Day of the Week** — Select a start day and then the start time and end time. The control is activated between the start and end times every 7 days.
- **Every Day This Week** — Select start and end time. The control is activated every weekday between the start and end times for the current week.
- **Every Day This Weekend** — Select a start and end time. The control is activated during each day of the current weekend and is removed once the Sunday end time is reached.
- **Every Week Day** — Select a start and end time. The control is activated between the start and end times every weekday.
- **Every Weekend** — Select a start and end time. The control is activated on each weekend day between the start and end times.

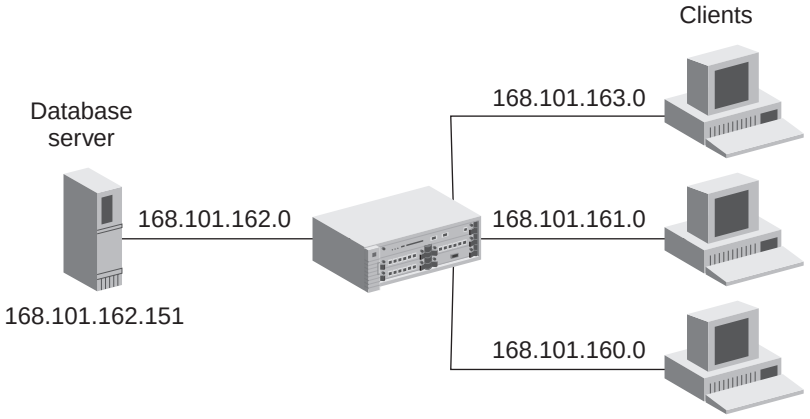
Examples of Classifiers and Controls

The following six examples show different ways to implement flow and nonflow classifiers and their associated controls.

Example 1: Traffic to and from a Specific Server

In the first example, a flow classifier is defined with two address and port patterns (filters) to classify traffic from subnets of the 168.101.0.0 network *to* the database server 168.101.162.151, and traffic *from* the server to the subnets. This kind of configuration could be called a to/from classifier. The control applied to this classifier gives the traffic to and from the server high priority.

Figure 86 Flow Classifier for Traffic to/from a Specific Server



To/from classifier definition with two address and port patterns:

Classifier Field	Classifier Definition
Classifier Number	15
Classifier Name	*DBServer1*
Cast Type	*unicast*
IP protocol type	UDP
Source IP address	*168.101.0.0*
Source IP address mask	255.255.0.0
Destination IP address	168.101.162.151
Destination IP address mask	255.255.255.255
UDP source port range (start)	0

Classifier Field	Classifier Definition
UDP source port range (end)	65535
UDP destination port range (start)	2020
UDP destination port range (end)	2020
Add another filter (address/port pattern)?	y
Source IP address	168.101.0.0
Source IP address mask	255.255.0.0
Destination IP address	168.101.162.151
Destination IP address mask	255.255.255.255
UDP source port range (start)	2020
UDP source port range (end)	2020
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another filter (address/port pattern)?	y
Source IP address	168.101.162.151
Source IP address mask	255.255.255.255
Destination IP address	168.101.0.0
Destination IP address mask	255.255.0.0
UDP source port range (start)	0
UDP source port range (end)	65535
UDP destination port range (start)	2020
UDP destination port range (end)	2020
Add another filter (address/port pattern)?	y
Source IP address	168.101.162.151
Source IP address mask	255.255.255.255
Destination IP address	168.101.0.0
Destination IP address mask	255.255.0.0
UDP source port range (start)	2020
UDP source port range (end)	2020
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another filter (address/port pattern)?	n

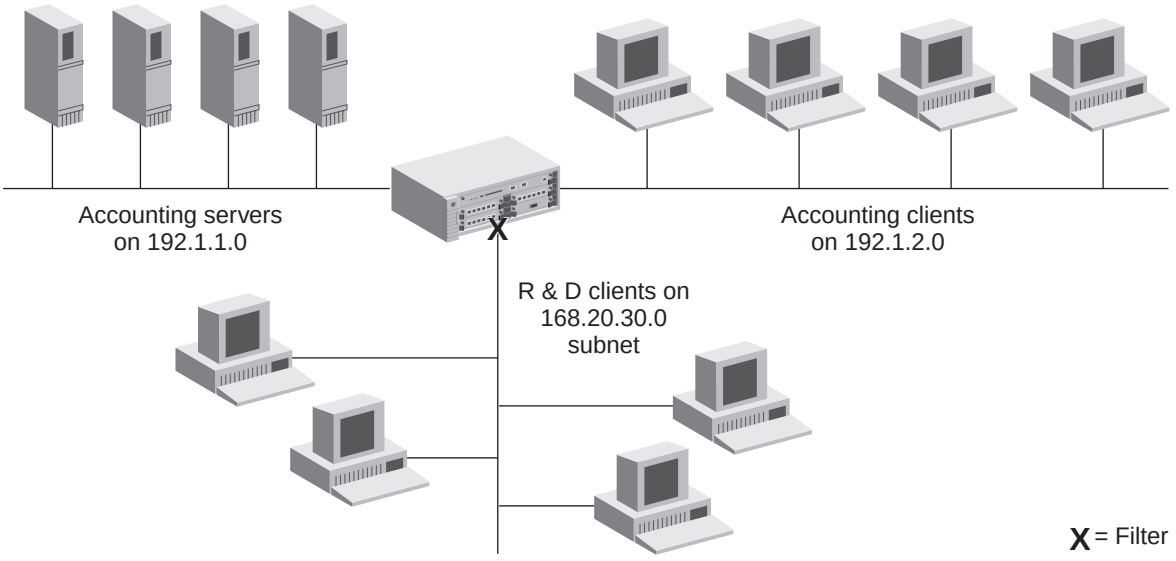
The control definition for the to/from classifier:

Control Field	Definition
Control Number	5
Control Name	*DBServer1*
Rate Limit Type	*none*
Service Level	high
Loss Eligible Status	*no*
802.1p tag for forwarded frames	none
Classifiers controlled	15

Example 2: Filtering Traffic to a Destination

In the following example, a classifier is defined to block access to the Accounting network 192.1.0.0 (which includes subnets 192.1.1.0 and 192.1.2.0) from the Research and Development 168.20.30.0 subnet. The associated control for this classifier sets a service level of *drop* to drop all traffic sent by the 168.20.30.0 subnet to the Accounting network.

Figure 87 Flow Classifier for Traffic to/from a Subnet



Classifier definition for filtering traffic to a specific destination:

Classifier Field	Classifier Definition
Classifier Number	26
Classifier Name	*IPFilter1*
Cast Type	*all*
IP protocol type	all
Source IP address	*168.20.30.0*
Source IP address mask	255.255.255.0
Destination IP address	192.1.0.0
Destination IP address mask	255.255.0.0
UDP source port range (start)	0
UDP source port range (end)	65535
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another address/port pattern?	n

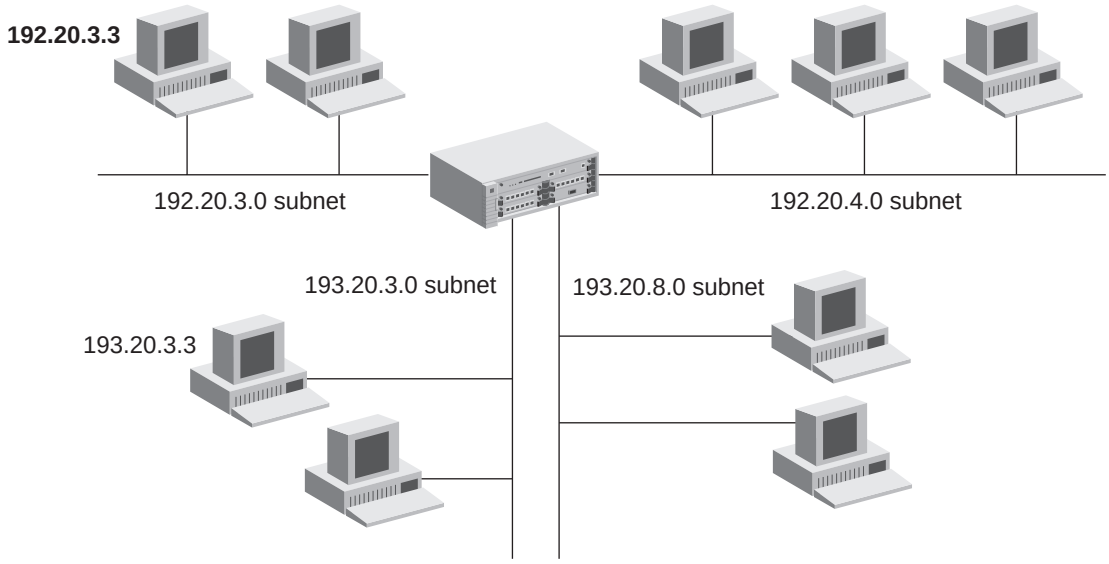
The control definition for this filtering classifier:

Control Field	Definition
Control Number	6
Control Name	*IPFilter1*
Rate Limit Type	*none*
Service Level	drop
Classifiers controlled	26

Example 3: Using Two Classifiers to Filter Traffic

In the following example, two flow classifiers (1 and 3) are defined with controls to filter IP traffic. Classifier 1 permits IP traffic between two hosts (192.20.3.3. and 193.20.3.3), while classifier 3 drops IP traffic TCP and UDP, (not ICMP) to and from one of the hosts (192.20.3.3).

Figure 88 Flow Classifier for Traffic to/from a Subnet



First classifier definition for filtering traffic to/from a specific destination:

Classifier Field	Classifier Definition
Classifier Number	1
Classifier Name	*192.20.3.3_to_193.20.3.3*
Cast Type	*all*
IP protocol type	all
Source IP address	*192.20.3.3*
Source IP address mask	255.255.255.255
Destination IP address	193.20.3.3
Destination IP address mask	255.255.255.255
UDP source port range (start)	0
UDP source port range (end)	65535
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another filter (address/port pattern)?	y
Source IP address	193.20.3.3
Source IP address mask	255.255.255.255
Destination IP address	192.20.3.3
Destination IP address mask	255.255.255.255
UDP source port range (start)	0
UDP source port range (end)	65535
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another filter (address/port pattern)?	n

The control definition for the first filtering classifier:

Control Field	Definition
Control Number	5
Control Name	*192.20.3.3_to_193.20.3.3*
Rate Limit Type	*none*
Service Level	best
802.1p tag for forwarded frames	none
Classifiers controlled	1

Second classifier definition for filtering traffic to/from a specific destination:

Classifier Field	Classifier Definition
Classifier Number	3
Classifier Name	*192.20.3.3_to_all*
Cast Type	*all*
IP protocol type	all
Source IP address	*192.20.3.3*
Source IP address mask	255.255.255.255
Destination IP address	0.0.0.0 (all)
Destination IP address mask	0.0.0.0
UDP source port range (start)	0
UDP source port range (end)	65535
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another filter (address/port pattern)?	y
Source IP address	0.0.0.0 (all)
Source IP address mask	0.0.0.0
Destination IP address	192.20.3.3
Destination IP address mask	255.255.255.255
UDP source port range (start)	0
UDP source port range (end)	65535
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another filter (address/port pattern)?	n

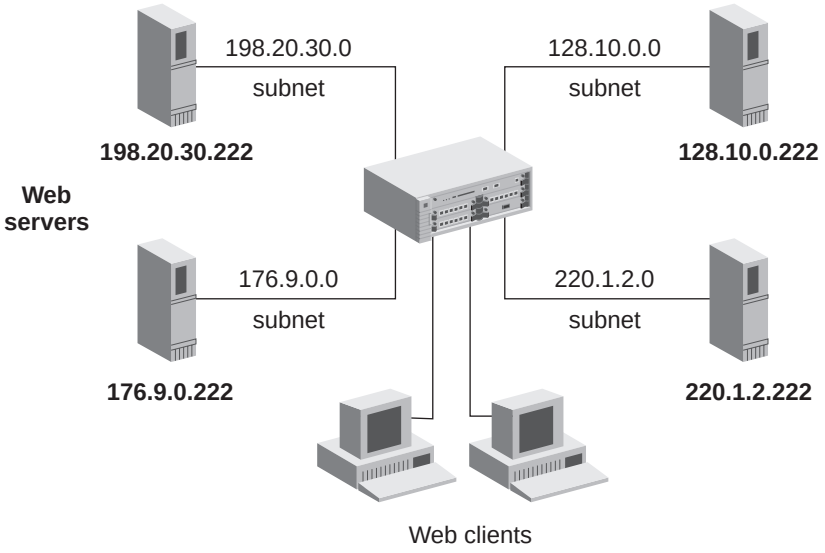
The control definition for the second filtering classifier:

Control Field	Definition
Control Number	7
Control Name	*192_20.3.3_to_all*
Rate Limit Type	*none*
Service Level	drop
Classifiers controlled	3

Example 4: Assigning High Priority to Specific Traffic

In the following example, a classifier is defined to give high priority to Web server (http) traffic. In this configuration, all Web servers have addresses that end in `.222`. This example could apply to any type of traffic that needs high priority (for example, mail server traffic).

Figure 89 Flow Classifier for Assigning High Priority to Web Traffic



Classifier definition for high-priority Web traffic:

Classifier Field	Classifier Definition
Classifier Number	17
Classifier Name	*httpServer1*
Cast Type	*unicast*
IP protocol type	TCP
Source IP address	*0.0.0.0*
Source IP address mask	0.0.0.0
Destination IP address	0.0.0.222
Destination IP address mask	0.0.0.255
UDP source port range (start)	0
UDP source port range (end)	65535
UDP destination port range (start)	80
UDP destination port range (end)	80

Classifier Field	Classifier Definition
Add another filter (address/port pattern)?	y
Source IP address	*0.0.0.0*
Source IP address mask	0.0.0.0
Destination IP address	0.0.0.222
Destination IP address mask	0.0.0.255
UDP source port range (start)	80
UDP source port range (end)	80
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another filter (address/port pattern)?	y
Source IP address	*0.0.0.222*
Source IP address mask	0.0.0.255
Destination IP address	0.0.0.0
Destination IP address mask	0.0.0.0
UDP source port range (start)	0
UDP source port range (end)	65535
UDP destination port range (start)	80
UDP destination port range (end)	80
Add another filter (address/port pattern)?	y
Source IP address	*0.0.0.222*
Source IP address mask	0.0.0.255
Destination IP address	0.0.0.0
Destination IP address mask	0.0.0.0
UDP source port range (start)	80
UDP source port range (end)	80
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another filter (address/port pattern)?	n

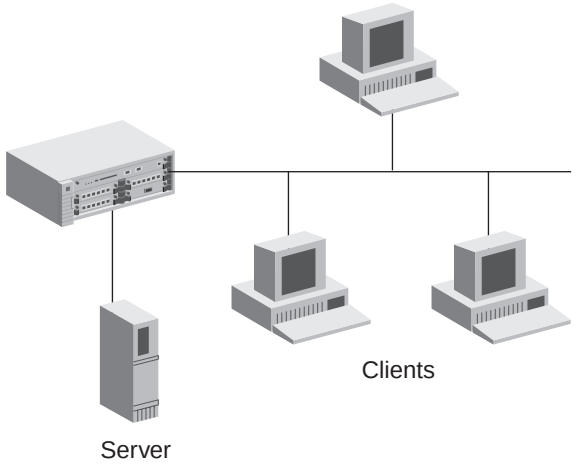
The control definition for this classifier is as follows:

Control Field	Definition
Control Number	7
Control Name	*httpServer1*
Rate Limit Type	*none*
Service Level	high
802.1p tag for forwarded frames	none
Classifiers controlled	17

Example 5: Nonflow Multimedia Tagged Traffic

In this example, a nonflow classifier is defined to classify bridged multimedia traffic with an IEEE 802.1p priority tag of 5 and control this traffic with a high priority transmit service level and a rate limit of 2048 Kbytes/sec.

Figure 90 Nonflow Classifier/Control for Bridged Multimedia Traffic



Nonflow classifier definition for Multimedia Traffic with priority tagging:

Classifier Field	**Classifier Definition**
Classifier Number	405
Classifier Name	*Interactive Multimedia*
Cast Type	*all (unicast, multicast broadcast, UMB)*
Protocol type	any
IEEE 802.1Q tag(s)	5

The control definition for this classifier is as follows:

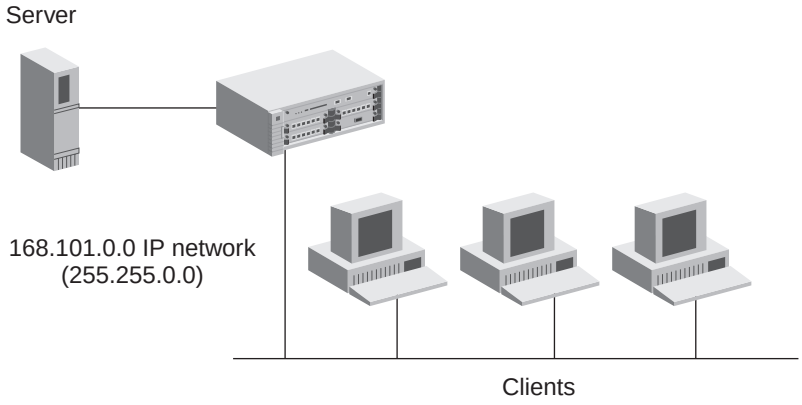
Control Field	**Definition**
Control Number	4
Control Name	*Interactive_Multimedia*
Rate Limit Type	*receivePort*
Service Level	high
Loss Eligible Status	*no*
Excess Service Level	drop
Excess Loss Eligible Status	–
Representation of Rate Limit	Kbytes/sec
Rate Limit Value	2048 KB
Burst Size	181 KB
Bridge Ports	1 to 13
802.1p tag for forwarded frames	– (uses tag from classifier, 5)
Classifiers controlled	405

Example 6: Bridged Nonflow IP Unicast Traffic

In this example, a nonflow classifier is defined to classify IP unicast traffic between clients and the server on the 168.101.0.0 network.

The applied control handles this *bridged* traffic with a high priority transmit service level and a rate limit of 75 percent of the link bandwidth.

Figure 91 Nonflow Classifier/Control for Bridged IP Unicast Traffic



Nonflow classifier definition for bridged IP Unicast Traffic:

Classifier Field	**Classifier Definition**
Classifier Number	430
Classifier Name	*IP_Unicast*
Cast Type	*unicast (U)*
Protocol type	IP
IEEE 802.1Q tag(s)	0 to 7

The control definition for this classifier is as follows:

Control Field	**Definition**
Control Number	5
Control Name	*IP_Unicast*
Rate Limit Type	*receivePort*
Service Level	high
Loss Eligible Status	*no*
Excess Service Level	low
Excess Loss Eligible Status	yes
Representation of Rate Limit	percent
Rate Limit Value	75 percent
Burst Size	363 KB
Bridge Ports	1 to 13
802.1p tag for forwarded frames	– (uses tags from classifier, 0 to 7)
Classifiers controlled	430

Modifying and Removing Classifiers and Controls

You can modify or remove a previously defined classifier or control. When modifying or removing a classifier, you specify the classifier number; when modifying removing a control, you specify the control number.

You may want to modify a classifier to alter source/destination information (flow classifier) or change IEEE 802.1p values (nonflow classifier). You may want to modify a control to specify a different service level (queue) or rate limit.

Important Considerations

Before modifying or removing classifiers or controls, review these guidelines:

- You cannot remove the default classifier or the default control, but you can modify the default control. You can modify other predefined classifiers and the predefined controls (for example, if you want to redefine the handling of Business Critical traffic, which is associated with predefined control 3).
- Once you apply a control to a classifier, you must remove the control for a classifier before you can modify or remove the classifier.
- When you remove a control, the associated classifiers are no longer controlled and no longer have a rate limit, service level, or 802.1p tag.
- If you want to modify a classifier that has several address/port definitions, you must supply them again during the modification process. If you do not reenter them, the system deletes these definitions.
- If you want to modify a control that uses a rate-limit type of aggregate or receivePort with several rate-limit values, you can change one rate-limit value without affecting the other defined rate-limit values.

QoS Excess Tagging

Your system enables you to tag nonconforming excess (packets that exceed the rate-limit criteria) with a special IEEE 802.1p tag value. This refers to any packets marked as excess that you want to tag. By default, excess tagging is disabled.

You can use your configuration tool (Administration Console or Web Console) to enable or disable excess tagging and display your excess tagging information.

If you *enable excess tagging*, you can specify an IEEE 802.1p tag value for nonconforming excess in the range of from *0 to 7*, with *0* as the default. (See "IEEE 802.1p" earlier in this chapter for a list of the tags and their associated priority levels). Specifying *1* means that nonconforming excess become background traffic.

Example of QoS Excess Tagging

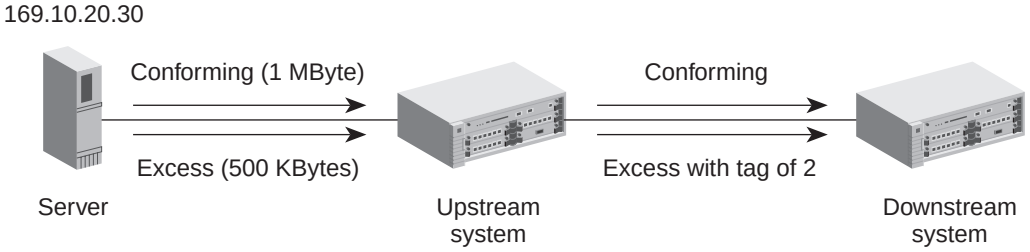
The following example shows how to use a classifier, control, and QoS excess tagging to tag conforming QoS multicast video traffic from a server as *Streaming Multimedia* 802.1p service and tag any excess traffic as *Standard* 802.1p service.

In this sample configuration:

- The configured rate limit is 1 MByte, so when the server sends 1.5 MBytes, the upstream system knows 1 MByte is conforming and 500 Kbytes is excess.
- The upstream system configures the classifier, control, and the tagging, and has the QoS flow. The upstream system passes the excess traffic with the tag of 2 (Standard priority) to the downstream system.
- The downstream system can prioritize traffic from this flow at layer 2, using its default 802.1p classifier 404 (for conforming packets) and classifier 402 (for nonconforming excess) along with the corresponding controls 4 and 2.

For this configuration, you must enable QoS excess tagging with a tag value of 2 in addition to defining the classifier and control.

Figure 92 QoS Excess Tagging



Classifier definition for QoS Excess Tagging:

Classifier Field	Classifier Definition
Classifier Number	25
Classifier Name	VideoServer1
Cast Type	*multicast*
IP protocol type	UDP
Source IP address	*169.10.20.30*
Source IP address mask	255.255.255.255
Destination IP address	0.0.0.0
Destination IP address mask	0.0.0.0
UDP source port range (start)	0
UDP source port range (end)	65535
UDP destination port range (start)	2010
UDP destination port range (end)	2020
Add another filter (address/port pattern)?	y
Source IP address	*169.10.20.30*
Source IP address mask	255.255.255.255
Destination IP address	0.0.0.0
Destination IP address mask	0.0.0.0
UDP source port range (start)	2010
UDP source port range (end)	2020
UDP destination port range (start)	0
UDP destination port range (end)	65535
Add another filter (address/port pattern)?	n

The accompanying control definition:

Control Field	Definition
Control Number	5
Control Name	*VideoServer1*
Rate Limit Type	*receivePort*
Service Level	high
Loss Eligible Status	*no*
Excess Service Level	low
Excess Loss Eligible Status	yes
Representation of Rate Limit	Kbytes/sec
Rate Limit Value	1024
Burst Size	128
Bridge Ports	all (1 to 19)
802.1p tag for forwarded frames	4
Classifiers controlled	25

Transmit Queues and QoS Bandwidth

QoS uses four transmit queues:

- **Control queue** — The transmit queue for reserved network control traffic, such as RIP or OSPF updates, as well as RSVP data flows. This queue is always serviced first. Bandwidth for this queue is set via RSVP.
- **High priority queue** — The transmit queue with the second highest priority. You can map classifiers directly to this queue.
- **Best effort queue** — The transmit queue used by default for all traffic except reserved traffic.
- **Low priority queue** — The transmit queue with the lowest priority. All traffic assigned to this queue is forwarded only if there is bandwidth still available after the other queues are serviced. Low priority packets do not have bandwidth allocated.

You can configure the weighting of the high priority and best effort transmit queues by using the option to modify QoS bandwidth. By default, the weighting of the queues is 75 percent high priority traffic and 25 percent best effort traffic. Keep in mind that the weighting does not represent guaranteed output bandwidth for these queues, since they are serviced in relative percentages after the control queue is serviced.

When you modify the QoS bandwidth, you specify the percentage of bandwidth used for the high priority transmit queue on the output link. You can specify a value in the range *0* to *100*. The value you specify determines the ratio of high priority to best effort traffic, as follows:

- The value *75* (the default) specifies that three high-priority packets are transmitted for each best effort packet (ratio of 75/25).
- The value *50* sets equal priority for high priority and best effort packets (ratio of 50/50).
- The value *100* is strict prioritization; it allows best effort packets to be sent only when no high priority packets need to be sent.

No bandwidth is ever lost. Because QoS uses ratios, any unused bandwidth can be used by a lower priority queue.

LDAP

Lightweight Directory Access Protocol (LDAP) is an Internet standard for directory services. LDAP directory services is based on the client/server model and runs over TCP/IP. The CoreBuilder 3500 contains the LDAP client software necessary to communicate and exchange configuration information for QoS parameters stored on the LDAP server.

Important Considerations

Review the following before you enable LDAP on your system.

- You must configure an LDAP server on either a workstation/PC (using LDAP from Netscape Navigator), or on a Solaris or HP Unix workstation (using University of Michigan LDAP). Each group of CoreBuilder 3500s have QoS parameters stored in a particular directory on the server.
- The LDAP server must have an initial set of QoS parameters before LDAP can be used. This is typically accomplished by installing an "ldif" file on the server for each QoS group. The ldif file is specific to the CoreBuilder 3500 and is necessary for the client software to run.

For the ldif file with default settings, see the Software and Documentation CD-ROM *that is shipped with your system.*

- Parameter changes for a specific group may affect more than one system. If you know that a change will affect more than one system, you should disable LDAP to test the change. After you are sure you want the change you can reenable LDAP.

Operation

When an LDAP client connects to the LDAP server and polls it for information, the server responds with an answer and downloads any changes if necessary. LDAP directory services can save you a tremendous amount of time by making it easy to update QoS parameters from a single source instead of having to update each individual client. In addition, QoS parameters stored on the server can be tailored to meet the needs of different users by assigning group configuration names. The ability to configure groups lets you associate a set of particular QoS parameters with multiple CoreBuilder 3500 systems.

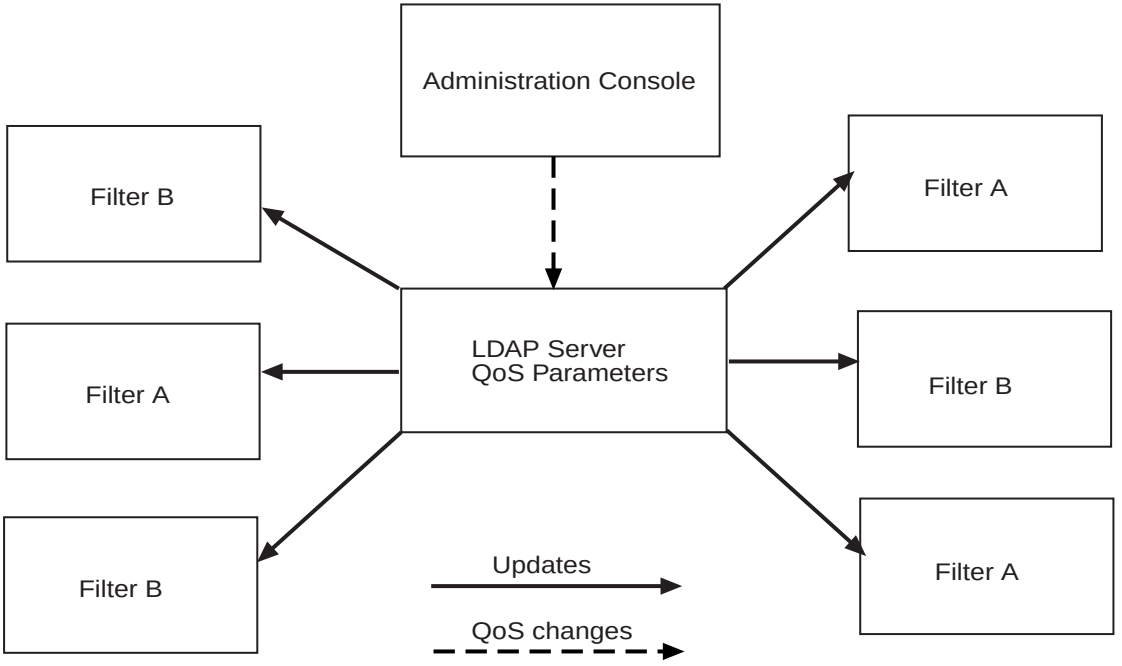
Group configuration names are created on the LDAP server.

When LDAP is enabled, it identifies the IP address of the LDAP server, a poll time, and a groupConfig name. If a connection to the LDAP server cannot be established when the CoreBuilder 3500 is powered on, the settings stored in nvFlash are used to provide the fundamental QoS parameters, including the default control (best effort) for the default nonflow classifier (499). When a successful connection is made, the QoS parameters are retrieved from the LDAP server using a search filter (a group name or a wildcard). Once a CoreBuilder 3500 is associated with a group configuration, it is automatically updated with the parameters associated with that group configuration.

Each time you make a change to the QoS parameters, the change is sent to the LDAP server immediately. In addition, the CoreBuilder 3500 clients poll the LDAP server every poll period for new configuration information and updates the new QoS configuration in nvFlash at that time.

You must have Administrator privileges (correct user name and password) to access the LDAP server from the Administration Console. Otherwise, the LDAP server denies the update request.

Figure 93 Updating QoS Parameters from the LDAP Server



RSVP

The Resource Reservation Protocol (RSVP) is an IP service that prevents real-time traffic such as voice or video from overwhelming bandwidth resources. In general, RSVP supports QoS IP flow specifications by placing and managing resource reservations across the network (setting admission control, policing, and restricting the creation of RSVP reservations). Your system can reserve and police the bandwidth requested for each RSVP session.

RSVP is receiver-oriented, that is, an end system can send an RSVP request on behalf of an application to request a specific QoS from the network. At each hop along the path back to the source, routers such as your system register the reservation and try to provide the required QoS. If a router cannot provide the required QoS, its RSVP process sends an error to the end system that initiated the request.

RSVP is designed for multicast applications, but it also supports resource reservations for unicast applications as well as point-to-point transmissions. RSVP does not implement a routing algorithm.

To use RSVP, you must be routing. (RSVP operates at Layer 3 for IP-based data flows.) Endstations in the configuration must support RSVP in order to request the reservation of bandwidth through the network.

By default, RSVP is disabled on the system. If you decide to use RSVP, it is recommended that you use the default RSVP settings.

RSVP Terminology

Familiarize yourself with the following RSVP terms:

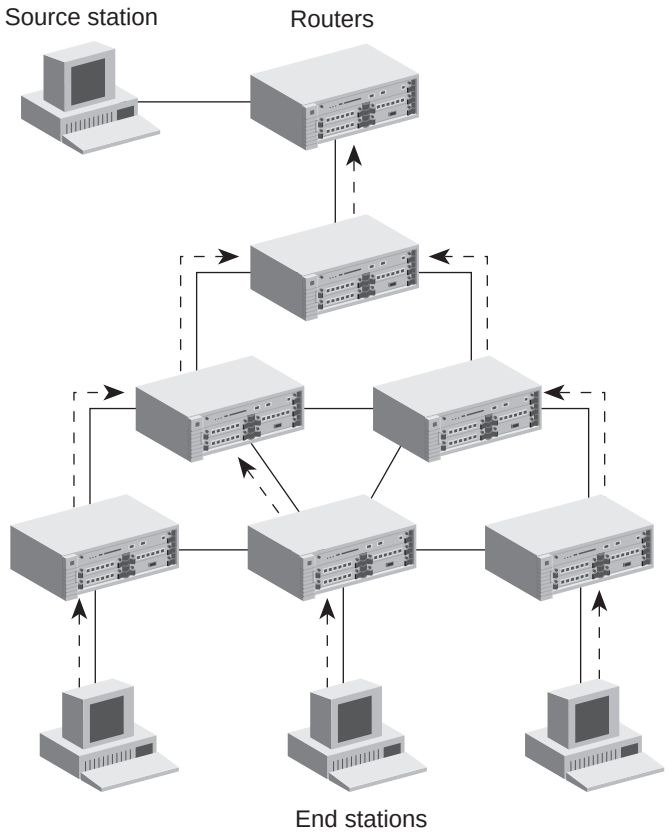
- **RSVP flow** — A data stream that operates in simplex, going one way from the origin to multiple destinations. The flows go from a set of senders to a set of receivers.
- **Reservation style** — The types of multicast flows that RSVP installs:
  - **Fixed filter (distinct) style** — A flow originating from one sender only (for example, a video application). This style requires a separate reservation per sender on each transmission type.
  - **Shared explicit** — A shared-reservation flow originating from a limited number of senders (for example, an audio application). This style identifies the flows for specific network resources. A single reservation can be applied to all senders in the set.
  - **Wildcard filter** — A shared-reservation flow from all senders.
- **Total reservable bandwidth percentage** — Controls the admission control policy. RSVP begins to refuse reservations when the requested bandwidth on an output link exceeds the total reservable bandwidth. You specify a percentage of the output link (a value of from *0* to *200,* with *50* as the default). This is the amount of bandwidth that you allow RSVP to reserve in the system. You can over-subscribe (over 100) and specify a value up to 200.
- **Maximum per-reservation bandwidth** — The largest reservation that RSVP attempts to install. Specify this bandwidth using a percentage of the output link (a value of from *0 to 100; 50* is the default).

- **Policing options** — Ensure that an RSVP session uses only as much bandwidth as it requested. The policing options mandate when to drop nonconforming excess packets. You configure the system to observe one of three policing options:
  - **Edge** — Causes nonconforming excess packets to be dropped only at the edge (that is, when the traffic has not yet passed through any network device that has already performed policing for that flow). The system polices the flow when RSVP requests it. This is the default policing option. The RSVP protocol knows how to detect what is edge and what is not when policing
  - **Always** — The system always polices the flow.
  - **Never** — The system never polices the flow, even if RSVP requests it.

Example of RSVP

Figure 94 shows an RSVP configuration in which an RSVP reservation request flows upstream along a multicast delivery tree (with routers, Layer 3 switches such as the CoreBuilder 3500) until it merges with another reservation request for the same source.

Figure 94 Sample RSVP Configuration



Setting RSVP Parameters

If you enable RSVP, you specify the following information:

- The maximum total reservable bandwidth
- The maximum per-reservation bandwidth
- The policing option (*edge, always,* or *never,* with *edge* as the default)
- The service level for excess/policed traffic (*best* or *low*, with *low* as the default). This setting applies to the excess traffic with the reserved bandwidth (that is, in which queue it should be placed).
- Whether nonconforming excess are loss eligible (*yes* or *no,* with *no* as the default)

After you enable RSVP, you can use your management interface (for example, the Administration Console) to display summary or detail information about RSVP.

18 DEVICE MONITORING

This chapter provides descriptions and key operational information about device monitoring features and tools of your CoreBuilder® 3500 system. The chapter covers these topics.

- Device Monitoring Overview
- Key Concepts and Tools
- Event Logging
- Baselining
- Roving Analysis
- Ping
- traceRoute
- SNMP
- Remote Monitoring (RMON)
- Management Information Base (MIB)

You can manage baselining, roving analysis, and SNMP in either of these ways:

- *From the menus of the Administration Console. See the* Command Reference Guide.
- *From the respective folders of the Web Management software. See the* Web Management User Guide.

You can manage event logging only from the menus of the Administration Console.

RMON MIBs are accessible only through SNMP-based applications.

Device Monitoring Overview

You can use the device monitoring features and tools described in this chapter to analyze your network periodically and to identify potential network problems before they become serious. To identify potential problems in your network, use:

- Event logging.
- Baselining.
- Roving analysis.
- RMON information.

To test and validate paths in your network, use:

- Ping.
- traceRoute.
- polling MIBs via SNMP.

The SNMP protocol and the Management Information Base (MIB) are described in this chapter to give you some background on how performance data is collected on the network.

Key Concepts and Tools

Key concepts and tools for the device monitoring of your system are described in this section to give you a perspective of the scope of device monitoring.

Administration Console

The Administration Console provides you with access to all the features of your system. It also provides you access to some of the device monitoring tools, such as event logging, baselining, roving analysis, ping, traceRoute, and snapshot. You access the Administration Console locally through the serial terminal or modem port on the Switch or remotely via a Telnet connection. See Chapter 2 for more information.

Web Management Tools

The Web Management tools provide you access to the Administration Console remotely via the Internet. It provides you with complete access to the Administration Console as if you are connected locally or through a Telnet connection. See the *Web Management User Guide* for more information.

Network Management Platform

The network management platform allows you to view the health of your overall network. With the platform, you can understand the logical configuration of your network and configure views of your network to understand how devices work together and the role they play in the users' work. The network management platform that supports your Transcend® Network Control Services software installation can provide valuable troubleshooting tools.

SmartAgent Embedded Software

Traditional SNMP management places the burden of collecting network management information on the management station. In this traditional model, software agents collect information about throughput, record errors or packet overflows, and measure performance based on established thresholds. Through a polling process, agents pass this information to a centralized network management station whenever they receive an SNMP query. Management applications then make the data useful and alert the user if there are problems on the device.

For more information about traditional SNMP management, see "SNMP" later in this chapter.

SmartAgent® software, which uses Remote Monitoring (RMON), is self-monitoring, collecting and analyzing its own statistical, analytical, and diagnostic data. In this way, you can conduct network management by exception — that is, you are only notified if a problem occurs. Management by exception is unlike traditional SNMP management, in which the management software collects *all* data from the device through polling.

Other Commonly Used Tools

These commonly used tools can help you troubleshoot your network:

- Built-in system features such as event logging, baselining, remote monitoring (RMON), and creating snapshots.
- Network software tools, such as ping and traceRoute. You can use these applications to troubleshoot and test your system.
- Monitor devices using analyzers connected to your system's roving analysis (RAP) ports.
- Network utility software, such as Telnet, FTP, and TFTP, can be used to troubleshoot, configure and upgrade your system.
- Tools such as Cable Testers are used for working on physical network problems.

Event Logging

The event log messages display real-time information about the state of the system, a specific service, or both, and can help you diagnose site-specific problems. The event log captures several types of log messages from various services (applications) and sends them to the Administration Console.

Important Consideration

Event logging generates a great deal of detailed data. It can therefore have significant performance implications. Use it only for troubleshooting purposes.

Displaying the Event Log Configuration

You can display the current settings for the event log. The display identifies the output device, the severity levels for the log messages, and the supported services.

Configuring the Output Devices

For the Administration Console, you can configure one or more of these severity levels:

- **error** — Logs application-specific error messages to the Console
- **warning** — Indicates a nonfatal problem
- **config** — Indicates configuration changes
- **info** — Indicates changes in the state of the system that are not caused by events at any other severity level

Configuring the Services

At this release, you can enable the logging of messages that pertain to the following services:

- System level
- AppleTalk
- IPX

For a specific service or all services, you can configure one or more of the four severity levels.

Baselining

Normally, statistics for MACs and ports start to compile when you turn the system on. Baselining allows you to view statistics compiled over the period of time since a baseline was set. By viewing statistics relative to a baseline, you can more easily evaluate recent activity in your system or on your network.

Important Considerations

- Baselining is maintained across Administration Console sessions. Statistics that you view after setting the baseline indicate that they are relative to the baseline. To view statistics as they relate only to the most recent power up, disable the baseline.
- Baselining affects the statistics that are displayed for Ethernet ports and bridges.

Displaying the Current Baseline

You can get a display the current baseline to see when the baseline was last set and to determine if you need a newer baseline for viewing statistics.

Setting a Baseline

You can reset the baseline counters to zero *(0)*. The system maintains the accumulated totals since power up. The baseline is time-stamped.

Enabling or Disabling Baselines

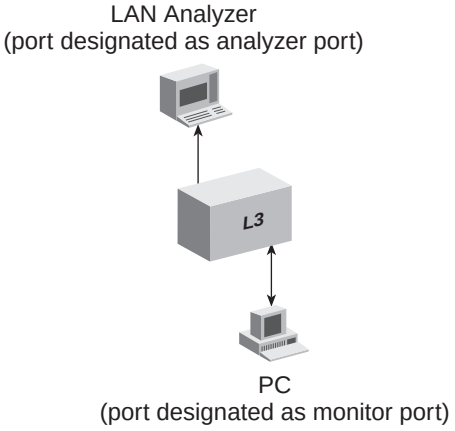
When you re-enable a baseline, the counters return to the values that accumulated since the most recent baseline that you set. Disabling a baseline returns the counters to the total accumulated values since the last power up.

Roving Analysis

Roving analysis is the mirroring of Fast Ethernet, Gigabit Ethernet, or Fiber Distributed Data Interface (FDDI) port traffic to another port of the same media type. This second port has an external RMON-1/RMON-2 probe or analyzer attached such as the 3Com Transcend Enterprise Monitor. Through the probe, you can monitor traffic on any switched segment. Figure 95 shows a sample configuration.

- The port with the analyzer attached is called the *analyzer port*.
- The port that is being monitored is called the *monitor port*.

Figure 95 Connecting an Analyzer to the System



The purpose of roving analysis is to:

- Analyze traffic loads on each segment so that you can continually optimize your network loads by moving network segments.
- Troubleshoot switched network problems (for example, to find out why a particular segment has so much traffic).

When you set up a roving analysis configuration, the system copies both transmit and receive port data and forwards it to the port on which the network analyzer is attached — without disrupting the regular processing of the packets.

Key Guidelines for Implementation

To enable the monitoring of ports on a system, follow these general steps:

1. Add the port on which you want to attach the network analyzer.
2. Start roving analysis.
   a. Select the port that you want to monitor.
   b. Enter the analyzer port's MAC address.

The system provides commands to add and remove (define and undefine) the analyzer port, to display the current analyzer and monitor ports, and to start and stop analysis.

See the "Roving Analysis" chapter in the *Command Reference Guide* for details.

Important Considerations

- You can connect a maximum of 16 network analyzers to a system.
- The network analyzer cannot be located on the same bridge port as the port that you want to monitor.
- For more accurate analysis, attach the analyzer to a dedicated port instead of through a repeater.
- When the analyzer port is set, it cannot receive or transmit any other data. Instead, it receives only the data from the port(s) to be monitored.
- If Spanning Tree Protocol was enabled on the analyzer port, it is automatically disabled. When the analyzer port is undefined, the port returns to its configured Spanning Tree state and functions as it did before it was set as an analyzer port.
- When you configure a port that is part of a virtual LAN (VLAN) as an analyzer port, the port is removed from all VLANs of which it is a member. When you remove the analyzer port, it becomes a member of the default VLAN. You have to manually add it back to its original VLANs.
- You cannot use roving analysis to monitor trunk ports or resilient link ports.

- If the physical port configuration changes in the system (that is, if you remove or rearrange modules), the MAC address of the analyzer port remains fixed. If the module with the analyzer port is replaced with a different media type module, the RAP configuration is cleared.
- Defining a monitor port (`analyzer start` command) affects that port's ability to collect RMON data. Table 62 shows which RMON groups can continue to collect data, and which cannot after the port has become a monitor port.

Table 62 Roving Analysis and RMON Data

RMON Groups	Works with Roving Analysis?
RMON-1 Groups	
Statistics	Yes
History	Yes
Alarm	Yes
Hosts	No
HostTopN	No
Matrix	No
Event	Yes
RMON-2 Groups	
protocolDir	Yes
protocolDist	No
addressMap	No
nlHost	No
nlMatrix	No
alHost	No
alMatrix	No
probeConfig. probeCapabilities	Yes

The RMON groups that require samples of traffic from the ASICs will not work because they do not receive any traffic data when a port is defined as a monitor port. The system is capable of doing either roving analysis or traffic sampling, but not both at the same time.

- The monitor and analyzer ports must be of the same type media. For example, FDDI ports cannot be monitored on Ethernet ports and Ethernet ports cannot be monitored on FDDI ports.
- You can use a Fast Ethernet port to monitor a Gigabit Ethernet port, but a warning message will be printed. Because the analyzer port is slower than the monitor port, the analyzer port may not see all frames.

Ping

The ping feature is a useful tool for network testing, performance measurement, and management. It uses the Internet Control Message Protocol (ICMP) echo facility to send ICMP echo request packets to the IP destination that you specify. See Chapter 11 for more information about ICMP.

When a router sends an echo request packet to an IP station using ping, the router waits for an ICMP echo reply packet. The response indicates whether the remote IP is available, unreachable, or not responding.

Important Consideration

When you specify a hostname with ping, the hostname and its associated IP address must be configured on a network name server. Also, you must add the IP address on the name server to the list of name server addresses that are associated with the network domain name. See Domain Name Servers in Chapter 11.

Using Ping

The system provides two ping functions:

- **ping** — Uses the hostname or IP address to ping a host with default options
- **advancedPing** — Uses the hostname or IP address to ping a host with the advanced ping options that you specify

Ping Responses

This list gives the possible responses to a ping:

- If the host is reachable, the system displays information about the ICMP reply packets and the response time to the ping. The amount of information depends on whether the quiet option is enabled or disabled.
- If the host does not respond, the system displays the ICMP packet information and this message: `Host is Not Responding.` (You may see this message if you have not configured your gateway IP address.)
- If the packets cannot reach the host, the system displays the ICMP packet information and this message: `Host is Unreachable.` A host is unreachable when there is no route to that host.

Strategies for Using ping

Follow these strategies for using ping:

- Ping devices when your network is operating normally so that you have a performance baseline for comparison.
- Ping by *IP address* when:
  - You want to test devices on different subnetworks. This method allows you to ping your network segments in an organized way, rather than having to remember all the hostnames and locations.
  - Your DNS server is down and your system cannot look up host names properly. You can ping with IP addresses even if you cannot access hostname information.
- Ping by *hostname* when you want to identify DNS server problems.
- To troubleshoot problems involving large packet sizes, ping the remote host repeatedly, increasing the packet size each time.

traceRoute

Use the traceRoute feature to track the route of an IP packet through the network. TraceRoute information includes all of the nodes in the network through which a packet passes to get from its origin to its destination. The traceRoute feature uses the IP time-to-live (TTL) field in User Datagram Protocol (UDP) probe packets to elicit an ICMP Time Exceeded message from each gateway to a particular host.

Using traceRoute

The system provides two traceRoute functions:

- **traceRoute** — Uses the hostname or IP address to trace a route to a host with default options
- **advancedTraceRoute** — Uses the hostname or IP address to trace a route to a host with the advanced traceRoute options that you specify

traceRoute Operation

To track the route of an IP packet, the traceRoute feature launches UDP probe packets with a small TTL value and then listens for an ICMP Time Exceeded reply from a gateway. Probes start with a small TTL of 1 and increase the value by 1 until one of the following events occurs:

- The system receives a Port Unreachable message, indicating that the packet reached the host.
- The probe exceeds the maximum number of hops. The default is 30.

At each TTL setting, the system launches three UDP probe packets, and the traceRoute display shows a line with the TTL value, the address of the gateway, and the round-trip time of each probe. If a probe answers from different gateways, the traceRoute feature prints the address of each responding system. If no response occurs in the 3-second time-out interval, traceRoute displays an asterisk (*) for that probe. Other characters that can be displayed include the following:

- **`!N`** — Network is unreachable
- **`!H`** — Host is unreachable
- **`!P`** — Protocol is unreachable
- **`!F`** — Fragmentation is needed
- **`!<`*n*`>`** — Unknown packet type

SNMP

Simple Network Management Protocol (SNMP), one of the most widely used management protocols, allows management communication between network devices and your management workstation across TCP/IP internets.

See Chapter 2 to review where SNMP fits in the Open System Interconnection (OSI) reference model for the network environment.

Most management applications, including Status Watch applications, require SNMP to perform their management functions.

SNMP Overview

The following sections provide an overview of SNMP.

Manager/Agent Operation

SNMP communication requires a *manager* (the station that is managing network devices) and an *agent* (the software in the devices that talks to the management station). SNMP provides the language and the rules that the manager and agent use to communicate.

Managers can discover agents:

- Through autodiscovery tools on Network Management Platforms (such as HP OpenView Network Node Manager)
- When you manually enter IP addresses of the devices that you want to manage

For agents to discover their managers, you must provide the agent with the IP address of the management station or stations.

Managers send requests to agents (either to send information or to set a parameter), and agents provide the requested data or set the parameter. Agents can also send information to the managers (without being requested by the managers) through trap messages, which inform the manager that certain events have occurred.

You can use either an in-band or an out-of-band IP interface to manage the system with SNMP. See Chapter 2 for more information.

IP Address Assignment

For the manager and agent to be able to communicate with one another you need to assign IP addresses as follows:

- Assign an IP address to either the system processor out-of-band Ethernet port or an in-band Ethernet port, depending on where the management station is attached.
  - **Out-of-Band** — Use the `management ip interface define` command to assign the IP address for the out-of band Ethernet port
  - **In-band** — Use the `ip interface define` command to assign IP address to the in-band Ethernet port.

Set the destination IP address to which the traps should be forwarded to the manager by the system agent. See "Trap Reporting" later in this chapter.

SNMP Messages

SNMP supports queries (called *messages*) that allow the protocol to transmit information between the managers and the agents. Types of SNMP messages:

- **Get** and **Get-next** — The management station requests an agent to report information.
- **Set** — The management station requests an agent to change one of its parameters.
- **Get Responses** — The agent responds to a Get, Get-next, or Set operation.
- **Trap** — The agent sends an unsolicited message informing the management station that an event has occurred.

Management Information Bases (MIBs) define what can be monitored and controlled within a device (that is, what the manager can Get and Set). An agent can implement one or more groups from one or more MIBs. See "Management Information Base (MIB)" later in this chapter for more information.

Trap Reporting

Traps are events that devices generate to indicate status changes. Every agent supports some trap reporting. You must configure trap reporting at the devices so that these events are reported to your management station to be used by the Network Management Platforms (such as HP OpenView Network Node Manager or SunNet Manager).

You do not need to enable all traps to effectively manage a switch. To decrease the burden on the management station and on your network, you can limit the traps reported to the management station.

MIBs are not required to document traps. The SNMP agent supports the limited number of traps defined in Table 63. More traps may be defined in vendors' private MIBs.

Table 63 Traps Supported by SNMP

Trap No.	Trap Name	Source	Indication
1	Cold Start	MIB II	The agent has started or been restarted.
2	Link Down	MIB II	The status of an attached communication interface has changed from *up* to *down*.
3	Link Up	MIB II	The status of an attached communication interface has changed from *down* to *up*.
4	Authentication Failure	MIB II	The agent received a request from an unauthorized manager.
5	New Root	Bridge MIB	The sending agent has become the new root of the Spanning Tree.
6	Topology Change	Bridge MIB	Any of bridge configured ports transitions from the Learning state to the Forwarding state, or from the Forwarding state to the Blocking state.
7	System Overtemperature	3C System MIB	The system temperature exceeds a certain threshold.
8	Power Supply Failure	3C System MIB	The trap that is generated when a power supply unit fails in a system with a dual power supply.
13	Address Threshold	3C System MIB	The number of addresses stored in the bridge reaches a certain threshold.
14	System Fan Failure	3C System MIB	One of the system fans fails.

Table 63 Traps Supported by SNMP (continued)

Trap No.	Trap Name	Source	Indication
15	SMT Hold Condition	3C FDDI MIB	FDDI SMT state either in holding-prm or holding-sec.
16	SMP Peer Wrap Condition	3C FDDI MIB	FDDI SMT connection does not connect to an M-port under DAS mode.
17	MAC Duplicate Address Condition	3C FDDI MIB	A status that there are more than one MAC address.
18	MAC Frame Error Condition	3C FDDI MIB	A status that the error frames rate reaches a certain threshold.
19	MAC Not Copied Condition	3C FDDI MIB	A status that the not copied frames rate reaches a certain threshold.
20	MAC Neighbor Change	3C FDDI MIB	A change in a MAC's upstream neighbor address or downstream neighbor address.
21	MAC Path Change	3C FDDI MIB	A status that the FDDI Path changes.
22	Port LER Condition	3C FDDI MIB	A status that the FDDI port link error rate reaches a certain threshold.
23	Port Undesired Connection	3C FDDI MIB	A port connection does not match the connection policy.
24	Port EB Error Condition	3C FDDI MIB	Elasticity Buffer has overflowed.
25	Port Path Change	3C FDDI MIB	Any port path change.
26	Rising Alarm	RMON MIB	An alarm entry crosses its rising threshold.
27	Falling Alarm	RMON MIB	An alarm entry crosses its falling threshold.
28	Response Received	POLL MIB	A disabled device begins responding.
29	Response Not Received	POLL MIB	An enabled device stops responding.
32	VRRP New Master	VRRP MIB	The sending agent transitioned from *Backup* state to *Master* state.

Table 63 Traps Supported by SNMP (continued)

Trap No.	Trap Name	Source	Indication
33	VRRP Authentication Failure	VRRP MIB	A VRRP packet is received from a router whose authentication failed. The authentication failure under this trap is sub-divided under three types: ■ Invalid authentication type ■ Authentication type is valid, but does not match the type configured ■ Authentication type is valid and matches, but has the wrong key
35	QOS Intruder	QOS MIB	This trap is generated when a user attempts to access a network restricted with a QoS One-Way TCP Filter. The trap contains the following information: ■ Source IP Address ■ Destination IP Address ■ Destination IP Port Number ■ QoS Classifier Number To prevent a denial-of-service (DOS) attack, the system will not generate more than one QOS Intruder trap per second. Thus, an attacker cannot flood a management station with traps or overload a switch's ability to pass or handle messages.

To minimize SNMP traffic on your network, you can implement trap-based polling. Trap-based polling allows the management station to start polling only when it receives certain traps. Your management applications must support trap-based polling for you to take advantage of this feature.

Security

SNMP uses community strings as a form of management security. To enable management communication, the manager must use the same community strings that are configured on the agent. You can define both read and read/write community strings.

Because community strings are included unencoded in the header of a User Datagram Protocol (UDP) packet, packet capture tools can easily access this information. As with any password, change the community strings frequently.

Setting Up SNMP on Your System

To manage your system from an external management application, you must configure SNMP community strings and set up trap reporting, as described in this section.

You must also assign an IP address to either the system processor out-of-band Ethernet port or an in-band Ethernet port, depending on where the management station is attached. See Chapter 2 for more information.

You can manage the system using an SNMP-based external management application. This application (called the SNMP manager) sends requests to the system, where they are processed by the internal SNMP agent.

You can gain access to the Remote Monitoring (RMON) capabilities of your system through SNMP applications such as Transcend® Network Control Services software. See "RMON in Your System" later in this chapter for information about the RMON capabilities of your system.

The SNMP agent provides access to the collection of information about your system. (You can view many system-specific settings.) Your views of MIB information differ depending on the system SNMP management method that you choose.

In addition, you can configure a system SNMP agent to send traps to an SNMP manager to report significant events.

Access to system information through SNMP is controlled by community strings.

Displaying Community Strings

You can display the current SNMP community strings that are assigned.

Configuring Community Strings

A community string is an octet string, included in each SNMP message, that controls access to system information. The system SNMP agents internally maintain two community strings that you can configure:

- *Read-only* community strings with the default *public*
- *Read-write* community strings with the default *private*

When an SNMP agent receives an SNMP request, the agent compares the community string in the request with the community strings that are configured for the agent:

- SNMP *get*, *get-next*, and *set* requests are valid if the community string in the request matches the agent's *read-write* community.
- SNMP *get* and *get-next* requests are valid if the community string in the request matches the agent's *read-only* community string or read-write community string.

Community string length

When you set a community string, you can specify any value up to 48 characters long.

Administering SNMP Trap Reporting

For network management applications, you can use the Administration Console to manually administer the trap reporting address information.

- **Displaying Trap Reporting Information** — When you display the trap reporting information, the system displays the various SNMP traps and their currently configured destinations.
- **Configuring Trap Reporting** — You can add new trap reporting destination configurations and modify existing configurations. You can define up to 10 destination addresses and the set of traps that are sent to each destination address.

The trap numbers that you enter tell the system to send the corresponding traps to the destination address when the events occur. No unlisted traps are transmitted.

- **Removing Trap Destinations** — When you remove a destination, no SNMP traps are reported to that destination.
- **Flushing All SNMP Trap Destinations** — When you flush the SNMP trap reporting destinations, you remove all trap destination address information for the SNMP agent.
- **Set SNMP smtProxyTraps** — Controls SNMP's ability to alert you, by means of an SNMP-to-SMT proxy, of a significant event occurring in the FDDI station statistics.

Control SNMP Write Requests

Allows or disallows SNMP write requests.

Remote Monitoring (RMON)

This section provides information about Remote Monitoring (RMON) and the RMON-1 and RMON-2 Management Information Base (MIB) groups implemented in your system. The following topics are included.

- Overview of RMON
- RMON Benefits
- 3Com Transcend RMON Agents
- RMON in Your System
- RMON-1 Groups
- RMON-2 Groups

The CoreBuilder 3500 does not provide RMON support for Gigabit Ethernet ports.

You can gain access to the RMON capabilities of the system through SNMP applications such as Transcend Network Control Services software, not through the serial interface or Telnet. For more information about the details of managing 3Com devices using RMON and Transcend tools, see the user documentation for the 3Com Transcend Network Control Services for Windows suite of applications.

Overview of RMON

RMON provides a way to monitor and analyze a local area network (LAN) from a remote location. The Internet Engineering Task Force (IETF) defines RMON-1 (RMON Version 1) in documents RFC 1271 and RFC 1757; RFC 2021 defines the extension of RMON-1, RMON-2 (RMON Version 2).

A typical RMON implementation has two components:

- **Your system** — Your system's built-in probe functionality examines all the LAN traffic on its segments, and keeps a summary of statistics (including historical data) in its local memory.
- **Management station** — Communicates with your system and collects the summarized data from it. The station can be on a different network from the system and can manage the system's probe function through either in-band or out-of-band connections.

The RMON specification consists almost entirely of the definition of the MIB. The RMON MIB contains standard MIB variables that are defined to collect comprehensive network statistics that alert you to significant network events. If the embedded RMON agent operates full time, it collects data on the correct port when the relevant network event occurs.

RMON Benefits

From a network management console, traditional network management applications poll network devices such as switches, bridges, and routers at regular intervals. The console gathers statistics, identifies trends, and highlights network events. The console polls network devices constantly to determine if the network is within its normal operating conditions.

As network size and traffic levels grow, however, the network management console can become overburdened by the amount of data it must collect. Frequent console polling also generates significant network traffic that itself can create problems for the network.

The RMON implementation in your system offers solutions to both of these problems:

- The system examines the network without affecting the characteristics and performance of the network.
- The system can report by exception rather than by reporting constant or frequent information. That is, the system informs the network management console directly if the network enters an abnormal state. The console can then use more information gathered by the system, such as historical information, to diagnose the abnormal condition.

RMON in Your System

Your system supports RMON as follows:

- **RMON-1 support** — The system software offers full-time embedded RMON support using SNMP for seven RMON-1 groups. (RMON-1 defines 10 groups.)
- **FDDI extensions** — The system software offers full support for two FDDI groups: axFddistatistics and axFddihistory. On FDDI modules, these supplement the RMON-1 statistics and history groups.
- **RMON-2 support** — The system software offers embedded RMON support for seven RMON-2 groups. (RMON-2 defines ten groups.) RMON-2 enables the system RMON feature to see above the MAC layer and monitor traffic based on network-layer protocols and addresses.

The embedded RMON support software cannot *receive RMON-2 updates for IP, IPX, and AppleTalk unless you have identified and configured a VLAN protocol type.*

3Com Transcend RMON Agents

RMON requires one probe per LAN segment. Because a segment is a portion of the LAN that is separated by a bridge or router, the cost of implementing many probes in a large network can be high.

To solve this problem, 3Com has built an inexpensive RMON probe into the Transcend SmartAgent software in each system. With this probe you deploy RMON widely around the network at a cost of no more than the cost of traditional network monitors.

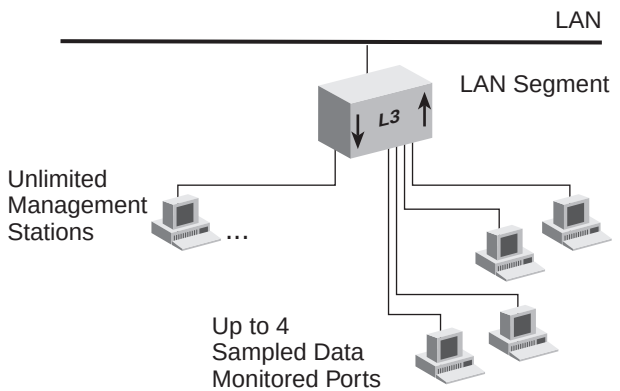
Placing probe functionality inside the system has these advantages:

- You can integrate RMON with normal device management.
- The system can manage conditions proactively.

The system associates statistics with individual ports and then takes action based on these statistics. For example, the system can generate a log event and send an RMON trap if errors on a port exceed a threshold set by the user.

Figure 96 shows an example of the RMON implementation.

Figure 96 Embedded RMON Implemented on the System



Important Considerations

- To manage RMON, you must assign an IP address to the system. See Chapter 11 for information about managing IP interfaces.
- The system will always keep RMON statistics (group 1) data on all ports.
- The system will keep RMON-1 history (group 2), alarm (group 3), and event (group 9) data on as many ports as its resources allow.
- The system will keep RMON-2 protocolDir (group 11), protocolDist (group 12), and probeConfig (group 19) data on as many ports as its resources allow.
- All other RMON group data is hardware sampled. The system can be configured to keep hardware-sampled RMON group data on up to four ports per CoreBuilder 3500 system.
- No RMON data is kept for Gigabit Ethernet modules.
- There is no limit to the number of network management stations monitoring the data.

RMON-1 Groups

The system supports seven of the RMON-1 groups that the IETF defines. Table 64 briefly describes these groups.

Table 64 RMON-1 Groups Supported in the System

RMON-1 Group	Group Number	Purpose
Statistics	1	Maintains utilization and error statistics for the segment being monitored
History	2	Gathers and stores periodic statistical samples from the statistics group
Alarm	3	Allows you to define thresholds for any MIB variable and trigger alarms
Host	4	Discovers new hosts on the network by keeping a list of source and destination physical addresses that are seen in good packets
HostTopN	5	Allows you to prepare reports that describe the hosts that top a list sorted by one of their statistics
Matrix	6	Stores statistics for conversations between pairs of addresses
Event	9	Allows you to define actions (generate traps, log alarms, or both) based on alarms

The system also supports the RMON/FDDI extension groups that the AXON Enterprise-specific MIB specifies. See Table 65.

Table 65 RMON/FDDI Extension Groups

Group	Group Number	Purpose
axFddiStatistics	axFddi group 1	Maintains utilization and error statistics for the monitored segment
axFddiHistory	axFddi group 2	Gathers and stores periodic statistical samples from the statistics group

Statistics and axFddiStatistics Groups

The statistics and axFDDIStatistics groups record frame statistics for Ethernet and FDDI interfaces. The information available per interface segment includes:

- Number of received octets
- Number of received packets

- Number of received broadcast packets
- Number of received multicast packets
- Number of received packets with CRC or alignment errors
- Number of received packets that are undersized but otherwise well-formed
- Number of received packets that are oversized but otherwise well-formed
- Number of received undersized packets with either a CRC or an alignment error
- Number of detected transmit collisions

Byte sizes include the 4-byte FCS, but exclude the framing bits. Table 66 lists the ethernet packet length counters that are implemented in the RMON-1 statistics group to keep track of the frame sizes that are encountered.

Table 66 Supported Frame Sizes for Ethernet and FDDI

Frame Lengths (Bytes)	
Ethernet	**FDDI**
64	22 or fewer
65 – 127	23 – 63, 64 – 127
128 – 511	128 – 511
512 – 1023	512 – 1023
1024 – 1518 (1024 – 1522 bytes when tagging is enabled)	1024 – 2047, 2048 – 4095

History and axFDDIHistory Groups

The history and axFDDIHistory groups record periodic statistical samples for Ethernet and FDDI interfaces and store them for later retrieval. The information available per interface for each time interval includes:

- Number of received octets
- Number of received packets
- Number of received broadcast packets
- Number of received multicast packets
- Number of received packets with CRC or alignment errors

- Number of received packets that are undersized but otherwise well-formed
- Number of received packets that are oversized but otherwise well-formed
- Number of received undersized packets with either a CRC or an alignment error
- Number of detected transmit collisions
- Estimate of the mean physical layer network utilization

Alarm Group

The system supports the following RMON alarm mechanisms:

- Counters
- Gauges
- Integers
- Timeticks

These RMON MIB objects yield alarms when the network exceeds predefined limits. The most frequently used objects are *counters*, although the other objects may be used in much the same way. The balance of this chapter illustrates RMON functions using counters.

Counters hold and update the number of times an event occurs on a port, module, or switch. *Alarms* monitor the counters and report when counters exceed their set threshold.

Counters are useful when you compare their values at specific time intervals to determine rates of change. The time intervals can be short or long, depending on what you measure.

Occasionally, counters can produce misleading results. Because counters are finite, they are useful for comparing rates. When counters reach a predetermined limit, they *roll over* (that is, return to 0). A single low counter value may accurately represent a condition on the network. On the other hand, the same value may simply indicate a rollover.

When you disable a port, the application may not update some of its associated statistics counters.

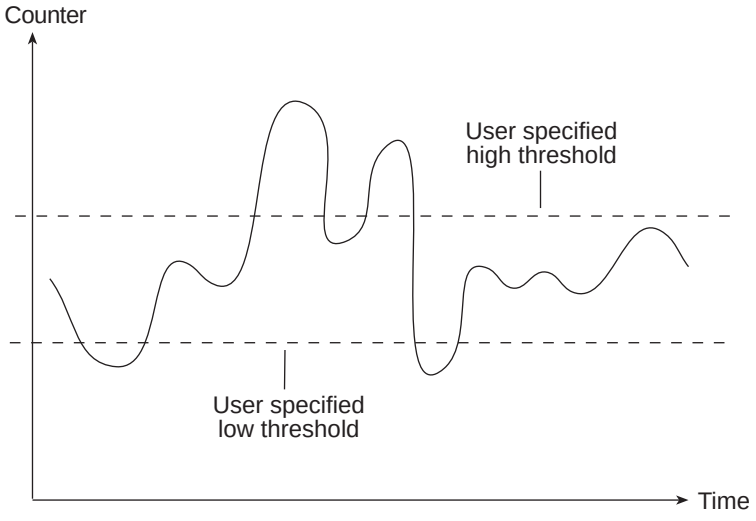
An alarm calculates the difference in counter values over a set time interval and remembers the high and low values. When the value of a counter exceeds a preset threshold, the alarm reports this occurrence.

Using Transcend Network Control Services or any other SNMP network management application, you can assign alarms to monitor any counter, gauge, timetick, or integer. See the documentation for your management application for details about setting up alarms.

Setting Alarm Thresholds Thresholds determine when an alarm reports that a counter has exceeded a certain value. You can set alarm thresholds manually through the network, choosing any value for them that is appropriate for your application. The network management software monitors the counters and thresholds continually during normal operations to provide data for later calibration.

Figure 97 shows a counter with thresholds set manually.

Figure 97 Manually Set Thresholds



You can associate an alarm with the high threshold, the low threshold, or both. The actions that occur because of an alarm depend on the network management application.

RMON Hysteresis Mechanism The RMON hysteresis mechanism prevents small fluctuations in counter values from causing alarms. Alarms occur only when either:

- The counter value exceeds the high threshold after previously falling below the low threshold. (An alarm does not occur if the value has not fallen below the low threshold before rising above the high threshold.)
- The counter value falls below the low threshold after previously exceeding the high threshold. (An alarm does not occur if the value has not first risen above the high threshold.)

For example, in Figure 97, an alarm occurs the first time that the counter exceeds the high threshold, but not the second time. At the first instance, the counter is rising from below the low threshold. In the second instance, the counter is not rising from below the low threshold.

Host Group

The host group records the following statistics for each host (the host group detects hosts on the network by their physical MAC addresses):

- Number of received packets
- Number of transmitted packets
- Number of received octets
- Number of transmitted octets
- Number of transmitted broadcast packets
- Number of transmitted multicast packets

These statistics, indexed by relative order in which the hosts are discovered, appear in *hostTimeTable*.

HostTopN Group

The HostTopN group reports on hosts that top a list that is sorted in order of one of their statistics. Information from this group includes:

- Number of received packets
- Number of transmitted packets
- Number of received octets
- Number of transmitted octets
- Number of transmitted broadcast packets
- Number of transmitted multicast packets

Matrix Group

The matrix group records the following statistics about conversations between sets of addresses:

- Number of packets transmitted from the source address to the destination address
- Number of octets, excluding errors, transmitted from the source address to the destination address
- Number of bad packets transmitted from the source address to the destination address

Event Group

The event group logs alarms or traps network event descriptions. Although alarm group thresholds trigger most events, other RMON groups may define event conditions.

RMON-2 Groups

The system software supports seven RMON-2 groups defined by the IETF in RFC 2021 and one object from the probe configuration group. Table 67 briefly describes these groups.

Table 67 RMON-2 Groups Supported in the System

RMON-2 Group	Group Number	Purpose
protocolDir	11	Provides a list of all protocols that the probe can interpret (protocols for which the probe can decode and count packets). The protocols can be different network-, transport-, and higher-layer protocols. This group allows the addition, deletion, and configuration of entries in the list.
protocolDist	12	Maintains a table of aggregate statistics on the amount of traffic generated by each protocol, per LAN segment (not for each host or application running on each host).
AddressMap	13	Maintains a table that maps each network address to a specific MAC address and port on an attached device and the physical address on the subnetwork
nlHost	14	Provides network-layer host statistics on the amount of traffic going in and out of hosts based on network-layer address

Table 67 RMON-2 Groups Supported in the System (continued)

RMON-2 Group	Group Number	Purpose
nlMatrix	15	A network-layer matrix that provides statistics on the amount of traffic between source/destination pairs of hosts based on network-layer address. It also maintains a TopN table to rank pairs of hosts based on the number of octets or number of packets sent between pairs of hosts.
alHost	16	Traffic statistics to and from each host by application layer. Same as nlHost except that traffic broken down by protocols can be recognized by ProtocolDir
alMatrix	17	Traffic statistics on conversations between pairs of hosts, segmented by application layer protocol.
probeConfig. probeCapabilities	19	Defines standard parameters that control the configuration of RMON probe functionality. The system currently supports only the probeCapabilities object from this group.

Protocol Directory Group

The protocolDir group provides information about the protocols that a particular RMON probe has or can interpret. It provides a common method of storing information about the protocols and makes it easier for a manager to monitor traffic above the MAC layer. It includes the base-level protocols such as IP, IPX, and AppleTalk, as well as higher-level protocols such as UDP, TCP, UDP-SNMP, and so forth.

This group features a protocol directory table that has an entry for each protocol. This enables the probe to decode and count protocol data units (PDUs). Information in the table includes the following:

- A protocol identifier (a unique octet string for a specific protocol)
- Protocol parameters (information about the probe's capabilities for a specific protocol)

Protocol Distribution Group

The protocolDist group tracks how many octets and packets the supported protocols have sent. It features two tables, a protocol distribution control table that manages the collection of the statistics for the supported protocols, and a protocol distribution statistics table that records the statistics. In the control table, each row represents a network interface associated with the probe and controls rows in the statistics table (a row for each protocol associated with the interface).

The protocolDist tables keep statistics for the three base protocols: IP, IPX, and AppleTalk. Unlike the protocolDir group, it does not keep statistics for higher-level protocols such as UDP, TCP, UDP-SNMP, and so forth.

The protocol distribution statistics table includes the following statistics:

- The number of packets received for each protocol
- The number of octets transmitted to this address since it was added to the network-layer host table

Address Map Group

The addressMap group maps each network address to a specific MAC-level address and to a specific port on the network device. This group provides three scalar objects (to track address-mapping entry insertions, deletions, and the maximum number of entries), an address map control table, and an address map data table.

Unlike other RMON control tables and data tables, the address map data table is not indexed by a row of the control table. The data table has entries that enable the mapping between the network addresses (normally IP addresses) and MAC addresses. The control table has an entry for each subnetwork connected to the probe so that addresses can be discovered on a new subnetwork and address mapping entries can be placed in the data table.

Network-Layer Host Group

The nlHost group gathers statistics about packets based on their network-layer address. (The RMON-1 host group gathers statistics based on MAC address.)

This group features a host control table and a host data table.

Network-Layer Matrix Group

The nlMatrix group gathers statistics about pairs of hosts based on network-layer address. (The RMON-1 matrix group gathers statistics based on MAC address.)

This group features two control tables and three data tables. One control table and its data tables collect matrix statistics; the other control table and its data table collect TopN statistics (reports describing hosts that top a list).

Application-Layer Host Group

The alHost group gathers statistics about IP packets over a monitored port based on their protocol. (The RMON-2 network-layer matrix group gathers statistics based on the network address.)

This group features a host data table.

Application-Layer Matrix Group

The alMatrix group gathers statistics about pairs of hosts conversing over a monitored port based on protocol. (The RMON-2 network-layer matrix group gathers statistics based on the network address.)

This group features one control table and three data tables. The alMatrix SD and alMatrix DS tables monitor traffic flows per conversation over monitored ports. The control table and its data table collect TopN statistics (reports describing hosts that top a list).

Probe Configuration Group Capabilities

The probeConfig group outlines a standard set of configuration parameters for RMON probes. Currently, your system supports one object in the probeConfig group, the probeCapabilities object. The function of this object is to identify the RMON groups that the probe supports.

Management Information Base (MIB)

This section provides information on the Management Information Base (MIB). A MIB is a structured set of data that describes the way that the network is functioning. The management software, known as the *agent*, gains access to this set of data and extracts the information it needs. The agent can also store data in the MIB. The following topics are covered:

- MIB Files
- Compiler Support
- MIB Objects
- MIB Tree
- MIB-II
- RMON-1 MIB
- RMON-2 MIB
- 3Com Enterprise MIBs

MIB Files

The organization of a MIB allows a Simple Network Management Protocol (SNMP) network management package, such as the Transcend Network Control Services application suite, to manage a network device without having a specific description of that device. 3Com ships the following MIB files with Extended System software as ASN.1 files:

- **BRIDGE-MIB.mib** — Bridge MIB, RFC 1493

  Unsupported groups and tables in this MIB:

  - dot1dSr group (see SOURCE-ROUTING-MIB.mib below)
  - dot1dStatic group
- **ETHERNET.mib** — Ethernet MIB, RFC 1398
- **FDDI-SMT73-MIB.mib** — FDDI SMT 7.3 MIB, RFC 1512
  - dot3CollTable
  - dot3Test group
  - dot3Errors group
  - dot3ChipSets group
- **FDDI-MIB.mib** — FDDI Station Management MIB, RFC 1285
- **IANAifType-MIB-V1SMI.mib** — Internet Assigned Numbers Authority MIB, SMI Version 1, RFC 1573

- **IF-MIB-V1SMI.mib** — Interface MIB, SMI Version 1, RFC 1573

  Unsupported tables in this MIB:

  - ifTestTable
  - ifRcvAddressTable
  - ifHC 64-bit counters
- **MIB2-MIB.mib** — MIB-II MIB, RFC 1213

  Unsupported groups and tables in this MIB:

  - egp group
- **OSPF-MIB.mib** — OSPF MIB, RFC 1850
- **RMON-MIB.mib** — RMON MIB, RFC 1757

RMON statistics for Gigabit Ethernet are not currently supported.

Supported groups in this MIB:

- statistics
- history
- alarm
- hosts
- hostTopN
- matrix
- event

- **axonFddiRmon.mib** — AXON RMON MIB, proprietary support
  - axFddiStatistics
  - axFddiHistory

- **RMON2-MIB-V1SMI.mib** — RMON v2, SMI Version 1 MIB, RFC 2021
  - protocolDir (RMONv2)
  - protocolDist (RMONv2)
  - addressMap (RMONv2)
  - nlHost (RMONv2)
  - nlMatrix (RMONv2)
  - alHost (RMONv2)
  - alMatrix (RMONv2)
  - probeCapabilities object of probeConfig group (RMONv2)

A maximum of four different ports can be configured for the following RMON groups at any given time:

  - addressMap
  - alHost
  - alMatrix
  - hosts
  - hostTopN
  - matrix
  - nlHost
  - nlMatrix
- **SNMPv2-MIB.mib** — used by other MIBs, RFC 1907
- **SOURCE-ROUTING-MIB.mib** — Source Routing Bridges MIB, RFC 1525
- **VRRP-MIB.mib** — Virtual Router Redundancy Protocol MIB, Draft RFC
- **3Com Enterprise MIBs** — See "3Com Enterprise MIBs" later in this chapter.

Compiler Support

Compiler Support ASN.1 MIB files are provided for these MIB compilers:

- SunNet Manager (version 2.0)
- SMICng (version 2.2.06)

MIB Objects

The data in the MIB consists of objects that represent features of the equipment that an agent can control and manage. Examples of objects in the MIB include a port that you can enable or disable and a counter that you can read.

A counter is a common type of MIB object used by RMON. A counter object may record the number of frames transmitted onto the network. The MIB may contain an entry for the counter object something like the one in Figure 98.

Figure 98 Example of an RMON MIB Counter Object

```
etherStatsPkts  OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION
                This is a total number of packets
                received, including bad packets,
                broadcast packets, and multicast
                packets.
        ::= {  etherStatsEntry 5 }
```

The counter object information includes these items:

- The name of the counter. In Figure 98, the counter is called *etherStatsPkts* (Ethernet, Statistics, Packets).
- Access level. In Figure 98, access is read-only.
- The number of the counter's column in the table. In Figure 98, the counter is in column 5 of the *etherStatsEntry* table.

The name of the table where the counter resides is *3CometherStatTable,* although this name does not appear in the display.

To manage a network, you do not need to know the contents of every MIB object. Most network management applications, including Transcend Network Control Services, make the MIB transparent. However, by knowing how different management features are derived from the MIB you can better understand how to use the information they provide.

MIBs include MIB-II, other standard MIBs (such as the RMON MIB), and vendors' private MIBs (such as enterprise MIBs from 3Com). These MIBs and their objects are part of the MIB tree.

MIB Tree The MIB tree is a structure that groups MIB objects in a hierarchy and uses an abstract syntax notation (ASN.1) to define manageable objects. Each item on the tree is assigned a number (shown in parentheses after each item), which creates the path to objects in the MIB. See Figure 99. This path of numbers is called the object identifier (OID). Each object is uniquely and unambiguously identified by the path of numeric values.

When the system software performs an SNMP Get operation, the management application sends the OID to the agent, which in turn checks to see if the OID is supported. If the OID is supported, the agent returns information about the object.

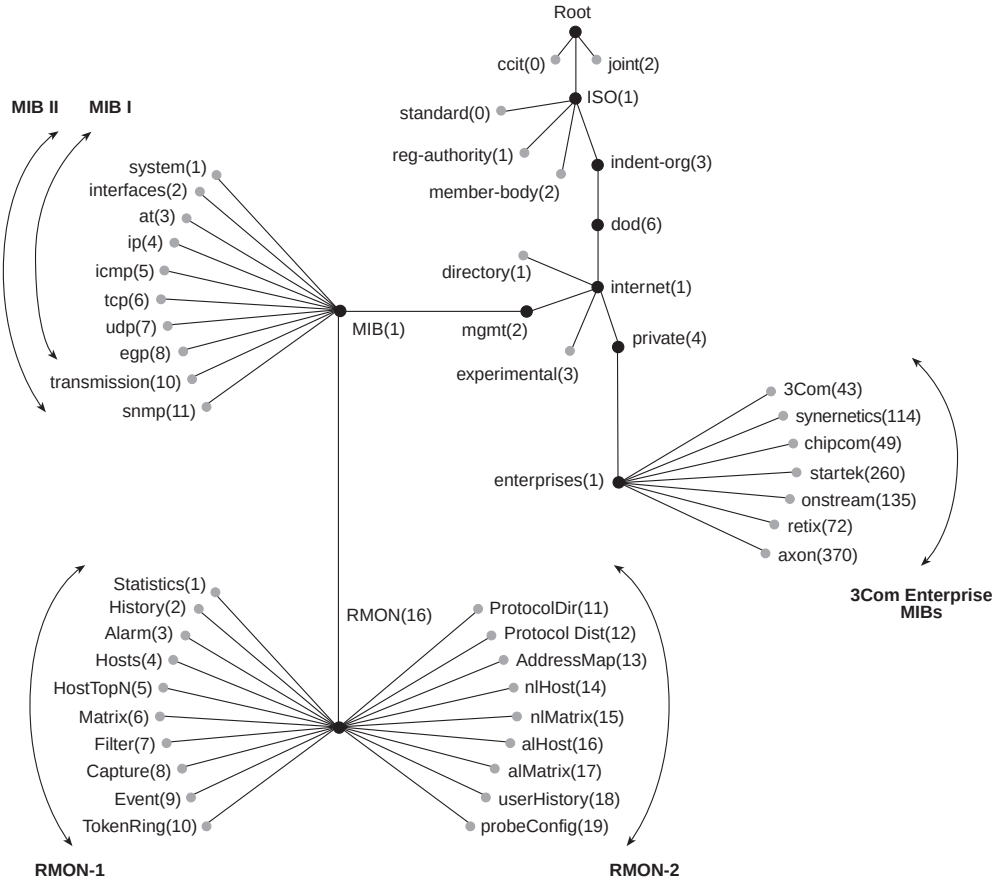
For example, to retrieve an object from the RMON MIB, the software uses this OID:

```
1.3.6.1.2.1.16
```

which indicates this path:

```
iso(1).indent-org(3).dod(6).internet(1).mgmt(2).mib(1).RMON(
16)
```

Figure 99 MIB Tree Showing Key MIBs



MIB-II MIB-II defines various groups of manageable objects that contain device statistics as well as information about the device, device status, and the number and status of interfaces.

The MIB-II data is collected from network devices using SNMP. As collected, this data is in its raw form. To be useful, data must be interpreted by a management application, such as Status Watch.

MIB-II, the only MIB that has reached Internet Engineering Task Force (IETF) standard status, is the one MIB that all SNMP agents are likely to support.

Table 68 lists the MIB-II object groups. The number following each group indicates the group's branch in the MIB subtree.

MIB-I supports groups 1 through 8; MIB-II supports groups 1 through 8, plus two additional groups.

Table 68 MIB-II Group Descriptions

MIB-II Group	Purpose
system(1)	Operates on the managed node
interfaces(2)	Operates on the network interface (for example, a port or MAC) that attaches the device to the network
at(3)	Were used for address translation in MIB-I but are no longer needed in MIB-II
ip(4)	Operates on the Internet Protocol (IP)
icmp(5)	Operates on the Internet Control Message Protocol (ICMP)
tcp(6)	Operates on the Transmission Control Protocol (TCP)
udp(7)	Operates on the User Datagram Protocol (UDP)
egp(8)	Operates on the Exterior Gateway Protocol (EGP)
transmission(10)	Applies to media-specific information (implemented in MIB-II only)
snmp(11)	Operates on SNMP (implemented in MIB-II only)

RMON-1 MIB

RMON-1 is a MIB that enables the collection of data about the network itself, rather than about devices on the network.

The IETF definition for the RMON-1 MIB specifies several groups of information. These groups are described in Table 69.

Table 69 RMON-1 Group Descriptions

RMON-1 Group	Description
Statistics(1)	Total LAN statistics
History(2)	Time-based statistics for trend analysis
Alarm(3)	Notices that are triggered when statistics reach predefined thresholds
Hosts(4)	Statistics stored for each station's MAC address
HostTopN(5)	Stations ranked by traffic or errors
Matrix(6)	Map of traffic communication among devices (that is, who is talking to whom)
Filter(7)	Packet selection mechanism
Capture(8)	Traces of packets according to predefined filters
Event(9)	Reporting mechanisms for alarms
Token Ring(10)	■ Ring Station — Statistics and status information associated with each token ring station on the local ring, which also includes status information for each ring being monitored ■ Ring Station Order — Location of stations on monitored rings ■ Source Routing Statistics — Utilization statistics derived from source routing information optionally present in token ring packets

RMON-2 MIB

RMON-1 and RMON-2 are complementary MIBs. The RMON-2 MIB extends the capability of the original RMON-1 MIB to include protocols above the MAC level. Because network-layer protocols (such as IP) are included, a probe can monitor traffic through routers attached to the local subnetwork.

Use RMON-2 data to identify traffic patterns and slow applications. The RMON-2 probe can monitor:

- The sources of traffic arriving by a router from another network.
- The destination of traffic leaving by a router to another network.

Because it includes higher-layer protocols (such as those at the application level), an RMON-2 probe can provide a detailed breakdown of traffic by application.

Table 70 shows the additional MIB groups available with RMON-2.

Table 70 RMON-2 Group Descriptions

RMON-2 Group	Description
Protocol Directory(11)	Lists the inventory of protocols that the probe can monitor
Protocol Distribution(12)	Collects the number of octets and packets for protocols detected on a network segment
Address Map(13)	Lists MAC-address-to-network-address bindings discovered by the probe, and the interface on which the bindings were last seen
Network Layer Host(14)	Counts the amount of traffic sent from and to each network address discovered by the probe
Network Layer Matrix(15)	Counts the amount of traffic sent between each pair of network addresses discovered by the probe
Application Layer Host(16)	Counts the amount of traffic, by protocol, sent from and to each network address discovered by the probe
Application Layer Matrix(17)	Counts the amount of traffic, by protocol, sent between each pair of network addresses discovered by the probe
User History(18)	Periodically samples user-specified variables and logs the data based on user-defined parameters
Probe Configuration(19)	Defines standard configuration parameters for RMON probes

3Com Enterprise MIBs

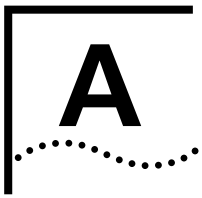
3Com Enterprise MIBs allow you to manage unique and advanced functionality of 3Com devices. These MIBs are shipped with your system on the *Software and Documentation CD-ROM*. Figure 99 shows some of the 3Com Enterprise MIB names and numbers. The following MIBs are included in 3Com(43).

- **3cFddi.mib** — 3Com FDDI MIB (43.29.10)
- **3cFilter.mib** — 3Com Packet Filtering MIB, standard and custom (43.29.4.20)
- **3cPolicy.mib** — 3Com Policy Management MIB (43.29.4.23)
- **3cPoll.mib** — 3Com Remote Polling MIB (43.29.4.22)
- **3cProd.mib** — 3Com Transcend Product Management MIB (43.1)
- **3cQos.mib** — 3Com QoS MIB (43.29.4.21)
- **3cSys.mib** — 3Com System MIB (43.29.4)

  Unsupported groups in this MIB:

  - a3ComSysSlot
  - a3ComSysControlPanel
  - a3ComSysSnmp
- **3cSysBridge.mib** — 3Com Bridging MIB (43.29.4.10)
- **3cSysFt.mib** — 3Com File Transfer MIB (43.29.4.14)
- **3cSysSmt.mib** — 3Com SMT MIB (43.29.4.9)
- **3cTrunk.mib** — 3Com Port Trunking MIB (43.10.1.15.1)
- **3cVlan.mib** — 3Com VLAN MIB (43.10.1.14.1)
- **3cWeb.mib** — 3Com Web Management MIB (43.29.4.24)

MIB names and numbers are usually retained when organizations restructure their businesses; therefore, some of the 3Com Enterprise MIB names may not contain the word "3Com."



TECHNICAL SUPPORT

3Com provides easy access to technical support information through a variety of services. This appendix describes these services.

Information contained in this appendix is correct at time of publication. For the most recent information, 3Com recommends that you access the 3Com Corporation World Wide Web site.

Online Technical Services

3Com offers worldwide product support 24 hours a day, 7 days a week, through the following online systems:

- World Wide Web site
- 3Com Knowledgebase Web Services
- 3Com FTP site
- 3Com Bulletin Board Service (3Com BBS)
- 3Com Facts℠ Automated Fax Service

World Wide Web Site

To access the latest networking information on the 3Com Corporation World Wide Web site, enter this URL into your Internet browser:

`http://www.3com.com/`

This service provides access to online support information such as technical documentation and software, as well as support options that range from technical education to maintenance and professional services.

3Com Knowledgebase Web Services

This interactive tool contains technical product information compiled by 3Com expert technical engineers around the globe. Located on the World Wide Web at **`http://knowledgebase.3com.com`**, this service gives all 3Com customers and partners complementary, round-the-clock access to technical information on most 3Com products.

3Com FTP Site

Download drivers, patches, software, and MIBs across the Internet from the 3Com public FTP site. This service is available 24 hours a day, 7 days a week.

To connect to the 3Com FTP site, enter the following information into your FTP client:

- Hostname: **`ftp.3com.com`**
- Username: **`anonymous`**
- Password: **`<your Internet e-mail address>`**

You do not need a user name and password with Web browser software such as Netscape Navigator and Internet Explorer.

3Com Bulletin Board Service

The 3Com BBS contains patches, software, and drivers for 3Com products. This service is available through analog modem or digital modem (ISDN) 24 hours a day, 7 days a week.

Access by Analog Modem

To reach the service by modem, set your modem to 8 data bits, no parity, and 1 stop bit. Call the telephone number nearest you:

Country	Data Rate	Telephone Number
Australia	Up to 14,400 bps	61 2 9955 2073
Brazil	Up to 28,800 bps	55 11 5181 9666
France	Up to 14,400 bps	33 1 6986 6954
Germany	Up to 28,800 bps	4989 62732 188
Hong Kong	Up to 14,400 bps	852 2537 5601
Italy	Up to 14,400 bps	39 2 27300680
Japan	Up to 14,400 bps	81 3 5977 7977
Mexico	Up to 28,800 bps	52 5 520 7835
P.R. of China	Up to 14,400 bps	86 10 684 92351
Taiwan, R.O.C.	Up to 14,400 bps	886 2 377 5840
U.K.	Up to 28,800 bps	44 1442 438278
U.S.A.	Up to 53,333 bps	1 847 262 6000

Access by Digital Modem

ISDN users can dial in to the 3Com BBS using a digital modem for fast access up to 64 Kbps. To access the 3Com BBS using ISDN, call the following number:

1 847 262 6000

3Com Facts Automated Fax Service

The 3Com Facts automated fax service provides technical articles, diagrams, and troubleshooting instructions on 3Com products 24 hours a day, 7 days a week.

Call 3Com Facts using your Touch-Tone telephone:

1 408 727 7021

Support from Your Network Supplier

If you require additional assistance, contact your network supplier. Many suppliers are authorized 3Com service partners who are qualified to provide a variety of services, including network planning, installation, hardware maintenance, application training, and support services.

When you contact your network supplier for assistance, have the following information ready:

- Product model name, part number, and serial number
- A list of system hardware and software, including revision levels
- Diagnostic error messages
- Details about recent configuration changes, if applicable

If you are unable to contact your network supplier, see the following section on how to contact 3Com.

Support from 3Com

If you are unable to obtain assistance from the 3Com online technical resources or from your network supplier, 3Com offers technical telephone support services. To find out more about your support options, call the 3Com technical telephone support phone number at the location nearest you.

When you contact 3Com for assistance, have the following information ready:

- Product model name, part number, and serial number
- A list of system hardware and software, including revision levels
- Diagnostic error messages
- Details about recent configuration changes, if applicable

Here is a list of worldwide technical telephone support numbers:

Country	Telephone Number	Country	Telephone Number
Asia, Pacific Rim			
Australia	1 800 678 515	P.R. of China	10800 61 00137 or
Hong Kong	800 933 486		021 6350 1590
India	+61 2 9937 5085	Singapore	800 6161 463
Indonesia	001 800 61 009	S. Korea	
Japan	0031 61 6439	From anywhere in S. Korea:	00798 611 2230
Malaysia	1800 801 777	From Seoul:	(0)2 3455 6455
New Zealand	0800 446 398	Taiwan, R.O.C.	0080 611 261
Pakistan	+61 2 9937 5085	Thailand	001 800 611 2000
Philippines	1235 61 266 2602		
Europe			
From anywhere in Europe, call:	+31 (0)30 6029900 phone +31 (0)30 6029999 fax		
Europe, South Africa, and Middle East			
From the following countries, you may use the toll-free numbers:			
Austria	0800 297468	Netherlands	0800 0227788
Belgium	0800 71429	Norway	800 11376
Denmark	800 17309	Poland	00800 3111206
Finland	0800 113153	Portugal	0800 831416
France	0800 917959	South Africa	0800 995014
Germany	0800 1821502	Spain	900 983125
Hungary	00800 12813	Sweden	020 795482
Ireland	1800 553117	Switzerland	0800 55 3072
Israel	1800 9453794	U.K.	0800 966197
Italy	1678 79489		
Latin America			
Argentina	AT&T +800 666 5065	Mexico	01 800 CARE (01 800 2273)
Brazil	0800 13 3266	Peru	AT&T +800 666 5065
Chile	1230 020 0645	Puerto Rico	800 666 5065
Colombia	98012 2127	Venezuela	AT&T +800 666 5065
North America	1 800 NET 3Com (1 800 638 3266) Enterprise Customers: 1 800 876-3266		

Returning Products for Repair

Before you send a product directly to 3Com for repair, you must first obtain an authorization number. Products sent to 3Com without authorization numbers will be returned to the sender unopened, at the sender's expense.

To obtain an authorization number, call or fax:

Country	Telephone Number	Fax Number
Asia, Pacific Rim	+ 65 543 6500	+ 65 543 6348
Europe, South Africa, and Middle East	+ 31 30 6029900	+ 31 30 6029999
Latin America	1 408 326 2927	1 408 326 3355
From the following countries, you may call the toll-free numbers; select option 2 and then option 2:		
Austria	0800 297468	
Belgium	0800 71429	
Denmark	800 17309	
Finland	0800 113153	
France	0800 917959	
Germany	0800 1821502	
Hungary	00800 12813	
Ireland	1800553117	
Israel	1800 9453794	
Italy	1678 79489	
Netherlands	0800 0227788	
Norway	800 11376	
Poland	00800 3111206	
Portugal	0800 831416	
South Africa	0800 995014	
Spain	900 983125	
Sweden	020 795482	
Switzerland	0800 55 3072	
U.K.	0800 966197	
U.S.A. and Canada	1 800 NET 3Com (1 800 638 3266)	1 408 326 7120 (not toll-free)
	Enterprise Customers: 1 800 876 3266	

INDEX

Numbers

3Com bulletin board service (3Com BBS) 568
3Com enterprise MIBs 565
3Com Facts 569
3Com Knowledgebase Web Services 567
499 (default classifier) 484
802.1p standard 478
  priority tags 478
802.1Q tagging 159, 348

A

AARP (AppleTalk Address Resolution Protocol) 456, 462
accept opcode 232, 233
access method
  IP 38
  modem port 38
  terminal port 38
Add Trusted Client 50
address
  classes 265
  filters 237
  IP 283
  MAC 263
  network 264
address ranges, OSPF 376
address table 116
address/port patterns
  limits 485, 488
  specifying for flow classifiers 487
addresses
  aging 116
  destination 116
  for SNMP trap reporting 539
  source 116
  specifying for flow classifiers 487
addresses, AppleTalk 462, 463
  interface 458
addressing scheme, OSPF 369
addressMap group, RMON V2 554
addressThresholdEvent 138
adjacencies, OSPF 366
Administration Console 46
  accessing 32, 40
  accessing the modem port 39
  managing from 32
  password levels 40
  sample menu output 32
  top-level menu 41
ADSP (AppleTalk Data Stream Protocol) 453
advertise RIP mode 297
advertisement address 298
advertising
  IEEE 802.1Q VLANs 183
AEP (AppleTalk Echo Protocol) 452, 456, 469
aggregated links
  Ethernet 73
alarm thresholds, RMON
  examples of 550
  setting 550
allClosed mode
  and network-based VLANs 193
  and port-based VLANs 180
  and protocol-based VLANs 186
  egress rules 198
  ingress rules 197
  selecting 169
allOpen mode
  and network-based VLANs 193
  and port-based VLANs 180
  and protocol-based VLANs 186
  egress rules 198
  ingress rules 197
  selecting 169, 171
anchor port (in trunk) 145
and (bit-wise AND) opcode 231
anycast client 56
AppleTalk
  Address Resolution Protocol (AARP) 462
  addresses 462, 463
  and OSI Reference Model 448
  benefits of 447
  changing zones 467
  checksum 469
  data link layer 449
  data stream protocol (ADSP) 453

hop count 456
interface address 458
interface states 458
interfaces 459, 460
elements of 458
key guidelines for configuring 457
Management Information Base II 474
Name Binding (NBP) 452
network devices 459
network layer 449
network ranges 458, 460
networks 454
node number assignment 459
nodes 454
nonseed routers 455
overview 445
packet filter 234
phase 1 networks 456
phase 2 networks 456
physical layer 449
presentation layer 453
printer access protocol (PAP) 453
protocols
about 445
and OSI levels 448
route flapping 461
routes 460
routing 445
Routing Table Maintenance Protocol (RTMP) 450
routing tables 452, 460
seed interfaces 458
seed routers 455
session layer 450
session layer protocol (ASP) 453
statistics 470
system features 446
traffic forwarding 468
transport layer 450
Zone Information Protocol (ZIP) 453
Zone Information Table (ZIT) 453, 461
zones 455, 458, 464, 465
AppleTalk Address Resolution Protocol (AARP) 456, 462
AppleTalk Echo Protocol (AEP) 452, 456, 469
AppleTalk interfaces
considerations and guidelines for configuring 459
AppleTalk interfaces, and VLANs 458
AppleTalk protocols
for VLANs 187
AppleTalk Session Protocol (ASP) 453
AppleTalk Transaction Protocol (ATP) 452
area border routers 364, 375, 378, 387, 389
area IDs, OSPF 385
areas 358, 361, 363, 372, 376
backbone 373, 377
backbone, OSPF 385
stub 373, 377, 400
transit 373
ARP (Address Resolution Protocol)
cache 285, 286
defined 286
location in OSI Reference Model 261
reply 287
request 287
ASBRs 370
ASCII-based editor
for packet filters 217
ASP (AppleTalk Session Protocol) 453
ATP (AppleTalk Transaction Protocol) 452
attachments
single and dual 94
authentication, OSPF 361, 384
autonegotiation, Ethernet 80
autonomous system boundary routers (ASBRs), OSPF 370
autonomous system boundary routers, OSPF 362, 389
autonomous systems 358, 363

B

backbone areas, OSPF 377, 385
backbone routers, OSPF 364
backup designated routers, OSPF 361, 364, 366
bandwidth
between servers and switches 73, 78
limiting with QoS 480, 489
QoS 481
reservation with RSVP 476
RSVP 518
to end stations 78
baseline
displaying current 525
enabling and disabling 525
reasons for 525
blocking port state 129
bridge
address threshold, setting 138
designated 119
IPX SNAP Translation 139
least cost path 120
root 119

Spanning Tree
bridge priority, setting 134
forward delay, setting 135
hello time, setting 134
maximum age, setting 134
bridge ports
associating with VLANs 176
in port-based VLANs 178
in protocol-based VLANs 186, 192
STP
enabling 136
path cost, setting 136
port priority, setting 136
bridging
and protocol-based VLANs 186
configuration messages 119
IEEE 802.1D compliant 142
sample VLAN configuration 189
standards 142
bridging and routing model 280
bridging rules
and VLANs 195, 198
broadcast address
description 295
security 295
bulletin board service 568
burst size, QoS control 480, 493
definition 480

C

cache, ARP 286
campus interconnects 78
Carrier Sense Multiple Access With Collision Detection (CSMA/CD) 84
cast types, for QoS classifiers
flow 486
nonflow 488
CBPDU
best 123
information 122
CD-ROM documentation 26
changing
default VLAN 176
port numbering via module removals 66
port numbering via module replacements 68
trunk ports via module replacements 69
VLAN ports via module replacements 69
checksum 456
configuring AppleTalk 469
Chooser, Macintosh 455
classifiers, QoS 482
assigning numbers 484
defining flow 485
defining nonflow 488
flow routing requirements 485, 488
predefined 483
restrictions 482
sample configurations 497, 499, 501, 504, 506, 508, 512
specifying ports and ranges 487
types of 479
using 483
collision, Ethernet 84
community strings
defined 538
values 539
compatibility mode 297
concentrator
dual-attachment 109
Configuration 50
configuration procedures and port numbering 61
configurations
dynamic VLAN via GVRP 182
Ethernet 73
sample GVRP 185
sample Ignore STP 173
sample network-based 194
sample port-based 181, 182
sample protocol-based 189, 191
sample QoS 497, 499, 501, 504, 506, 508, 512
sample RSVP 518
conforming packets 489
definition 480
continuous operation 73
controls, QoS 482, 492
assigning numbers 490
default 490
definition of 480
predefined 490
restrictions 482, 489
sample configurations 497, 499, 501, 504, 506, 508, 512
setting priorities 491
TCP drop control 494
timer option 491
timer option 481, 495
timer options 496
using 489
conventions
notice icons, About This Guide 22
text, About This Guide 22

convergence, OSPF 377
cost
  Spanning Tree settings 136
cost, OSPF 381
creating
  VLANs via GVRP 182
CSMA/CD (Carrier Sense Multiple Access With Collision Detection) 84
custom packet filters 214

D

DAS
  pairs and port numbering 61, 65
DAS (Dual Attached Station) 109
data centers 78
data link layer, AppleTalk 449
data link layer, IP 261
database description packets, OSPF 365
datagrams 470
DDP (Datagram Delivery Protocol) 449, 470
dead interval, OSPF 369, 384, 385, 386
DECnet protocols
  for VLANs 187
default classifier (499) 484
  restrictions 482
default control (1) 490
  restrictions 482
default route metrics, OSPF 358, 372, 379
default route, IP 273
  gateway address 285
default route, OSPF 379
default VLAN 167
  effects of trunking 177
  modifying 176
  removing 177
defaults
  multicast limit for bridge ports 140
defining IP interfaces 284
designated bridge 119
designated port 119, 120
designated routers, OSPF 361, 364, 367, 388
destination address
  for SNMP trap reporting 539
destination addresses 486
destination IP address masks 486
devices
  GVRP-enabled 182
DeviceView 33
DHCP (Dynamic Host Configuration Protocol) 311
directed broadcast 295
disabled
  port state 129
disabled RIP mode 297
distance, AppleTalk routes 460
distance-vector protocols 358
DNS (Domain Name Server) 310
DNS server problems 531
documentation
  comments 26
  defined 24
  for the system 24
  on CD-ROM 26
  orders 24
DPGM (destination port group mask) 242
drop service level 480
dual homing 97
dual ring, FDDI 93
dual-attachment stations 109
duplex mode, Ethernet ports 82
DVMRP
  multicast routing table 353
dynamic route, IP 273
dynamic routes, IPX 435
dynamic VLAN configuration 159, 182

E

edge policing, RSVP 518
editor
  for packet filters 217
egress rules 162, 195
  for transmit ports 199
EMACS editor 217
Embedded Web Management applications
  DeviceView 33
  Performance features 33
  WebConsole 33
empty slots, working with 63
enabled RIP mode 297
end stations
  bandwidth to 73
end stations, bandwidth to 78
enterprise MIBs 565
eq opcode 230
equation
  for calculating number of VLANs 163, 167
errors
  ICMP echo reply 290
  ICMP echo request 290
  ICMP redirect 290
  ICMP time exceeded 291
  ping 530

Ethernet 85
aggregated links 73
collision 84
configurations 73
CSMA/CD 84
definition 72
Fast Ethernet 72
frames, processing 76
Gigabit Ethernet 72
Gigabit Interface Converter (GBIC) 85
guidelines 73
link aggregation 73
media specifications 85
modules and port numbering 60
network capacity, recommendations 78
packet fields 210
ports 79
autonegotiation 80
duplex mode 82
managing 71
PACE Interactive Access 73, 84
speed 82
replacing modules 68
sample Fast Ethernet configuration (empty slot) 63
sample Fast Ethernet port numbering 62
sample Gigabit Ethernet configuration 64
standards (IEEE) 84, 154
trunks 73
event group, RMON 552
event log 524
configuring services 524
displaying configuration of 524
Exact Match 54
exception flooding 200
excess packet tagging 511
sample QoS configuration 512
excess packets
definition 480
tagging 481
export policies 300
extended memory 52
extended network numbers 454
extended network prefix 267
external link state advertisements, OSPF 389
external LSAs, and stub areas 400
external metrics, OSPF
type 1 390
type 2 390
external routes, OSPF 390

F

Fast Ethernet 72
ports, autonegotiation 80, 81
ports, duplex mode 82
ports, speed 82
sample port numbering 62
trunks 149
fax service (3Com Facts) 569
FDDI
and OSI model 90
dual homing 97
dual ring 93
MIB 98
modules and port numbering 60
nodes 94
optical bypass switch 109
overview 87
packet fields 210
port numbering 65
replacing modules 68
ring 92
sample VLAN configuration using 188
FDDI MAC
condition report 106
FrameErrorThreshold, setting 106
LLC Service, enabling 101, 107
NotCopiedThreshold, setting 106
FDDI path
maxT-Req, setting 105
tmaxLowerBound, setting 105
tvxLowerBound, setting 104
FDDI port
labeling 108
lerAlarm, setting 107
lerCutoff, setting 108
FDDI station
and SRFs 104
status reporting, enabling 104
T-notify, setting 104
feedback on documentation 26
fiber
multimode 85
singlemode 85
Fiber Distributed Data Interface (FDDI)
SNMP smtProxyTraps 540
File Transfer Protocol 46, 48
File Transfer Protocol (FTP) 45, 47
Filter Builder 33

filtering
for VLANs 198
IP multicast 331, 336
QoS 501
fixed filter style, RSVP 517
flooding 198
exception 200
samples of 200
flow classifiers
defining 485
definition of 479
IP and VLAN requirements 485
range of numbers 484
routing requirements 485, 488
specifying addresses and masks 487
specifying ports and ranges 487
flow control, Gigabit Ethernet ports 83
flows, RSVP 517
flush command
snmp trap 540
flushing
SNMP trap addresses 540
forward delay 135
forwarded frames
setting priority tags 481
forwarding
for VLANs 198
port state 129
forwarding and flooding decisions
for network-based VLANs 204
forwarding, AppleTalk traffic 468
frame-based protocols 98
FrameErrorThreshold
defined 106
frames
Ethernet, processing 76

G

GARP (Generic Attribute Registration Protocol) 159
gateway address 272
GBIC (Gigabit Interface Converter)
1000BASE-LX 85
1000BASE-SX 85
ge opcode 231
Gigabit Ethernet 72, 85
media specifications 85
port numbering changes when adding 68
ports, autonegotiation 80, 81
ports, flow control 83
sample port numbering 64
trunks 149
Gigabit Ethernet, and RMON 546
group address
Spanning Tree, setting 135
gt opcode 231
guidelines
configuration and port numbering 61
for accessing your system 38
key for configuring AppleTalk 457
QoS 482
GVRP (GARP VLAN Registration Protocol) 159
sample configuration 185
STP and 141, 183
using 182

H

Hello interval, OSPF 369, 385
hello packets, OSPF 365, 366, 395
hello time 134
high priority traffic
sample QoS configuration 504
hop count, AppleTalk 456
hop count, OSPF 360
host group, RMON 551
hostTopN group, RMON 551
hysteresis mechanism, RMON 551

I

ICMP (Internet Control Message Protocol)
description 290
location in OSI Reference Model 261
ICMP Redirect
description 292
example 293
ICMP Router Discovery
description 294
example 295
guidelines 294
IEEE 802.1p
recognizing priorities with classifiers 488
setting priorities with controls 481, 491
standard 478
IEEE 802.1Q 159
advertising VLANs 183
tagging rules 198
terms for VLAN modes 169
IEEE 802.1Q tagging 159, 348
IEEE Ethernet standards 84, 154
IETF
MIB-II MIB 562
OSPF 360
RMON MIB 563

IGMP
default setting 349
host membership reports 341
query mode 349
snooping mode 349
Ignore STP mode 161, 167, 170, 173
sample configuration 173
Implementing SNTP 56
import policies 300
in-band IP management interface 271
in-band management 37, 40
Independent VLAN Learning (IVL) 169
index, VLAN interface 283
ingress rules, VLANs 162, 195, 197
initialize state 319
instructions, packet filter
opcodes 226, 228
operands 226, 227
interface address, AppleTalk 458
interfaces
and VLANs 161
AppleTalk seed 458, 459
AppleTalk states 460
in-band versus out-of-band 271
IP 284, 312
VLAN 156
interfaces, OSPF 359, 384, 385
and OSPF areas 375
area ID 381
cost 381
dead interval 384
delay 382
elements of 380
mode 380
password 384
priority 380
retransmit interval 383
state 366
statistics 384
Interior Gateway Protocols (IGPs) 273, 431
internal routers, OSPF 364
international time standards 57
Internet MBONE 337
intranetwork routing 259
IP
ping functions 530, 532
traceRoute functions 532
IP (Internet Protocol)
addresses 264, 283
administering DNS 310
assigning addresses to in-band or out-of-band ports 39
broadcast address 295
defining a management interface 36
interfaces 284
management concepts 37
management interface 39
managing in-band 40
managing out-of-band 40
networking protocol 39
overlapped interfaces 312
UDP Helper 311
IP address
classes of 265
defined 264
derivation 264
division of network and host 264
DNS 310
example 266
network layer 261
next hop 261
RIP 296
routing table 272
subnet mask 265
subnetwork portion 265
IP addresses
and restoring NV data 54
flow classifier 486
pinging 531
IP hostnames
pinging 531
IP interfaces
defining 284
IP multicast
addressing 332, 339
benefits of 333
cache display 353
filtering 336
groups 338
MBONE 337
routing table 353
spanning tree 344
supported protocols 348
system displays 353
tunnels 350
IP multicast filtering 331, 336
IP multicast routing 331, 334
child interface 345
parent interface 345
pruning branches 346
IP multicast tunnels 350
default characteristics 350
defining end points 350
IP packets filter 248, 252
IP protocols
for VLANs 187

IP routing
address classes 265
administering 285
defining static routes 285
features and benefits 262
OSI reference model 261
router interface 271
routing table 272, 273
transmission process 261
types of routes 285
IPX
RIP policies 439
SAP policies 441
IPX protocols
for VLANs 187
IPX routing
packet format 422
IPX SNAP Translation 139

L

labeling Ethernet ports 79
LANs, virtual 156
Layer 3 addresses
for IP VLANs 162, 186
layers, OSI Reference Model 90
LDAP (Lightweight Directory Access Protocol)
ldif file 514
operation 515
overview 514
QoS parameters 514
server 514
le opcode 231
learn RIP mode 297
learning port state 129
learning state 135
LER (Link Error Rate)
alarm value 107
cutoff value 108
lerAlarm
defined 107
lerCutoff
and lerAlarm value 108
defined 108
limits, rate (QoS) 489
link aggregation
Ethernet 73
link data, OSPF 388
link state
acknowledge packets, OSPF 365
advertisements (LSAs), OSPF 363, 364, 366, 367, 372, 387
protocol, OSPF 358
request packets, OSPF 365
update packets, OSPF 365
link state age, OSPF 387
link state databases, OSPF 359, 383, 387
viewing 391
link state ID, OSPF 387
link state sequence, OSPF 387
listening port state 129, 135
LLC
service description 101, 107
log, event 524
logical topology 91
loss-eligible packets 480
lt opcode 230

M

M port 101
MAC (Media Access Control) 89, 99
addresses
and restoring NV data 54
description 263
in switching 280
IP address 264
located with ARP 286
use in IP routing 287
Macintosh, Chooser 455
management
port labels 108
SNMP community strings 539
management station
RMON MIB 542
management, IP interface 271
manual versus dynamic VLAN configuration 183
masks
flow classifier 486
masks, subnet 265, 283
Master port 101
matrix group, RMON 552
maximum age 134
maxT-Req
defined 105
MBONE 337
media
Ethernet 85
Gigabit Ethernet 85
memory partition 359
memory partition, OSPF 397
methods of using QoS 477
metric 272

metrics, OSPF 388
external
type 1 390
external type 2 390
MIB (Management Information Base)
FDDI 98
RMON 542, 556
MIB browser
viewing the tree 560
MIB-II
objects 562
MIBs 568
enterprise 565
example of OID 560
in SNMP management 534
MIB-II 562
RMON 563
RMON-2 564
tree representation 561
tree structure 560
modem port
access 41
establishing a connection 39
setting up 39
modes
allOpen 171
Ignore STP 170, 173
modifying VLAN 170
selecting VLAN 161, 167, 169
modifying
default VLAN 176
modules in system 66, 68
VLAN mode 170
VLANs 206
modules
effects of removals 66
effects of replacements 68, 70
empty slots 63
Ethernet 60, 62, 63
FDDI 65
mixing different types 64
port numbering 59
sample configurations 62
supported types 60
types 60
multicast frames
and packet filters 212
multicast server 56
multimedia traffic, handling with QoS 476
multimode fiber 85
multiple IP interfaces 263

N

Name Binding Protocol (NBP) 452
name opcode 228
named entities 455
names
for VLANs 162
NBP (Name Binding Protocol) 473
ne opcode 230
neighbor notification
and LLC Service 101, 107
neighbors, OSPF 359, 363, 365, 366, 367, 369, 383, 384, 392
guidelines for administering 395
static 395
viewing information 392
network
address 264
campus interconnects 78
capacity, recommendations 78
data centers 78
segmentation 262
wiring closets 78
network layer 261
network layer, AppleTalk 449
network link state advertisements, OSPF 388
network management platforms
defined 523
network numbers
extended 454
nonextended 454
network ranges 455, 458, 460
aging out of AppleTalk tables 466
changing 467
network supplier support 569
Network Time Protocol (NTP) 56
network topologies, FDDI 91
network troubleshooting 526
network-based VLANs
allOpen mode and 171
ingress rules 197
sample configuration 194
sample forwarding and flooding decisions 204
using 192
networks
and AppleTalk devices 459
AppleTalk phase 1 456
AppleTalk phase 2 456
connecting to AppleTalk phase 1 457
nlHost group, RMON V2 554
nlMatrix group, RMON V2 555
node number, AppleTalk 459

nodes
AppleTalk 454
FDDI 94
types 95
nonconforming excess packets
definition 480
nonextended network numbers 454
nonflow classifiers
defining 488
definition of 479
range of numbers 484
setting priorities 488
nonoverlapped VLANs
port-based 178, 181
protocol-based 186, 189, 192
nonseed routers, AppleTalk 455
not opcode 231
NotCopiedThreshold
defined 106
Notepad 217
Novell
in packet filter 238
null VLAN 199
number of VLANs 163, 167
numbering
physical port 62, 66, 68, 70
port overview 59, 60
numbers
QoS classifier 484
QoS control 490
NV data
and packet filters 214
contents saved 53
nvData Operations 45, 46
nvData save operation 53

O

OID
example 560
MIB tree 561
online technical services 567
opcode
and packet filter language 224
and writing packet filters 225
descriptions 228
operand 226
and opcodes 227
sizes supported 226
optical bypass switch
on intermediate systems 109
or opcode 231
OSI Reference Model 34, 261
and FDDI 90
AppleTalk routing and 448
OSPF
and imported RIP routes 404
OSPF (Open Shortest Path First)
addresses
addressing scheme 369
ranges 372, 376
adjacencies 366
area border routers 376, 378, 387, 389
areas 358, 361, 363
area IDs 381, 385
backbone 373, 377, 385
guidelines for configuring 376
parameters 372
authentication 361
autonomous systems 363
boundary routers 362, 387
benefits of 360
cost 381
dead interval 369, 384, 385, 386
default route metrics 358, 372, 379
default router 379
delay 382
designated routers 388
external link state advertisements 389
external routes 390
Hello interval 369, 385
hop count 360
importing non-OSPF routing information 362
interfaces 359, 375, 384
guidelines for configuring 385
parameters 369
parts of 380
state 366
statistics 384
key guidelines for implementing 369
link data 388
link state
acknowledge packets 365
advertisements (LSAs) 363, 364, 366, 367, 372, 387
databases 359, 383, 387
protocol 358
request packets 365
update packets 365
link state databases
viewing 391
link state sequence 387
location in OSI Reference Model 261
memory partition 359, 397
metric 388

mode 380
neighbors 359, 365, 366, 367, 369, 383, 384, 392
and adjacencies 363
static 395
viewing information 392
network link advertisements 388
packets
database description 365
Hello 365
hello 366, 395
password 369, 384, 386
path trees, shortest 367
priority 380, 385
protocol packets 365
protocols
Hello 395
retransmit interval 383, 386
route summarization 376
route support 361
router databases 375
router IDs 359, 365, 387, 396
guidelines for configuring 396
types of 396
router placement 369
router updates 360
routers
area border 364, 375
autonomous system boundary (ASBRs) 370
backbone 364
backup designated 361, 364, 366
designated 361, 364, 367
internal 364
routing
inter-area 368
intra-area 368
to different autonomous systems 368
to stub area 368
routing algorithm 366
routing policies 404
routing policies, OSPF 360, 403
and static routes 404
export 408, 411, 412, 415
import 405, 408
shortest path trees 367
soft restarts 398
statistics 360, 416
stub areas 372, 373, 377, 400
stub default metrics 359, 400
summary 358
summary link state advertisements 389
transit areas 373
transmit delay 385
type 1 external metrics 390
type 2 external metrics 390
types of routers 364
variable length subnet mask 362
virtual links 360, 362, 364, 368, 375, 377, 378, 387, 392, 401, 402
OUI
in packet filter 239
out-of-band management 37, 40, 271
overlapped IP interfaces 312
overlapped VLANs
port-based 178, 182
protocol-based 186, 192

P

PACE Interactive Access, Ethernet 73, 84
packet
Ethernet type 210
FDDI type 210
fields for operands 227
packet filter
basic elements 210
concepts 230
creating 214
custom 214
definitions 214
editor
commands 218
description 217
examples 247
filtering criteria, groups 242, 255
instructions 226
language description 214, 224
listing 215
opcodes 228
operands 226
port group example 242
predefined 220
procedure for writing 225
processing paths 216
pseudocode 248
run-time storage 241
sequential tests 233
stack 226
standard 213
storage space 241
syntax errors 235, 236
packets
conforming 480
excess 480
loss-eligible 480
tagging excess 481

PAP (Printer Access Protocols) 453
password, OSPF 369, 384, 386
passwords
  community strings 538
path cost
  defined 136
PCMCIA flash memory card 52
phase 1 networks, AppleTalk 456
  connecting to 457
phase 2 networks, AppleTalk 456
PHY standard
  defined 89
physical layer, AppleTalk 449
physical port numbering 59
physical topology 91
ping, AppleTalk 469
ping, strategies for using 531
pin-outs
  modem and terminal ports 39
platforms 523
PMD standard
  defined 89
poison reverse 298
policies
  IPX SAP 441
policies, OSPF routing 360, 403
  and imported RIP routes 404
  and static routes 404
  export 408, 411
    examples 412, 415
  import 405
  import examples 408
policing options, RSVP 518
policy-based services 476
port
  designated 119
  FDDI 100
  identifier 122
  maximum number in group 245
  root 119, 120
port group
  adding ports 245
  as filtering criteria 242, 255
  copying 245
  deleting 245
  displaying contents 245
  listing 245
  loading on system 246
  removing ports 246
  used in packet filter 242
port membership
  for VLANs 162
port numbering
  configuration guidelines 61
  effects of empty slots 63
  effects of module removals 66
  effects of module replacements 68, 70
  FDDI 65
  overview 59
  rules 60
port ranges
  guidelines for specifying 482
port state
  learning 129
  listening 129
port-based VLANs
  allOpen mode and 171
  dynamic configuration via GVRP 182
  sample configurations 181, 182
  using 175
ports
  anchor (in trunk) 145
  associating with rate limits 491
  bridging priority 136
  bridging states 129, 130
  Ethernet, autonegotiation 80
  Ethernet, duplex mode 82
  Ethernet, labeling 79
  Ethernet, PACE Interactive Access 73, 84
  Ethernet, speed 82
  Fast Ethernet, autonegotiation 80, 81
  Fast Ethernet, duplex mode 82
  Fast Ethernet, speed 82
  Gigabit Ethernet, autonegotiation 80, 81
  Gigabit Ethernet, flow control 83
  numbering
    sample configurations 62
  numbering, in a trunk 145
  path cost 136
  removing trunk 178
precedence
  indicating with classifier numbers 484
predefined packet filters 220
predefined QoS classifiers 483
predefined QoS controls 490
presentation layer, AppleTalk 453
primary IP address 319
Printer Access Protocol (PAP) 453
printer access protocol (PAP) 453
priorities
  IEEE 802.1p 478
    priority tags 478
  sample QoS configuration 506
  setting with controls 481, 491
  setting with nonflow classifiers 488

priority, OSPF 385
probe, RMON 542
probeConfig group, RMON V2 555
procedures
for establishing routing between VLANs 190
QoS 482
protocol packets, OSPF 365
protocol suites
for VLANs 162, 186, 187
unspecified 176, 178
protocol types
flow classifier 486
nonflow classifier 488
protocol-based VLANs
allOpen mode and 171
for bridging and routing 186
sample configuration 189, 191
sample flooding decisions 200
using 186
protocolDir group, RMON V2 553
protocolDist group, RMON V2 554
protocols
AppleTalk 450, 453
AppleTalk Address Resolution (AARP) 456, 462
AppleTalk Echo (AEP) 452, 456, 469
AppleTalk Session (ASP) 453
AppleTalk Transaction (ATP) 452
Datagram Delivery (DDP) 470
FDDI Station Management 35
frame-based 98
GARP and GVRP 159
Hello 395
Internet Protocol (IP) 39
Name Binding (NBP) 452, 473
Open Systems Interconnection (OSI) 34
printer access (PAP) 453
Routing Table Maintenance Protocol (RTMP) 450, 460, 465, 471
Simple Network Management Protocol (SNMP) 35
User Data Protocol (UDP) 37
using with classifiers 479
virtual terminal protocols 35, 36
Zone Information (ZIP) 465, 472
pruning
IP multicast 346
pushDPGM opcode 230, 242
pushField.size 228
pushLiteral. opcode 228
pushSPGM opcode 230, 242
pushTop opcode 229

Q

QoS (Quality of Service) 475
and RSVP 479
bandwidth 481, 513
burst size 480, 493
classifiers 479
assigning numbers 484
defining flow 485
defining nonflow 488
predefined 483
removing 510
specifying ports and ranges 487
using 483
controls 480, 492
assigning numbers 490
predefined 490
setting IEEE 802.1p priorities 491
using 489
excess packet tagging 481, 511
guidelines 482
IEEE 802.1p priority tags 478
management tools 475
methods 477
overview 476
related standards 478
sample configurations 497, 499, 501, 504, 506, 508, 512

R

ranges
classifier number 484
OSPF address 372
QoS control numbers 490
starting and ending 486
TCP or UDP 482, 486, 487
VLAN ID 160
rate limits, QoS 489
assigning 482
definition 480
redundant router connections 172
reject opcode 232, 233
Remove Trusted Client 50
removing
default VLAN 178
modules 66
trunk ports 178
VLANs 206
replacing
modules 68, 70
reservable bandwidth 518
reservation styles, RSVP 517

Restore 47
Restoring nvData 54
restrictions
  QoS 482
  QoS control 489
retransmit interval, OSPF 386
returning products for repair 571
reverse path multicasting (RPM)
  broadcasting 345
  grafting 346
  pruning 346
ring of trees 92
RIP (Routing Information Protocol) 358
  advertisement address 298
  compatibility mode 297
  defined 296
  location in OSI Reference Model 261
  poison reverse 298
  route aggregation 299
  route configuration 273
  routing policies 300
RIP routing policies
  administrative weight 302
  example 309
  explained 301
  IPX 439
  metric adjustment 302, 303
  parameters 307
  policy conditions 304
  policy conflicts 305
RIP-1 mode 297
RMON (Remote Monitoring) 542
  addressMap group 554
  agents 545
  alarms 549, 550
  and roving analysis 528
  benefits of 543
  event group 552
  groups 563
  host group 551
  hostTopN group 551
  hysteresis mechanism 551
  matrix group 552
  MIB 542, 556, 563
  nlhost (network-layer host) group 554
  nlMatrix (network-layer matrix) group 555
  on Gigabit Ethernet ports 546
  probe 542
  probeConfig group 555
  protocolDir group 553
  protocolDist group 554
  SmartAgent software 523
  statistics 547, 548
  Version 1 544
    groups 547
  Version 2 544
    groups 552
RMON-2 564
  groups 564
  MIB definition 564
root bridge 119
root port 119
route aggregation 268, 299
route flapping, AppleTalk networks 461
route summarization, OSPF 376
route support, OSPF 361
routed traffic
  and flow classifiers 479
router databases, OSPF 375
router IDs, OSPF 359, 365, 387, 396
router interface 271
router updates, OSPF 360
routers
  area border 364, 375, 376, 387, 389
  autonomous system boundary (ASBRs), OSPF 370
  autonomous system boundary, OSPF 387, 389
  backbone, OSPF 364
  backup designated, OSPF 364, 366
  databases, OSPF 375
  default, OSPF 379
  designated, OSPF 364, 367, 388
  IDs, OSPF 365
  internal, OSPF 364
  link state databases, OSPF 383, 387
  OSPF 369
  OSPF, types of 364
  placement of OSPF 369
  seed 455
routes
  AppleTalk 460
  default 379
  external, OSPF 390
    and stub areas 400
routing
  and protocol-based VLANs 186
  AppleTalk 445
  as requirement for RSVP 477
  between VLANs 168, 189
  inter-area, OSPF 368
  intra-area, OSPF 368
  IP multicast 331

overview 258
sample VLAN configuration 191
system 281
to different autonomous systems 368
to stub area, OSPF 368
Routing Information Protocol (RIP) 358
routing policies
adding routes to the routing table 301
advertising routes to other routers 301
defined 300
routing policies, OSPF 360, 403
and static routes 404
export 408, 411, 412
examples 415
import 405, 408
routing policies, OSPF (Open Shortest Path First)
and imported RIP routes 404
routing table, AppleTalk 452
routing table, IP
contents 272
default route 273, 285
dynamic routes 273
metric 272
static routes 273, 285
status 272
routing table, IP (Internet Protocol)
described 272
routing table, IPX
example 433
roving analysis
and RMON 528
and Spanning Tree 527
configuration rules 529
process overview 527
rules 527
RSVP (Resource Reservation Protocol) 475, 479, 516
overview 476
protocol standard 478
routing requirement 477
sample configuration 518
setting parameters 520
terms 516
RTMP (Routing Table Maintenance Protocol) 450, 460, 465, 471
rules
ingress and egress VLAN 162, 195
port numbering 60

S

S port 101
sample configurations
GVRP 185
Ignore STP mode 173
multiple QoS classifiers and control 501
port numbering 62
QoS excess tagging 512
QoS filtering classifiers and controls 499
QoS high priority 504
QoS nonflow classifiers and controls 506, 508
QoS to/from classifiers and controls 497
RSVP 518
SAP (Service Advertising Protocol)
aging mechanism 436
request handling 436
using for dynamic routes 435
SAP policies
IPX 441
SAS (single attach station) ports
port numbering 65
Save 47
saving nvData 55
scripting 246
security 45, 46
limiting IP management access 49
SNMP community strings 538
seed interfaces, AppleTalk 458, 459
seed routers, AppleTalk 455
segmentation, network 262
serial line, and management access 38
servers, bandwidth to 73, 78
service levels, QoS 480, 489
services for event logging 524
session layer protocols, AppleTalk 452
session layer, AppleTalk 450
session protocol, AppleTalk (ASP) 453
shared explicit style, RSVP 517
Shared VLAN Learning (SVL) 169
shiftl opcode 232
shiftr opcode 232
shortest path trees, OSPF 367
Simple Network Time Protocol (SNTP) 45, 46, 47, 56
singlemode fiber 85
Slave port 101

SMT (Station Management) 89
lerAlarm value 107
lerCutoff value 108
smtProxyTraps (SNMP) 540
snapshot 46
SNMP 37
access 43
accessing external applications 34
agent
defined 533, 538
working with SNMP manager 538
community strings
defined 538
values 539
defined 533
displaying configurations 539
external applications 31
manager
defined 533
working with SNMP agent 538
messages
Get 534
Get Responses 534
Get-next 534
Set 534
overview 34
smtProxyTraps 540
trap reporting
configuring destinations 539
displaying configuration 539
flushing addresses 540
SNMP traps
addressThresholdEvent 138
defined 535
message description 534
supported objects 535
SNTP (Simple Network Tiime Protocol) 56
SNTP client 57
socket values filter 248, 251
soft restarts 398
software update 45, 46
source addresses 486
source IP address masks 486
Spanning Tree
algorithm 117
blocking paths 117
CBPDU 119
designated bridge 119
designated port 119
enabling 134
IP multicast 344
port identifier 122
root bridge 119
root port 119
spanning tree
IP multicast 344
speed
Ethernet ports 82
Fast Ethernet ports 82
SPGM (source port group mask) 242
SRF (Status Report Frames)
and FDDI stations 104
and lerAlarm 107
stack 226
standard packet filter 213
standards
IEEE 802.1p 478
related to QoS and RSVP 478
states, AppleTalk interface 458
static neighbors, OSPF 395
static route, IP 273, 285
static route, IPX 435
statistics
AppleTalk 470
baselining 525
DDP (Datagram Delivery Protocol) 470
NBP (Name Binding Protocol) 473
OSPF soft restart 398
RMON 547, 548
RTMP (Routing Table Maintenance Protocol) 471
VLAN 207
ZIP (Zone Information Protocol) 472
statistics, OSPF 360, 416
statistics, OSPF interface 384
status reporting
configuring 104
defined 104
status, routing table 272
STP (Spanning Tree Protocol)
bridge priority, setting 134
enabling on bridge 134
enabling on bridge port 136
forward delay, setting 135
group address, setting 135
hello time, setting 134
ignoring blocking 172
maximum age, setting 134
port priority 136
stub areas, OSPF 372, 373, 377, 400
stub default metrics 359, 400
subnet mask 265
defined 265
example 266
IP interface parameter 283
routing table 272
subnet mask numbering 267

subnetworking
defined 265
Ethernet switching 259
subnet mask 265
summary link state advertisements, OSPF 389
swapping modules 68
switched traffic
and nonflow classifiers 479
switches, bandwidth to 73, 78
system
access methods 38
access overview 31
system console security 49
system ID mismatch 54
system menu 45
system parameters
options and guidelines 48
security 49
software updates 52

T

T_Opr 105
table, Routing Table Maintenance Protocol (RTMP) 460
tag status rules 198
tagging 162
egress rules for transmit ports 199
excess packets 481
for port-based VLANs 178
for protocol-based VLANs 186, 192
IEEE 802.1Q VLAN 159
in allOpen mode 171
priority 481
tags
priority 478, 481, 488, 491
TCP
drop control 480, 494
one-way filtering 480, 494
ports 487
technical support
3Com Knowledgebase Web Services 567
bulletin board service 568
fax service 569
network suppliers 569
product repair 571
Telnet 36
terminal emulation 36
terminal port
access 41
baud rate setting 38
establishing a connection 38
setting up 38
using an emulator 41
terms
VLAN 161
text editor, built-in 217
timer option 481
tmaxLowerBound
defined 105
setting 105
T-notify
defined 104
token 100
topology, FDDI 91 to 93
total reservable bandwidth 518
traffic patterns
RMON2 MIB 564
transit areas, OSPF 373
transmit delay, OSPF 385
transmit ports
VLAN rules for 199
transmit priorities, QoS 480, 489
transparent bridging
and aging addresses 116
IEEE 802.1D compliant 142
transport layer, AppleTalk 450
trap commands (SNMP)
flush 540
trap reporting
addressThresholdEvent 138
configuring destinations 539
defined 535
flushing addresses 540
removing destinations 540
trap-based polling 537
T-Req 105
triggered updates
SAP 438, 441
Trivial File Transfer Protocol (TFTP) 47, 48, 49
Trunk Control Message Protocol (TCMP) 146

trunks
anchor port 145
and default VLAN 177
and port numbering 61
benefits of 144
capacity 148
configuring before establishing IP interfaces 282
configuring before VLANs 168
defining 150
effects of module removals 66
effects of module replacements 68, 70
Ethernet 73
explained 144
Fast Ethernet 149
Gigabit Ethernet 149
implementing 147
managing 143
modifying 152
port numbering 145
removing 153
removing ports 178
Trunk Control Message Protocol (TCMP) 146
trusted IP clients 47, 50, 51
tvxLowerBound
defined 104

U

UDP
ports 487
UDP Helper
administering 311
configuring overlapped IP interfaces 312
display 313
guidelines 313
hop count 312
overlapped IP interfaces 312
threshold 312
UDP Port number 312, 313
unicast client 56
unspecified protocol 176, 178
untagged ports 199
updates
SAP triggered 438, 441
updates, GVRP 182
User Datagram Protocol (UDP) 37

V

variable length subnet mask (VLSM), and OSPF 362
vi editor 217
VID (VLAN ID) 159
GVRP and 182
range 160
router port IP interfaces and 160
Viewing nvData 55
virtual links, OSPF 360, 362, 364, 368, 375, 377, 378, 387, 392, 401, 402
virtual router
Backup router 319
bridge loops 327
DHCP 329
IGMP 328
initialize state 319
Master router 319
primary IP address 319
prioritizing backup routers 321
using 318, 319
with RIP and OSPF 327
virtual terminal protocol
Telnet 36
VLANs
and AppleTalk interfaces 458
configuring before establishing IP interfaces 282
interface index 283
VLANs (virtual LANs) 155
allClosed mode 169
allOpen mode 169, 171
allOpen or allClosed mode 158
calculating number of 163, 167
default VLAN 161, 167, 176
effects of module removals 66
effects of module replacements 68, 70
flooding decisions 200
forwarding decisions 204
GVRP 182
IEEE 802.1Q
tag format 159
Ignore STP mode 170, 172, 173
ingress and egress rules 195
modifying 206
modifying the VLAN mode 170
network-based 192

origin 160, 182
overview 156
port-based 175
protocol-based 186
removing 206
routing between 168, 189
selecting modes 169
statistics 207
supported protocol suites 187
terms 161
trunks and 168
VIDs 160, 182
VLSMs (Variable Length Subnet Masks) 268
VRRP (Virtual Router Redundancy Protocol) 316, 322
advertisement messages 322
and dynamic routing protocols 327
and ICMP Redirect 329
and IGMP 328
concepts 318
DHCP 329
initialize state 319
primary IP address 319
prioritzing backup routers 321
QOS
virtual router 329
STP (Spanning Tree Protocol) 327
using 319
virtual router 318, 322
virtual router backup 319
virtual router master 319

W

Web Management software 51
access 42
applications 31, 45
browser requirements 42
interface window 42
using Internet Explorer 42
using Netscape Navigator 42
wildcard filter style, RSVP 518
wildcards for flow classifier addresses/masks 486
wiring closets 78

X

XNS
in packet filter 240, 248, 250
xor opcode 231

Z

zeroes, in classifier addresses and masks 487
ZIP (Zone Information Protocol) 465, 472
ZIT (Zone Information Table) 453, 461, 465
aging entries 465
Zone Information Protocol (ZIP) 453, 465, 472
zones, AppleTalk 455, 458, 461, 464
changing 467
changing names of 466
example of 455
guidelines for configuring 465
naming 465