Nways Multiprotocol Access Services

SC30-3884-00

Protocol Configuration and Monitoring Reference Volume 1 Version 1 Release 1

Nways Multiprotocol Access Services

SC30-3884-00

Protocol Configuration and Monitoring Reference Volume 1 Version 1 Release 1

Note

Before using this document, read the general information under “Notices” on page xxi.

First Edition (March 1997)

This edition applies to Version 1 Release 1 of the IBM Nways Multiprotocol Access Services and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments is provided at the back of this publication. If the form has been removed, you may address your comments to:

Department CGF
Design & Information Development
IBM Corporation
P.O. Box 12195
RESEARCH TRIANGLE PARK NC 27709
USA

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices xxi
Trademarks xxi

Preface xxiii
About the Software xxiii
Conventions Used in This Manual xxiv
Library Overview xxvi

Part 1. Configuring and Monitoring Bridge Functions

Chapter 1. Bridging Basics 1-1
Bridging Overview 1-1
Bridges versus Routers 1-2
Router Connections 1-2
Bridge Connections 1-2
Types of Bridges 1-3
Simple Bridges 1-3
Complex Bridges 1-4
Local Bridges 1-4
Remote Bridges 1-4
Basic Bridge Operation 1-5
Operation Example 1: Local Bridge Connecting Two LANs 1-5
Operation Example 2: Remote Bridging Over a Serial Link 1-5
MAC Bridge Frame Formats 1-7

Chapter 2. Bridging Methods 2-1
Transparent Bridging 2-1
Routers and Transparent Bridges 2-2
Network Requirements 2-2
Transparent Bridge Operation 2-2
Shaping the Spanning Tree 2-3
Spanning Tree Bridges and Ethernet Packet Format Translation 2-5
IBM RT Feature for SNA Traffic 2-6
UB Encapsulation of XNS Frames 2-6
Transparent Bridging and Frame Relay 2-6
Transparent Bridge Terminology and Concepts 2-6
Source Routing Bridging (SRB) 2-10
Source Routing Bridge Operation 2-11
Source Routing Frames 2-11
The Spanning Tree Explore Option 2-14
Protocol Filtering 2-15
Source Route Bridging and Frame Relay 2-15
Source Routing Bridge Terminology and Concepts 2-16
Source Routing Transparent (SRT) Bridge 2-17
General Description 2-18
Source Routing Transparent Bridge Operation and Architecture 2-18
SRT Bridge and Frame Relay 2-19
Source Routing Transparent Bridge Terminology 2-19
ASRT Bridge Overview 2-20

Adaptive Source Routing Transparent Bridge (ASRT) (SR-TB Conversion) . 2-20
General Description 2-21
Source Routing—Transparent Bridge Operation 2-21
SR-TB and Frame Relay 2-27
Source Routing—Transparent Bridge (SR-TB) Terminology and Concepts 2-27
Transparent-Source Routing Compatibility - Issues and Solutions 2-28
ASRT Configuration Considerations 2-29
ASRT Configuration Matrix 2-30

Chapter 3. Bridging Features 3-1
Bridging Tunnel 3-1
Encapsulation and OSPF 3-2
TCP/IP Host Services (Bridge-Only Management) 3-3
Bridge-MIB Support 3-3
NetBIOS Name Caching 3-3
Duplicate Frame Filtering 3-4
NetBIOS Name and Byte Filters 3-4
Types of NetBIOS Filtering 3-4
Building a Filter 3-6
Simple and Complex Filters 3-6
Multiple Spanning Tree Protocol Options 3-7
Background: Problems with Multiple Spanning Tree Protocols 3-7
STP/8209 3-7
Threading (Router Discovery) 3-8

Chapter 4. Using the Boundary Access Node (BAN) Feature 4-1
About the Boundary Access Node Feature 4-1
Benefits of BAN 4-2
How BAN works 4-2
Bridged Versus DLSw BAN 4-2
Which Method Should You Use? 4-4
Using the BAN Feature 4-4
Using Multiple DLCIs for BAN Traffic 4-7
Scenario 1: Setting up a Fault-Tolerant BAN Connection 4-7
Scenario 2: Increasing Bandwidth to the IBM Environment 4-8
Setting up Multiple DLCIs 4-8
Checking the BAN Configuration 4-8
Enabling Event Logging System (ELS) Messages for BAN 4-9

Chapter 5. Basic Bridging Configurations 5-1
Accessing the ASRT Configuration Environment 5-1
Basic Bridging Configuration Procedures 5-1
Bridging Interfaces 5-1
Enabling the Transparent Bridge 5-2
Enabling the Source Routing Bridge 5-2
Enabling the SR-TB Bridge 5-3

Chapter 6. Overview of Routing Over ATM 6-1
Overview of Routing 6-1
Overview of Bridging 6-1
Bridging Behaviors 6-2
Overview of RFC 1483 Support 6-3
Overview of RFC 1483 Support for Routing 6-3
RFC 1483 Support for IPX Routing 6-3

Chapter 7. Configuring Bridging . . . 7-1
Accessing the ASRT Configuration Environment . . . 7-1
ASRT Configuration Commands . . . 7-1
? (Help) . . . 7-3
Add . . . 7-3
BAN . . . 7-12
Change . . . 7-12
Delete . . . 7-13
Disable . . . 7-15
Enable . . . 7-18
List . . . 7-21
NetBIOS . . . 7-28
Set . . . 7-28
Tunnel . . . 7-34
Exit . . . 7-34
Tunnel Configuration Commands . . . 7-34
Tunneling and Multicast Packets . . . 7-35
? (Help) . . . 7-35
Add . . . 7-36
Delete . . . 7-36
Join . . . 7-36
Leave . . . 7-37
List . . . 7-38
Set . . . 7-38
Exit . . . 7-38
Frame-Relay Commands . . . 7-39
List . . . 7-39

Chapter 8. Monitoring Bridging . . . 8-1
Accessing the ASRT Console Environment . . . 8-1
ASRT Console Commands . . . 8-1
? (Help) . . . 8-2
Add . . . 8-2
BAN . . . 8-3
Cache . . . 8-3
Delete . . . 8-4
Flip . . . 8-4
List . . . 8-5
NetBIOS . . . 8-18
Exit . . . 8-18
Accessing the BAN Console Prompt . . . 8-19
BAN Console Commands . . . 8-19
? (Help) . . . 8-19
List . . . 8-20
Exit . . . 8-20

Chapter 9. Using, Configuring, and Monitoring NetBIOS . . . 9-1
About NetBIOS . . . 9-1
NetBIOS Names . . . 9-1
NetBIOS Name Conflict Resolution . . . 9-2
NetBIOS Session Setup Procedure . . . 9-2
NetBIOS Broadcast Data Flows . . . 9-2
NetBIOS Status Flows . . . 9-2
NetBIOS All-Stations Broadcast Frames . . . 9-3

Reducing NetBIOS Traffic 9-3
Frame Type Filtering 9-4
Duplicate Frame Filtering 9-5
Response Frame Filtering 9-8
NetBIOS Name Lists 9-9
NetBIOS Name Caching and Route Caching 9-12
NetBIOS Host Name and Byte Filtering Configuration Procedures 9-15
Creating a Host Name Filter 9-16
Creating a Byte Filter 9-18
About NetBIOS Configuration and Monitoring Commands 9-20
Accessing the NetBIOS Configuration Environment 9-21
Accessing the NetBIOS Console Environment 9-21
Configuring NetBIOS for DLSw 9-21
NetBIOS Commands 9-24
? (Help) 9-24
Add 9-24
Delete 9-26
Disable 9-27
Enable 9-28
List (Configuration) 9-29
List (Monitoring) 9-31
Set 9-37
Test (Monitoring only) 9-41
Exit 9-42

Chapter 10. Configuring NetBIOS Filtering 10-1
Accessing the ASRT and the DLSW Configuration Environments 10-1
NetBIOS Filtering Configuration Commands 10-1
? (Help) 10-2
Create 10-2
Delete 10-2
Disable 10-3
Enable 10-3
Filter-on 10-3
List 10-5
Update 10-5
Exit 10-10

Chapter 11. Monitoring NetBIOS Filtering 11-1
Accessing the ASRT and the DLSw NetBIOS Filtering Console Environments 11-1
NetBIOS Filtering Monitoring Commands 11-1
? (Help) 11-1
List 11-2
Exit 11-3

Chapter 12. Using, Configuring, and Monitoring LAN Network Manager (LNM) 12-1
About LNM 12-1
LNM Agents and Functions 12-1
LNM Configuration Restrictions 12-3
Configuring LNM 12-4
LNM Commands 12-5
? (Help) 12-6
Disable 12-6

Enable 12-6
List (configuration command) 12-7
List (monitoring command) 12-8
Set 12-9
Exit 12-9

Chapter 13. Configuring TCP/IP Host Services 13-1
Basic Configuration Procedures 13-1
Setting the IP Address 13-1
Adding a Default Gateway 13-1
Enabling TCP/IP Host Services 13-1
Accessing the TCP/IP Host Configuration Environment 13-1
TCP/IP Host Configuration Commands 13-2
? (Help) 13-2
Add 13-2
Delete 13-3
Disable 13-3
Enable 13-3
List 13-4
Set 13-5
Exit 13-5

Chapter 14. Monitoring TCP/IP Host Services 14-1
Accessing the TCP/IP Host Console Environment 14-1
TCP/IP Host Console Commands 14-1
? (Help) 14-2
Dump 14-2
Interface 14-3
Ping 14-3
Traceroute 14-4
Routers 14-5
Exit 14-5

Part 2. Configuring and Monitoring Router Protocols

Chapter 15. Overview of Classical IP Over ATM 15-1
Benefits of Classical IP 15-1
Components of Classical IP 15-2
Timeouts and Refresh 15-2
IP Addresses and CIP components 15-3
ATM Addresses of CIP components 15-3
Virtual Channel Connections 15-3
Key Configuration Parameters for Classical IP 15-4

Chapter 16. Using and Configuring IP 16-1
Basic Configuration Procedures 16-1
Assigning IP Addresses to Network Interfaces 16-1
Enabling Dynamic Routing 16-2
Adding Static Routing Information 16-4
Setting Up ARP Configuration 16-6
Enabling ARP Subnet Routing 16-6
IP Filtering 16-7
Access Control 16-7

Route Filtering . . . 16-10
Configuring the BOOTP/DHCP Forwarding Process . . . 16-10
Enabling/Disabling BOOTP Forwarding . . . 16-11
Configuring a BOOTP/DHCP Server . . . 16-12
Configuring UDP Forwarding . . . 16-12
Enabling/Disabling UDP Forwarding . . . 16-12
Adding a UDP Destination . . . 16-13
IP Multicast Support . . . 16-13
Accessing the IP Configuration Environment . . . 16-16
IP Configuration Commands . . . 16-16
? (Help) . . . 16-16
Add . . . 16-17
Change . . . 16-22
Delete . . . 16-24
Disable . . . 16-26
Enable . . . 16-28
List . . . 16-34
Move . . . 16-36
Set . . . 16-37
Update . . . 16-41
Exit . . . 16-44

Chapter 17. Monitoring IP . . . 17-1
Accessing the IP Console Environment . . . 17-1
IP Console Commands . . . 17-1
? (Help) . . . 17-2
Access Controls . . . 17-2
Cache . . . 17-3
Counters . . . 17-4
Dump Routing Table . . . 17-5
Interface Addresses . . . 17-6
Packet-filter . . . 17-7
Ping . . . 17-7
Route . . . 17-8
Sizes . . . 17-8
Static Routes . . . 17-9
Traceroute . . . 17-9
UDP-Forwarding . . . 17-10
Exit . . . 17-10

Chapter 18. Using and Configuring IPX . . . 18-1
IPX Overview . . . 18-1
IPX Addressing . . . 18-1
Configuring IPX . . . 18-2
Optional Configuration Tasks . . . 18-3
Specifying the Size of IPX RIP Network Table . . . 18-3
Specifying RIP Update Interval . . . 18-3
Specifying the Size of IPX SAP Services Table . . . 18-4
Specifying SAP Update Interval . . . 18-4
IPX Keepalive and Serialization Packet Filtering . . . 18-4
Configuring Multiple Routes . . . 18-5
Configuring Global IPX Filters (IPX Access Controls) . . . 18-5
Global SAP Filters . . . 18-7
IPX Interface Filters - Overview . . . 18-8

IPX Performance Tuning 18-11
Split-Horizon Routing 18-13
Accessing the IPX Configuration Environment 18-15
IPX Configuration Commands 18-15
? (Help) 18-16
Add 18-16
Delete 18-19
Disable 18-20
Enable 18-21
Filter-lists 18-22
Frame 18-23
List 18-24
Move 18-25
Set 18-26
Exit 18-30
Accessing the IPX Interface Filter Configuration Environment 18-30
IPX Interface Filter Configuration Commands 18-30
? (Help) 18-31
Attach 18-31
Create 18-32
Default 18-32
Delete 18-32
Detach 18-33
Disable 18-33
Enable 18-33
List 18-34
Move 18-34
Set-cache 18-35
Update 18-35
Add (Update subcommand) 18-35
Delete (Update subcommand) 18-40
List (Update subcommand) 18-40
Move (Update subcommand) 18-40
Set-action (Update subcommand) 18-41
Exit 18-41

Chapter 19. Monitoring IPX 19-1
Accessing the IPX Console Environment 19-1
IPX Console Commands 19-1
? (Help) 19-2
Access Controls 19-2
Cache 19-3
Config 19-4
Counters 19-5
Delete 19-6
Disable 19-6
Dump 19-6
Enable 19-7
Filters 19-8
Filter-lists 19-8
IPXWAN 19-8
Keepalive 19-10
Ping 19-11
Sizes 19-11

Slist . . . 19-12
Exit . . . 19-13
IPX Interface Filter Monitoring Commands . . . 19-14
Cache . . . 19-14
Clear . . . 19-14
Disable . . . 19-15
Enable . . . 19-15
List . . . 19-15
Exit . . . 19-16

Chapter 20. Using and Configuring ARP . . . 20-1
ARP Overview . . . 20-1
Inverse ARP Overview . . . 20-3
Classical IP and ARP Over ATM Overview (RFC 1577) . . . 20-4
Classical IP (CIP) Logical IP Subnets (LIS) . . . 20-4
Advantages of Classical IP . . . 20-4
Classical IP Components . . . 20-5
Timeouts and Refresh . . . 20-6
IP Addresses and CIP Components . . . 20-6
ATM Addresses of CIP Components . . . 20-7
Virtual Channel Connection (VCC) . . . 20-7
Key Configuration Parameters for Classical IP . . . 20-8
How to Enter Addresses . . . 20-8
IPX and ARP Over ATM Overview (RFC 1483) . . . 20-10
Accessing the ARP Configuration Environment . . . 20-10
ARP and Inverse ARP Configuration Commands . . . 20-11
? (Help) . . . 20-11
Add Entry . . . 20-12
Change Entry . . . 20-12
Delete Entry . . . 20-13
Disable Auto-Refresh . . . 20-13
Enable Auto-Refresh . . . 20-13
List . . . 20-14
Set . . . 20-15
Exit . . . 20-15
ARP Over ATM Configuration Commands . . . 20-16
Differences for IP and IPX . . . 20-16
Effect on ARP Table Entries . . . 20-16
? (Help) . . . 20-16
Add . . . 20-17
Change . . . 20-23
Delete . . . 20-25
List . . . 20-27
Exit . . . 20-29

Chapter 21. Monitoring ARP . . . 21-1
Accessing the ARP Console Environment . . . 21-1
ARP Console Commands . . . 21-2
? (Help) . . . 21-2
Clear . . . 21-2
Dump . . . 21-2
Hardware . . . 21-3
Ping . . . 21-3
Protocol . . . 21-4

Statistics . . . 21-4
Exit . . . 21-5
ARP Over ATM Console Commands . . . 21-6
? (Help) . . . 21-7
Delete . . . 21-7
Display . . . 21-7
Dump . . . 21-8
Hardware . . . 21-9
Ping . . . 21-9
Protocol . . . 21-9
Statistics . . . 21-10

Chapter 22. Using, Configuring, and Monitoring XTP . . . 22-1
The X.25 Transport Protocol . . . 22-1
Configuration Information . . . 22-1
DTE Address Wildcards . . . 22-3
XTP Backup Peer Function . . . 22-4
Searching for a Remote DTE . . . 22-4
Connection Request Timer . . . 22-4
Configuring XTP . . . 22-5
Configuration Procedures . . . 22-5
Setting the Data Link . . . 22-6
Configuring the IP Interface . . . 22-6
Configuring X.25 . . . 22-6
Setting the National Personality . . . 22-8
Defining the IP Address . . . 22-8
Setting the Internal IP Address . . . 22-8
Configuring XTP . . . 22-9
Sample Configuration of Remote Routers . . . 22-10
XTP Configuration Commands . . . 22-13
? (Help) . . . 22-13
Add . . . 22-13
Change . . . 22-15
Delete . . . 22-15
Enable . . . 22-16
Disable . . . 22-16
Set . . . 22-16
List . . . 22-16
Exit . . . 22-18
XTP Monitoring Commands . . . 22-19
? (Help) . . . 22-19
Add . . . 22-19
Delete . . . 22-20
List . . . 22-20
Exit . . . 22-23

Chapter 23. Using and Configuring OSPF . . . 23-1
The OSPF Routing Protocol . . . 23-1
OSPF Routing Summary . . . 23-1
Multicast OSPF . . . 23-3
Configuring OSPF over ATM . . . 23-4
Configuring OSPF . . . 23-4
Enabling the OSPF Protocol . . . 23-5
Defining Backbone and Attached OSPF Areas . . . 23-5

Setting OSPF Interfaces 23-9
Multicast Forwarding 23-11
Setting Non-Broadcast Network Interface Parameters 23-11
Configuring Wide Area Subnetworks 23-11
Enabling AS Boundary Routing 23-13
Other Configuration Tasks 23-13
Converting from RIP to OSPF 23-15
Dynamically Changing Interface Costs 23-16
Accessing the OSPF Configuration Environment 23-17
OSPF Configuration Commands 23-17
? (Help) 23-17
Add 23-18
Delete 23-19
Disable 23-21
Enable 23-21
Join 23-23
Leave 23-23
List 23-23
Set 23-27
Exit 23-32

Chapter 24. Monitoring OSPF 24-1
Accessing the OSPF Console Environment 24-1
OSPF Console Commands 24-1
? (Help) 24-2
Advertisement Expansion 24-2
Area Summary 24-5
AS-external advertisements 24-6
Database Summary 24-7
Dump Routing Tables 24-8
Interface Summary 24-9
Join 24-11
Leave 24-11
Mcache 24-11
Mgroups 24-12
Mstats 24-13
Neighbor Summary 24-14
Ping 24-16
Traceroute 24-16
Routers 24-16
Size 24-17
Statistics 24-17
Weight 24-19
Exit 24-19

Chapter 25. Configuring SNMP 25-1
Accessing the SNMP Configuration Environment 25-1
SNMP Configuration Commands 25-1
? (Help) 25-3
Add 25-3
Delete 25-5
Disable 25-7
Enable 25-8
List 25-9

Set 25-11
Exit 25-11

Chapter 26. Monitoring SNMP 26-1
Accessing the SNMP Console Environment 26-1
SNMP Console Commands 26-1
? (Help) 26-2
Add 26-2
Delete 26-3
Disable 26-3
Enable 26-3
List 26-3
Revert 26-4
Save 26-5
Set 26-5
Statistics 26-5
Exit 26-5

Chapter 27. Using and Configuring DLSw 27-1
About DLSw 27-1
How DLSw Works 27-1
Benefits of DLSw 27-3
Using DLSw Features 27-3
TCP Connections, Neighbor Discovery, and Multicast Exploration 27-4
LLC Device Support 27-6
SDLC Device Support 27-6
QLLC Device Support 27-10
APPN Interface Support 27-15
Using the Neighbor Priority Feature 27-16
Balancing SNA and NetBIOS Traffic 27-17
Setting up DLSw 27-18
DLSw Configuration Requirements 27-18
Setting Global Buffers 27-18
Configuring OSPF for DLSw 27-18
Configuring Adaptive Source Route Bridging (ASRT) for DLSw 27-19
Configuring the Internet Protocol (IP) for DLSw 27-20
Configuring SDLC Interfaces 27-21
Configuring X.25 Interfaces 27-22
Configuring DLSw 27-22
Sample DLSw Configuration 27-23
Sample Diagram 27-23
Sample Configuration Commands 27-24
Accessing the DLSw Configuration Environment 27-35
Pre-Configuration Requirements 27-36
DLSw Configuration Commands 27-37
? (Help) 27-38
Add 27-38
BAN 27-42
Close-Sap 27-43
Delete 27-43
Disable 27-44
Enable 27-45
Join-Group 27-46
Leave-Group 27-47

List 27-48
NetBIOS 27-51
Open-Sap 27-51
Set 27-52
Exit 27-56

Chapter 28. Monitoring DLSw 28-1
Accessing the DLSw Console Environment 28-1
DLSw Console Commands 28-1
? (Help) 28-2
Add 28-2
BAN 28-3
Close-SAP 28-3
Delete 28-3
Disable 28-4
Enable 28-4
Join-Group 28-5
Leave-Group 28-5
List 28-5
NetBIOS 28-18
Open-Sap 28-18
Set 28-19

Appendix A. MIB Support A-1
How to Access MIBs on the Internet A-1
MIB Support Cross-Reference Table A-2
Standard MIBs A-3
AppleTalk MIB A-3
ATM MIB A-4
BGP4 MIB A-4
Bridge MIB A-5
DLSw MIB A-5
Ethernet MIB A-6
Frame Relay MIB A-6
Interfaces MIB A-6
LAN Emulation Client MIB A-7
LLC MIB A-7
MIB II A-8
OSPF MIB A-8
PPP MIB A-9
RS-232 MIB A-9
SDLC MIB A-10
Source Route Bridge MIB A-10
SNA NAU MIB A-10
Token-Ring MIB A-11
X.25 LAPB A-11
X.25 Packet A-11
X.25 MIOX A-12
Enterprise MIBs A-13
APPN MIB A-13
BRS MIB A-13
DLUR MIB A-14
HPR MIB A-14
IBM 2216 MIBs A-14

IBM Common MIB A-14
Novell IPX MIB A-15
Novell RIP/SAP MIB A-16
Proteon Enterprise MIB A-16
Remote Ping A-16
SNMP Generic Traps A-17
Enterprise-specific Traps A-17
Settable Values A-18

Appendix B. Interoperating with the IBM 6611 Router B-1
Bridge Configuration Considerations B-1
DLSw-Related Considerations B-1
IP-Related Configuration Considerations B-2
TCP-Related Considerations B-2
Miscellaneous Interoperability Considerations B-3

Appendix C. Interoperating with the IBM 6611 Bridge C-1
Bridging Configuration Considerations C-1
Configuration Examples C-1

List of Abbreviations X-1

Glossary X-5

Index X-29

Figures

1-1. Simple and Complex Bridging Configuration 1-2
1-2. Two-Port Bridge Connecting Two LANs 1-5
1-3. Bridging Over a Point-to-Point Link 1-6
1-4. Data Encapsulation Over a Point-to-Point Link 1-6
1-5. Examples of MAC Frame Formats 1-7
2-1. Networked LANs Before Spanning Tree 2-4
2-2. Spanning Tree Created With Default Values 2-5
2-3. User-Adjusted Spanning Tree 2-5
2-4. Example of Source Routing Bridge Connectivity 2-10
2-5. 802.5 Source Address Format 2-12
2-6. 802.5 Routing Information Field 2-12
2-7. Example of Parallel Bridges 2-14
2-8. Using Spanning Tree Explore for Load Balancing 2-15
2-9. Bridge Instances within a Bridge 2-16
2-10. SRT Bridge Operation 2-18
2-11. SR-TB Bridge Connecting Two Domains 2-22
2-12. SR-TB Bridging Examples 2-25
3-1. Example of the Bridge Tunnel Feature 3-2
4-1. Direct Connection of End Stations to an SNA Node Using BAN . . . 4-1
4-2. BAN Type 1: The Router as an LLC2 Bridge 4-3
4-3. BAN Type 2: Local DLSw Conversion 4-4
4-4. BAN Configuration with Multiple DLCIs to Different SNA Nodes . . . 4-7
6-1. IP Routing in the router 6-1
6-2. IPX Routing in the router 6-1
6-3. Bridging Over the Emulated LAN Interface 6-3
9-1. Setting Up a NetBIOS Session Over DLSw 9-6
12-1. LNM Station and Agents 12-1
16-1. Access Control - Searching a Packet for Forwarding 16-7
18-1. Keepalive Filtering 18-5
18-2. Sample IPX Network 18-13
18-3. Partially Meshed Frame-Relay Network 18-14
20-1. ARP Address Resolution Broadcast 20-2
22-1. Configuration Before and After XTP 22-2
22-2. Sample XTP Configuration 22-5
23-1. OSPF Areas 23-7
23-2. OSPF Routing Hierarchy 23-15
27-1. Traditional Approach to Bridging Across WAN Links 27-2
27-2. Data Link Switching over the WAN 27-3
27-3. Example DLSw SDLC Configurations 27-7
27-4. Example DLSw QLLC Configurations 27-11
27-5. APPN-to-DLSw Software Interface 27-15
27-6. Sample Diagram for DLSw Configuration 27-24

Tables

2-1. Spanning Tree Default Values 2-4
2-2. Route/Bridge Decision Table 2-15
2-3. SR-TB Bridge Decision Table 2-23
7-1. ASRT Configuration Command Summary 7-2
7-2. Tunnel Configuration Commands 7-35
8-1. ASRT Console Commands Summary 8-2
8-2. BAN Console Commands Summary 8-19
9-1. NetBIOS Filters 9-3
9-2. NetBIOS List Cache Configuration Commands 9-15
9-3. NetBIOS List Cache Monitoring Commands 9-15
9-4. NetBIOS Configuration and Monitoring Commands 9-24
10-1. NetBIOS Filtering Configuration Commands 10-1
11-1. NetBIOS Filtering Monitoring Commands Summary 11-1
12-1. LNM Command Summary 12-5
13-1. TCP/IP Host Configuration Commands Summary 13-2
14-1. TCP/IP Host Console Commands Summary 14-1
16-1. IP Configuration Commands Summary 16-16
17-1. IP Console Command Summary 17-1
18-1. IPX Configuration Commands Summary 18-15
18-2. IPX Filter Configuration Command Summary 18-30
19-1. IPX Console Command Summary 19-1
19-2. IPX Interface Filter Command Summary 19-14
20-1. ARP Configuration Commands Summary 20-11
20-2. ARP Over ATM Configuration Command Summary 20-16
21-1. ARP Console Command Summary 21-2
21-2. ARP Over ATM Console Command Summary 21-6
22-1. XTP Configuration Commands Summary 22-13
22-2. XTP Monitoring Commands Summary 22-19
23-1. OSPF Configuration Command Summary 23-17
24-1. OSPF Console Command Summary 24-1
25-1. SNMP Configuration Commands Summary 25-1
25-2. SNMP Configuration Commands Options Summary 25-2
26-1. SNMP Console Command Summary 26-2
27-1. DLSw Optional Protocols 27-18
27-2. DLSw Configuration Commands Summary 27-37
28-1. DLSw Console Commands Summary 28-1
A-1. Supported MIBs A-2
A-2. AppleTalk Phase 2 Supported MIB Groups A-3
A-3. Exceptions for AppleTalk MIB A-4
A-4. ATM Supported MIB Groups A-4
A-5. Exceptions for ATM MIB A-4
A-6. BGP4 Supported MIB Groups A-4
A-7. Bridge Supported MIB Groups A-5
A-8. DLSw Supported MIB Groups A-5
A-9. Exceptions for DLSw MIB A-6
A-10. Ethernet Supported MIB Groups A-6
A-11. Frame Relay Supported MIB Groups A-6
A-12. Interface MIB Supported Groups A-6
A-13. Exceptions for Interface MIB A-7
A-14. LAN Emulation Client Supported MIB Groups A-7

A-15. Exceptions for LAN Emulation Client MIB A-7
A-16. LLC Supported MIB Groups A-8
A-17. MIB II Supported Groups A-8
A-18. OSPF Supported MIB Groups A-8
A-19. Exceptions for OSPF MIB A-9
A-20. PPP Supported MIB Groups A-9
A-21. RS-232 Supported MIB Groups A-9
A-22. SDLC Supported MIB Groups A-10
A-23. Source Route Bridge Supported MIB Groups A-10
A-24. SNA NAU Supported MIB Groups A-10
A-25. Token-Ring Supported MIB Groups A-11
A-26. X.25 LAPB Supported MIB Groups A-11
A-27. X.25 Packet Supported MIB Groups A-11
A-28. X.25 MIOX Supported MIB Groups A-12
A-29. APPN Supported MIB Groups A-13
A-30. IBM 2216 BRS MIB Groups A-13
A-31. DLUR Supported MIB Groups A-14
A-32. HPR Supported MIB Groups A-14
A-33. IBM 2216 MIB Supported Groups A-14
A-34. IBM Common Supported MIB Groups A-15
A-35. Novell IPX Supported MIB Groups A-15
A-36. Exceptions for Novell IPX MIB A-15
A-37. Novell RIP/SAP Supported MIB Groups A-16
A-38. Proteon MIB Groups A-16
A-39. Remote Ping Supported MIB Groups A-16
A-40. SNMP Generic Traps A-17
A-41. Enterprise-Specific Traps A-17
A-42. Settable Values A-18

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Peer-to-Peer Networking
AIX
AIXwindows
APPN
BookManager
IBM
Micro Channel
NetView
Nways
PS/2
RISC System/6000
System/370
VTAM

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Preface

This manual contains the information you will need to configure bridging and routing functions on an Nways device. The manual describes all of the features and functions that are in the software. A specific Nways device might not support all of the features and functions described. If a feature or function is device-specific, a notice in the relevant chapter or section indicates that restriction.

This manual supports the IBM 2216 and refers to this product as either “the router” or “the device.” The examples in the manual represent the configuration of an IBM 2216 but the actual output you see may vary. Use the examples as a guideline to what you might see while configuring your device.

Who Should Read This Manual: This manual is intended for persons who install and operate computer networks. Although experience with computer networking hardware and software is helpful, you do not need programming experience to use the protocol software.

To Get Additional Information: Changes may be made to the documentation after the books are printed. If additional information is available or if changes are required after the books have been printed, the changes will be in a file (named README) on diskette 1 of the configuration program diskettes. You can view the file with an ASCII text editor.

About the Software

IBM Nways Multiprotocol Access Services is the software that supports the IBM 2216 (licensed program number 5765-B87). This software has these components:

- The base code, which consists of:
  - The code that provides the routing, bridging, data link switching, and SNMP agent functions for the device.
  - The router user interface, which allows you to configure, monitor, and use the IBM Nways Multiprotocol Access Services base code installed on the device. The router user interface is accessed locally through an ASCII terminal or emulator attached to the service port, or remotely through a Telnet session or modem-attached device.

  The base code is installed at the factory on the 2216.
- The Configuration Program for IBM Nways Multiprotocol Access Services (*Configuration Program*), a graphical user interface that allows you to configure the device from a stand-alone workstation. The Configuration Program includes error checking and online help information.

  The Configuration Program is not preloaded at the factory; it is shipped separately from the device as part of the software order.

  You can also FTP the Configuration Program for IBM Nways Multiprotocol Access Services. See *Configuration Program User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1,* GC30-3830, for the server address and directories.

Conventions Used in This Manual

The following conventions are used in this manual to show command syntax and program responses:

1. The abbreviated form of a command is shown in the following example:

   reload

   In this example, you can enter either the whole command (reload) or its abbreviation (rel).

2. Three periods following an option mean that you enter additional data (for example, a variable) after the option. For example:

   time host ...

   In this example, you enter the IP address of the host in place of the periods, as explained in the description of the command.

3. In information displayed in response to a command, defaults for an option are enclosed in brackets immediately following the option. For example:

   ```
   Media (UTP/STP) [UTP]
   ```

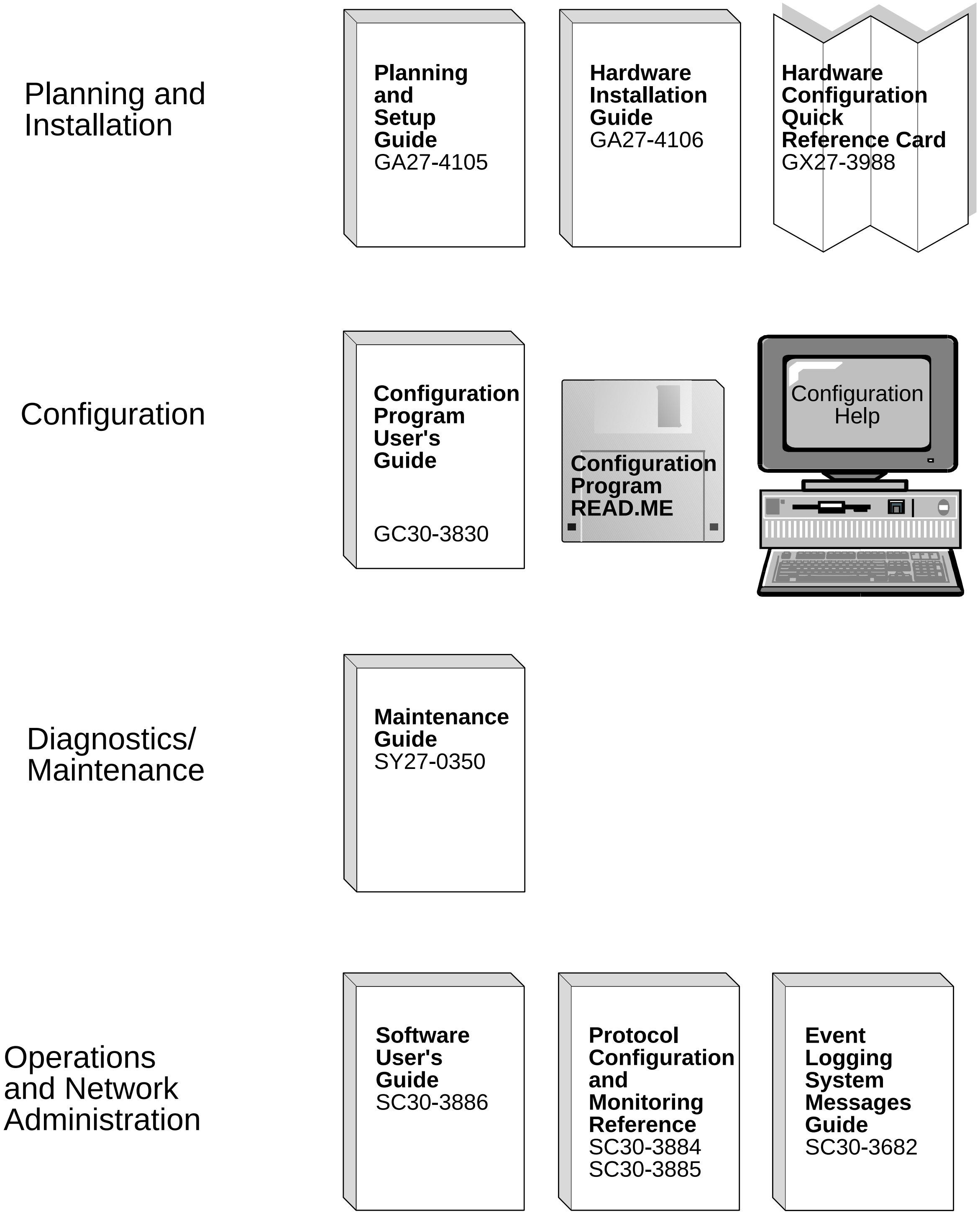
   In this example, the media defaults to UTP unless you specify STP.

4. Keyboard key combinations are indicated in text in the following ways:

   Ctrl-P

   Ctrl P

Common Tasks and the IBM 2216 Library


Planning and Installation
Planning and Setup Guide GA27-4105
Hardware Installation Guide GA27-4106
Hardware Configuration Quick Reference Card GX27-3988
Configuration
Configuration Program User's Guide GC30-3830
Configuration Program READ.ME
Configuration Help
Diagnostics/ Maintenance
Maintenance Guide SY27-0350
Operations and Network Administration
Software User's Guide SC30-3886
Protocol Configuration and Monitoring Reference SC30-3884 SC30-3885
Event Logging System Messages Guide SC30-3682

Library Overview

The following list shows the books in the IBM 2216 library, arranged according to tasks.

Information updates and corrections: To keep you informed of engineering changes, clarifications, and fixes that were implemented after the books were printed, refer to the IBM 2216 home pages at:

```
      http://www.networking.ibm.com/216/216prod.html
   and
      http://www.networking.ibm.com/216/216lib.html
```

Planning

GA27-4105 *IBM 2216 Multiaccess Connector Planning and Setup Guide*

This book is shipped with the IBM 2216. It explains how to prepare for installation and perform an initial configuration.

Installation

GA27-4106 *IBM 2216 Nways Multiaccess Connector Hardware Installation Guide*

This booklet is shipped with the IBM 2216. It explains how to install the IBM 2216 and verify its installation.

GX27-3988 *2216 Nways Multiaccess Connector Hardware Configuration Quick Reference*

This reference card is used for entering and saving hardware configuration information used to determine the correct state of an IBM 2216.

Diagnostics and Maintenance

SY27-0350 *2216 Nways Multiaccess Connector Maintenance Guide.*

This book is shipped with the IBM 2216 It provides instructions for diagnosing problems with and repairing the IBM 2216.

Operations and Network Management

The following list shows the books that support the Nways Multiprotocol Access Services program.

SC30-3886 *Nways Multiprotocol Access Services Software User’s Guide*

This book explains how to:

- Configure, monitor, and use the Nways Multiprotocol Access Services software.
- Use the Nways Multiprotocol Access Services command-line router user interface to configure and monitor the network interfaces and link-layer protocols shipped with the IBM 2216.

SC30-3884 *Nways Multiprotocol Access Services Protocol Configuration and Monitoring Reference, Volume 1*

SC30-3885 *Nways Multiprotocol Access Services Protocol Configuration and Monitoring Reference, Volume 2*

These books describe how to access and use the Nways Multiprotocol Access Services command-line user interface to configure and monitor the routing protocol software shipped with the product.

They include information about each of the protocols that the devices support.

SC30-3682 *Nways Event Logging System Messages Guide*

This book contains a listing of the error codes that can occur, along with descriptions and recommended actions to correct the errors.

Configuration

GC30-3830 *Configuration Program User's Guide*

This book discusses how to use the Nways Multiprotocol Access Services Configuration Program.

Safety

SD21-0030 *Caution: Safety Information—Read This First*

This book, shipped with the IBM 2216, provides translations of caution and danger notices applicable to the installation and maintenance of a IBM 2216.

Marketing

URL: **http://www.networking.ibm.com/216/216prod.html**

This IBM Web page provides product information through the World Wide Web.

Part 1. Configuring and Monitoring Bridge Functions

Chapter 1. Bridging Basics 1-1
Bridging Overview 1-1
Bridges versus Routers 1-2
Types of Bridges 1-3
Basic Bridge Operation 1-5
MAC Bridge Frame Formats 1-7

Chapter 2. Bridging Methods 2-1
Transparent Bridging 2-1
Source Routing Bridging (SRB) 2-10
Source Routing Transparent (SRT) Bridge 2-17
ASRT Bridge Overview 2-20
Adaptive Source Routing Transparent Bridge (ASRT) (SR-TB Conversion) . 2-20

Chapter 3. Bridging Features 3-1
Bridging Tunnel 3-1
TCP/IP Host Services (Bridge-Only Management) 3-3
Bridge-MIB Support 3-3
NetBIOS Name Caching 3-3
Duplicate Frame Filtering 3-4
NetBIOS Name and Byte Filters 3-4
Multiple Spanning Tree Protocol Options 3-7
Threading (Router Discovery) 3-8

Chapter 4. Using the Boundary Access Node (BAN) Feature 4-1
About the Boundary Access Node Feature 4-1
Using the BAN Feature 4-4
Using Multiple DLCIs for BAN Traffic 4-7
Checking the BAN Configuration 4-8
Enabling Event Logging System (ELS) Messages for BAN 4-9

Chapter 5. Basic Bridging Configurations 5-1
Accessing the ASRT Configuration Environment 5-1
Basic Bridging Configuration Procedures 5-1

Chapter 6. Overview of Routing Over ATM 6-1
Overview of Routing 6-1
Overview of Bridging 6-1
Overview of RFC 1483 Support 6-3

Chapter 7. Configuring Bridging 7-1
Accessing the ASRT Configuration Environment 7-1
ASRT Configuration Commands 7-1
Tunnel Configuration Commands 7-34
Frame-Relay Commands 7-39

Chapter 8. Monitoring Bridging 8-1
Accessing the ASRT Console Environment 8-1
ASRT Console Commands 8-1
Accessing the BAN Console Prompt 8-19

BAN Console Commands 8-19

Chapter 9. Using, Configuring, and Monitoring NetBIOS 9-1
About NetBIOS 9-1
Reducing NetBIOS Traffic 9-3
NetBIOS Host Name and Byte Filtering Configuration Procedures 9-15
About NetBIOS Configuration and Monitoring Commands 9-20
NetBIOS Commands 9-24

Chapter 10. Configuring NetBIOS Filtering 10-1
Accessing the ASRT and the DLSW Configuration Environments 10-1
NetBIOS Filtering Configuration Commands 10-1

Chapter 11. Monitoring NetBIOS Filtering 11-1
Accessing the ASRT and the DLSw NetBIOS Filtering Console Environments 11-1
NetBIOS Filtering Monitoring Commands 11-1

Chapter 12. Using, Configuring, and Monitoring LAN Network Manager (LNM) 12-1
About LNM 12-1
Configuring LNM 12-4
LNM Commands 12-5

Chapter 13. Configuring TCP/IP Host Services 13-1
Basic Configuration Procedures 13-1
Accessing the TCP/IP Host Configuration Environment 13-1
TCP/IP Host Configuration Commands 13-2

Chapter 14. Monitoring TCP/IP Host Services 14-1
Accessing the TCP/IP Host Console Environment 14-1
TCP/IP Host Console Commands 14-1

Chapter 1. Bridging Basics

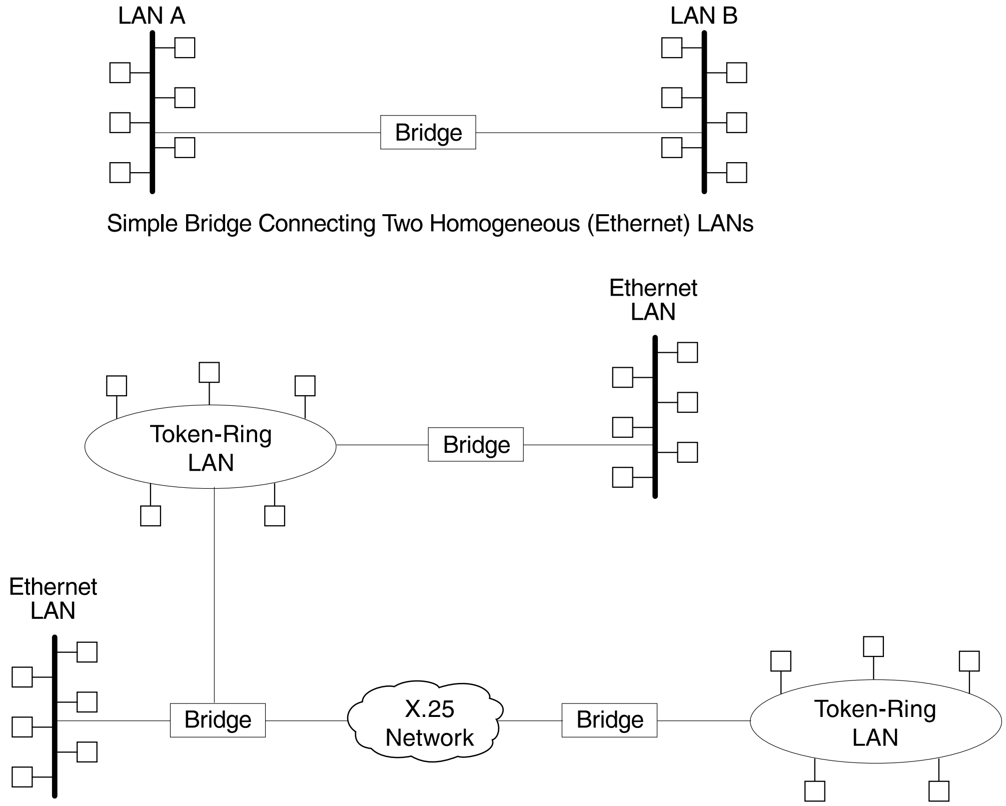
This chapter discusses basic information about bridges and bridging operation. The chapter includes the following sections:

- “Bridging Overview”
- “Bridges versus Routers” on page 1-2
- “Types of Bridges” on page 1-3
- “Basic Bridge Operation” on page 1-5
- “MAC Bridge Frame Formats” on page 1-7

Bridging Overview

A bridge is a device that links two or more local area networks. The bridge accepts data frames from each connected network and then decides whether to forward each frame based on the MAC header contained in the frame. Bridges originally linked two or more homogeneous networks. The term *homogeneous* means that the connected networks use the same bridging method and media types. Examples of these would be networks supporting the source routing bridging method ***only*** or transparent bridging algorithm ***only*** (these methods will be explained later).

Current bridges also allow communication between non-homogeneous networks. *Non-homogeneous* refers to networks that can mix different bridging methods and can also offer more configuration options. Figure 1-1 on page 1-2 illustrates examples of simple and complex bridging configurations.



Simple Bridge Connecting Two Homogeneous (Ethernet) LANs

Complex Bridging Configuration Connecting Different LAN Technologies

Figure 1-1. Simple and Complex Bridging Configuration

Bridges versus Routers

Internetworking devices such as bridges and routers have similar functions in that they connect network segments. However, each device uses a different method to establish and maintain the LAN-to-LAN connections. Routers connect LANs at Layer 3 (Network Layer) of the OSI model while bridges connect LANs at Layer 2 (Link Layer).

Router Connections

Connecting at Layer 3 with a router allows connectivity and path selection between end stations located in distant geographical areas. Using routing protocols, you can select the best path for connecting distant and diverse LANs. Because of the variety of network and subnetwork configuration options available to you in large networks, connecting LANs through the Network Layer is usually the preferred method. Network layer protocols have also proven to be very efficient in moving information in large and diverse network configurations.

Bridge Connections

Connecting at Layer Two with a bridge provides connectivity across a physical link. This connection is essentially “transparent” to the host connected on the network.

Note: Source routing bridges are not considered completely “transparent.” See Chapter 2, “Bridging Methods” in this guide for more information on source routing and transparent bridges.

The Link Layer maintains physical addressing schemes (vs. logical at Layer 3), line discipline, topology reporting, error notification, flow control, and ordered delivery of data frames. Isolation from upper layer protocols is one of the advantages of bridging. Since bridges function at the Link Layer, they are not concerned with looking at the protocol information that occurs at the upper layers. This provides for lower processing overhead and fast communication of network layer protocol traffic. Because bridges are not concerned with Layer 3 information, they can also forward different types of protocol traffic (for example, IP or IPX) between two or more networks (as routers do).

Bridges can also filter frames based on Layer 2 fields. This means that the bridge can be configured to accept and forward only frames of a certain type or ones that originate from a particular network. This ability to configure filters is very useful for maintaining effective traffic flow.

Bridges are advantageous when dividing large networks into manageable segments. The advantages of bridging in large networks can be summed up as follows:

- Bridging lets you isolate specific network areas giving them less exposure to major network problems.
- Filtering lets you regulate the amount of traffic that is forwarded to specific segments.
- Bridges allow communication among a larger number of internetworking devices than would be supported on any single LAN connected to a bridge.
- Bridging eliminates node limitation (the total number of nodes on a segment). Local network traffic is not passed on to all of the other connected networks.
- Bridges extend the connected “length” of a LAN by allowing the connection of distant LAN segments. Bridges connect two LAN segments at layer 2 so that larger networks can be formed. This overcomes the congestion problems with too many stations on an Ethernet and the 256 station limit in the token-ring architecture.

Types of Bridges

The following sections describe specific types of bridges and how they can be classified by their hardware and software capabilities.

Simple Bridges

Simple bridges consist of two or more linked network interfaces connecting local area networks (Figure 1-1 on page 1-2). Bridges interconnect separate local area networks (LANs) by relaying data frames between the separate MAC (medium access control) entities of the bridged LANs.

The main functions of a simple bridge may be summarized as follows:

- The bridge reads all data frames transmitted on LAN A and receives those addressed to LAN B. Simple bridges make no changes to the content or format of the data frames that they receive. They also do not encapsulate frames with any additional headers.

Most simple bridges contain routing addressing and routing intelligence. At a minimum, the bridge must know which addresses are on each connected network so that it can know which frames to pass on.

- The bridge retransmits the data frames addressed to LAN B on to LAN B using the MAC protocol for that LAN. Bridges should have enough buffer space to meet peak data traffic demands because data frames may arrive faster than the bridge can transmit them.
- The bridge does the same for LAN B-to-LAN A data frame traffic.

Complex bridges are capable of carrying out even more sophisticated functions.

Complex Bridges

Complex bridges carry out more sophisticated functions than simple bridges. These functions may include the bridge maintaining status information on the other bridges. This information includes the communication path cost as well as the number of hops required to reach each connected network. Periodic exchanges of information between bridges update all bridge information. These types of exchanges allow for dynamic routing between bridges.

Complex bridges can also modify frames and recognize and transmit packets from different LAN technologies (for example, Token-Ring and Ethernet). In this case the bridge is sometimes referred to as a *translational* bridge.

The adaptive source routing transparent (ASRT) bridge is the 2216's implementation of bridge technology. The ASRT Bridge is a collection of software components capable of several of the bridging options just described and more. All of these functions are explained in greater detail later in this chapter.

Local Bridges

Local bridges provide connections among several LAN segments in the same geographical area. An example of this would be a bridge used to connect the various LANs located in your company's main headquarters.

Remote Bridges

Remote bridges connect multiple LAN segments in different geographical areas. An example of this would be bridges used to the connect LANs located in your company's main headquarters to LANs in other branch offices around the country. Because of the geographical differences, this configuration moves from a local area network configuration to a wide area network (WAN) configuration.

Remote bridges can differ from local bridges in several ways. One major difference is found in the speed at which data is transmitted. WAN connections may be slower than LAN connections. This difference in speed can make quite a difference when running time-sensitive applications. Another difference is found in the physical way in which remote and local bridges are connected to LANs. In local bridges, the connections are made through local cabling media (for example, Ethernet, Thinet). Remote bridge connections are made over the serial lines.

Basic Bridge Operation

According to the IEEE 802 LAN standard, all station addresses are specified at the MAC level. At the LLC (Logical Link Control) level, only SAP (Service Access Point) addresses are designated. Accordingly, the MAC level is the level at which the bridge functions. The following examples explain how bridging functions proceed at this level.

Operation Example 1: Local Bridge Connecting Two LANs

Figure 1-2 shows a two-port bridge model connecting end stations on two separate LANs. In this example, the local bridge connects LANs with identical LLC and MAC layers (that is, two token-ring LANs). Conceptually, you can think of the bridge as a data link relay that forwards frames between the media access control (MAC) sublayers and physical channels of the attached LANs, thus providing data link connectivity between them.

To summarize the bridging process, the bridge captures MAC frames whose destination addresses are not on the local LAN (that is, the LAN connected to the interface receiving the transmitted frame). It then forwards them to the appropriate destination LAN. Throughout this process, there is a dialogue between the peer LLC entities in the two end-stations. Architecturally, the bridge need not contain an LLC layer because the function of the LLC layer is to merely relay MAC frames that come from upper levels of the OSI model.

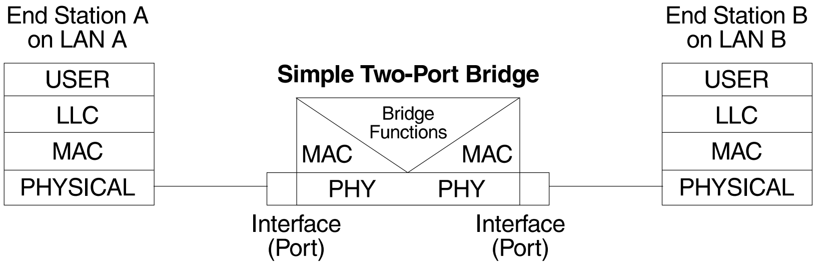


Figure 1-2. Two-Port Bridge Connecting Two LANs

Operation Example 2: Remote Bridging Over a Serial Link

Figure 1-3 on page 1-6 shows a pair of bridges connected over a serial link. These remote bridges connect LANs with identical LLC and MAC layers (that is, two token-ring LANs).

To summarize, the bridge captures a MAC frame whose destination address is not on the local LAN and then sends it to the appropriate destination LAN via the bridge on that LAN. Throughout this process, there is a dialogue between the peer LLC entities in the two end stations. Architecturally, the bridge need not contain an LLC layer because the function of the LLC layer is to merely relay MAC frames that come from upper levels of the OSI model.

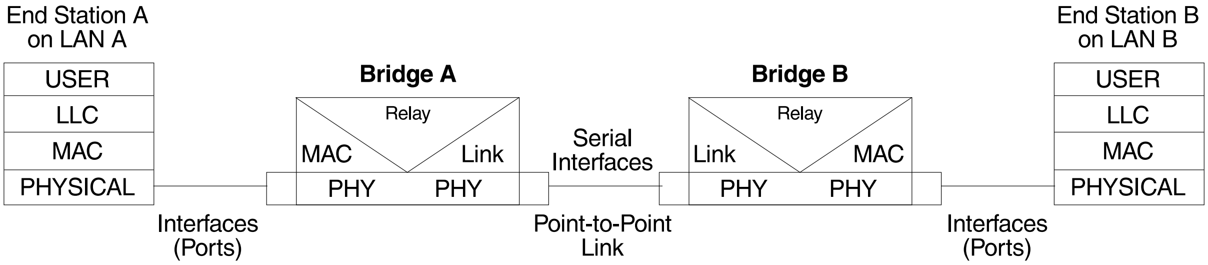


Figure 1-3. Bridging Over a Point-to-Point Link

Data is encapsulated as the bridges communicate data over the serial link. Figure 1-4 illustrates the encapsulation process.

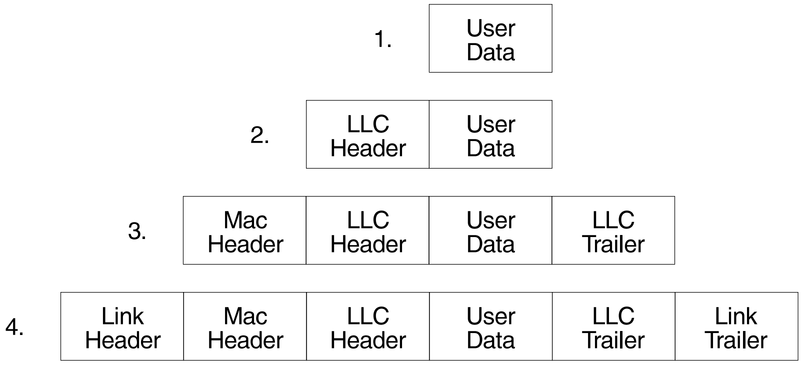


Figure 1-4. Data Encapsulation Over a Point-to-Point Link

Encapsulation proceeds as follows:

1. End station A provides data to its LLC.
2. LLC appends a header and passes the resulting data unit to the MAC level.
3. MAC then appends a header (3) and trailer to form a MAC frame. Bridge A captures the frame.
4. Bridge A does not strip off the MAC fields because its function is to relay the intact MAC frame to the destination LAN. In the point-to-point configuration, however, the bridge appends a link layer (for example, HDLC) header and trailer and transmits the MAC frame across the link.

When the data frame reaches Bridge B (the target bridge), the link fields are stripped off and Bridge B transmits the *original, unchanged* MAC frame to its destination, end station B.

MAC Bridge Frame Formats

As mentioned, bridges interconnect LANs by relaying data frames, specifically MAC frames, between the separate MAC entities of the bridged LANs. MAC frames provide the necessary "Where?" information for frame forwarding in the form of source and destination addresses. This information is essential for the successful transmission and reception of data.

IEEE 802 supports three types of MAC frames: CSMA/CD (802.3), token bus (802.4), and token-ring (802.5). Figure 1-5 shows the CSMA/CD and Token-Ring MAC frame formats supported by the bridges. The specific frames are detailed in the following section.

Note: A separate frame format is used at the LLC level. This frame is then embedded in the appropriate MAC frame.

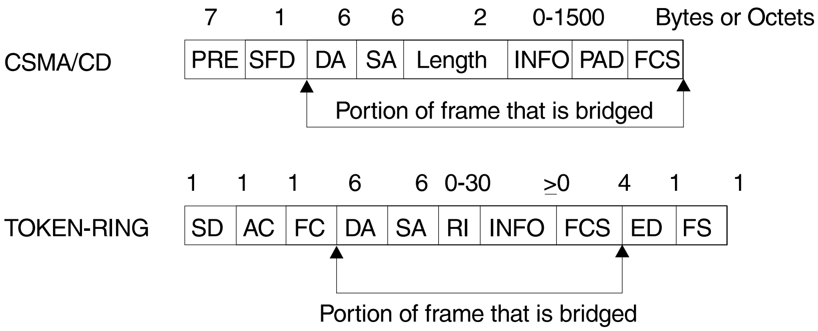


Figure 1-5. Examples of MAC Frame Formats

CSMA/CD (Ethernet) MAC Frames

The following information describes each of the fields found in CSMA/CD (Ethernet) MAC frames:

- *Preamble (PRE)*. A 7-byte pattern used by the receiving end-station to establish bit synchronization and then locate the first bit of the frame.
- *Start Frame Delimiter (SFD)*. Indicates the start of the frame.

The portion of the frame that is actually bridged consists of the following fields:

- *Destination Address (DA)*. Specifies the end-station for which the frame is intended. This address may be a unique physical address (one destination), a multicast address (a group of end-stations as a destination), or a global address (all stations as the destination). The format is 48-bit (6 octets) and must be the same for all stations on that particular LAN.
- *Source Address (SA)*. Specifies the end-station that transmitted the frame. The format must be the same as the destination address format.
- *Length*. Specifies the number of LLC bytes that follow.
- *Info (INFO)*. Embedded fields created at the LLC level that contain service access point information, control information, and user data.
- *Pad*. Sequence of bytes that ensures that the frame is long enough for proper collision detection (CD) operation.

- *Frame Check Sequence (FCS)*. A 32-bit cyclic redundancy check value. This value is based on all fields, starting with the destination address.

Token-Ring MAC Frames

The following information describes each of the fields in token-ring MAC frames:

- *Starting Delimiter (SD)*. Unique 8-bit pattern that indicates the start of the frame.
- *Access Control (AC)*. Field with the format PPPTMRRR where PPP and RRR are 3-bit priority and reservation variables, M is the monitor bit, and T indicates that this is either a token or a data frame. If it is a token frame, the only other field is the ending delimiter (ED).
- *Frame Control (FC)*. Indicates if this is an LLC data frame. If not, bits in this field control operation of the token-ring MAC protocol.

The portion of the frame that is actually bridged consists of the following fields:

- *Destination Address (DA)*. Same as CSMA/CD and token bus.
- *Source Address (SA)*. Identifies the specific station that originates the frame. The length of the field may be either a 2- or 6-octet address. Both address lengths carry a routing information indicator (RII) bit that indicates if a routing information field (RIF) is present in the frame after the source address, as follows:

RII=1	Routing information field is present.
RII=0	Routing information field is not present.

  This field is explained in more detail in “Source Routing Bridging (SRB)” on page 2-10.
- *Routing Information Field (RIF)*. The RIF is required for the source routing protocol. It consists of a 2-octet routing control field and a series of 2-octet route designator fields. This field is explained in more detail in “Source Routing Bridging (SRB)” on page 2-10.
- *Info (INFO)*. Embedded fields created at the LLC level that contain service access point information, control information, and user data.
- *Frame Check Sequence (FCS)*. A 32-bit cyclic redundancy check value. This value is based on all fields, starting with the destination address.

Finally, the *End Delimiter (ED)* contains the error detection (E) bit, and the intermediate frame (I) bit. The I bit indicates that this is not the final frame of a multiple frame transmission. The *Frame Status (FS)* contains the address recognized (A) and frame copied (C) bits.

Chapter 2. Bridging Methods

This chapter describes the methods of bridging supported by the adaptive source routing transparent (ASRT) bridge. Each section gives an overview of a specific technology and is followed by a description of the data frames supported by that technology. The chapter includes the following sections:

- “Transparent Bridging”
- “Source Routing Bridging (SRB)” on page 2-10
- “Source Routing Transparent (SRT) Bridge” on page 2-17
- “ASRT Bridge Overview” on page 2-20
- “Adaptive Source Routing Transparent Bridge (ASRT) (SR-TB Conversion)” on page 2-20

Transparent Bridging

The transparent bridge is also commonly known as a spanning tree bridge (STB). The term *transparent* refers to the fact that the bridge silently forwards non-local traffic to attached LANs in a way that is *transparent* or unseen to the user. End station applications do not know about the presence of the bridge. The bridge learns about the presence of end stations by promiscuously listening to traffic passing by. From this listening process it builds a database of end station addresses attached to its LANs.

For each frame it receives, the bridge checks the frame’s destination address against the ones in its database. If the frame’s destination is an end station on the same LAN, the frame is not forwarded. If the destination is on another LAN, the frame is forwarded. If the destination address is not present in the database, the frame is forwarded to all the LANs that are connected to the bridge except the LAN from which it originated.

All transparent bridges use the spanning tree protocol and algorithm. The spanning tree algorithm produces and maintains a loop-free topology in a bridged network that might contain loops in its physical design. In a mesh topology where more than one bridge is connected between two LANs, *looping* occurs. In such cases, data packets bounce back and forth between two LANs on parallel bridges. This creates a redundancy in data traffic and produces the phenomenon known as looping.

When looping occurs, you must configure the local and/or remote LAN to remove the physical loop. With spanning tree, a self-configuring algorithm allows a bridge to be added anywhere in the LAN without creating loops. Upon adding the new bridge, the spanning tree protocol automatically reconfigures all bridges on the LAN into a single loop-free *spanning tree*.

A spanning tree never has more than one active data route between two end stations, thus eliminating data loops. For each bridge, the algorithm determines which bridge ports can forward data and which ones must be blocked to form a loop-free topology. Among its features, spanning tree provides the following:

- *Loop detection*. Detects and eliminates physical data link loops in extended LAN configurations.

- *Automatic backup of data paths*. Deliberately configured from redundant paths. The bridges connecting to the redundant paths enter backup mode automatically. When a primary bridge fails, a backup bridge becomes active.
- *User configurability*. Lets you tailor your network topology. Sometimes the default settings do not produce the desired network topology. You can adjust the bridge priority, port priority, and path cost parameters to shape the spanning tree to your network topology.
- *Seamless interoperability*. Allows LAN interoperability without configuration limitations caused by diverse communications environments.
- *Bridging of non-routing protocols*. Provides cost-effective bridging of non-routing protocols.

Routers and Transparent Bridges

During the operation of a router equipped with the spanning tree option, bridge and router software run concurrently. In this mode, the router is a bridge and a router.

During this operation, the following actions occur:

- Packets are routed if a specific protocol forwarder is globally enabled
- Packets are filtered if you configure specific protocol filters
- Packets that are not routed or filtered are candidates for bridging, depending on the destination MAC (Media Access Control) address.

Network Requirements

Transparent Bridge implements a spanning tree bridge that conforms to the IEEE 802.1D standard. All transparent bridges (such as Ethernet and Token-Ring) on the network must be 802.1D spanning tree bridges. This spanning tree protocol is not compatible with bridges implementing the proprietary Digital Equipment Corporation spanning tree protocol used in some older bridges.

Transparent Bridge Operation

In a mesh topology where more than one bridge is connected between two LANs, a looping phenomenon can occur where two LANs bounce packets back and forth over parallel bridges. A loop is a condition where multiple data paths exist between two LANs. The spanning tree protocol operating automatically eliminates loops by blocking redundant paths.

During startup, all participating bridges in the network exchange Hello bridge protocol data units (BPDUs) which provide configuration information about each bridge. BPDUs include information such as the bridge ID, root ID, and root path cost. This information helps the bridges to unanimously determine which bridge is the root bridge and which bridges are the designated bridges for LANs to which they are connected.

Of all the information exchanged in the HELLO messages, the following parameters are the most important for computing the spanning tree:

- *Root Bridge ID* The root bridge ID is the bridge ID of the bridge. The root bridge is the designated bridge for all the LANs to which it is connected.
- *Root Path Cost*. The sum total of the designated path costs to the root via this bridge's root port. This information is transmitted by both the root bridge and

the designated bridges to update all bridges on path information if the topology changes.

- *Bridge ID* A unique ID used by the spanning tree algorithm to determine the spanning tree. Each bridge in the network is assigned a unique bridge identifier.
- *Port ID* The ID of the port from which the current HELLO BPDU message was transmitted.

With this information available, the spanning tree begins to determine its shape and direction and then creates a logical path configuration. This process can be summarized as follows:

1. A root bridge for the network is selected by comparing the bridge IDs of each bridge in the network. The bridge with the lowest ID (that is, highest value) wins.
2. The spanning tree algorithm then selects a designated bridge for each LAN. If more than one bridge is connected to the same LAN, the bridge with the smallest path cost to the root is selected as the designated bridge. In the case of duplicate path costs, the bridge with the lowest bridge ID is selected as the designated bridge.
3. The non-designated bridges on the LANs put each port that has not been selected as a root port into a BLOCKED state. In the BLOCKED state, a bridge still listens to Hello BPDUs so that it can act on any changes that are made in the network (for example, designated bridge fails) and change its state from BLOCKED to FORWARDING (that is, it will be forwarding data).

Through this process, the spanning tree algorithm reduces a bridged LAN network of arbitrary topology into a single spanning tree. With the spanning tree, there is never more than one active data path between any two end stations, thus eliminating data loops. For each bridge on the network, the spanning tree determines which bridge ports to block from forming loops.

This new configuration is bounded by a time factor. If a designated bridge fails or is physically removed, other bridges on the LAN detect the situation when they do not receive Hello BPDUs within the time period set by the bridge maximum age time. This event triggers a new configuration process where another bridge is selected as the designated bridge. A new configuration is also created if the root bridge fails.

Shaping the Spanning Tree

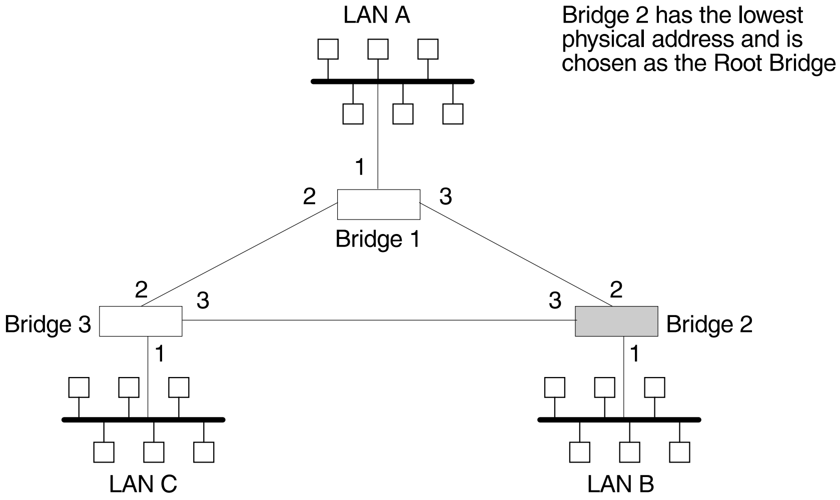
When the spanning tree uses its default settings the spanning tree algorithm generally provides acceptable results. The algorithm, however, may sometimes produce a spanning tree with poor network performance. In this case you can adjust the bridge priority, port priority, and path cost to shape the spanning tree to meet your network performance expectations. The following examples explain how this is done.

Figure 2-1 on page 2-4 shows three LANs networked using three bridges. Each bridge is using default bridge priority settings for its spanning tree configuration. In this case, the bridge with the lowest physical address is chosen as the root bridge because the bridge priority of each bridge is the same. In this example, this is Bridge 2.

The newly configured spanning tree stays intact due to the repeated transmissions of Hello BPDUs from the root bridge at a preset interval (bridge hello time). Through this process, designated bridges are updated with all configuration information. The designated bridges then regenerate the information from the Hello BPDUs and distribute it to the LANs for which they are designated bridges.

Table 2-1. Spanning Tree Default Values

Bridge 1	**Bridge 2**	**Bridge 3**
Bridge Priority: 32768 Address: 00:00:90:00:00:10 **Port 1** Priority: 128 Path Cost: 100 **Port 2** Priority: 128 Path Cost: 17857 **Port 3** Priority: 128 Path Cost: 17857	Bridge Priority: 32768 Address: 00:00:90:00:00:01 **Port 1** Priority: 128 Path Cost: 100 **Port 2** Priority: 128 Path Cost: 17857 **Port 3** Priority: 128 Path Cost: 17857	Bridge Priority: 32768 Address: 00:00:90:00:00:05 **Port 1** Priority: 128 Path Cost: 100 **Port 2** Priority: 128 Path Cost: 17857 **Port 3** Priority: 128 Path Cost: 17857



Bridge 2 has the lowest physical address and is chosen as the Root Bridge

Figure 2-1. Networked LANs Before Spanning Tree

The spanning tree algorithm designates the port connecting Bridge 1 to Bridge 3 (port 2) as a backup port and blocks it from forwarding frames that would cause a loop condition. The spanning tree created by the algorithm using the default values in Table 2-1 is shown in Figure 2-2 on page 2-5 as the heavy lines connecting Bridge 1 to Bridge 2, and then Bridge 2 to Bridge 3. The root bridge is Bridge 2.

This spanning tree results in poor network performance because the workstations on LAN C can get to the file server on LAN A only indirectly through Bridge 2 rather than using the direct connection between Bridge 1 and Bridge 3.

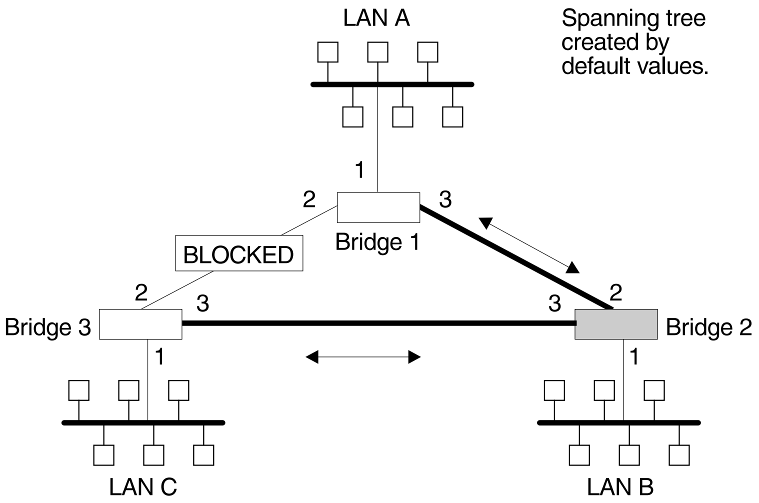


Figure 2-2. Spanning Tree Created With Default Values

Normally, this network uses the port between Bridge 2 and Bridge 3 infrequently. Therefore you can improve network performance by making Bridge 1 the root bridge of the spanning tree. You can do this by configuring Bridge 1 with the highest priority of 1000. The spanning tree that results from this modification is shown in Figure 2-3 as the heavy lines connecting Bridge 1 to Bridge 3 and Bridge 1 to Bridge 2. The root bridge is now Bridge 1. The connection between Bridge 2 and Bridge 3 is now blocked and serves as a backup data path.

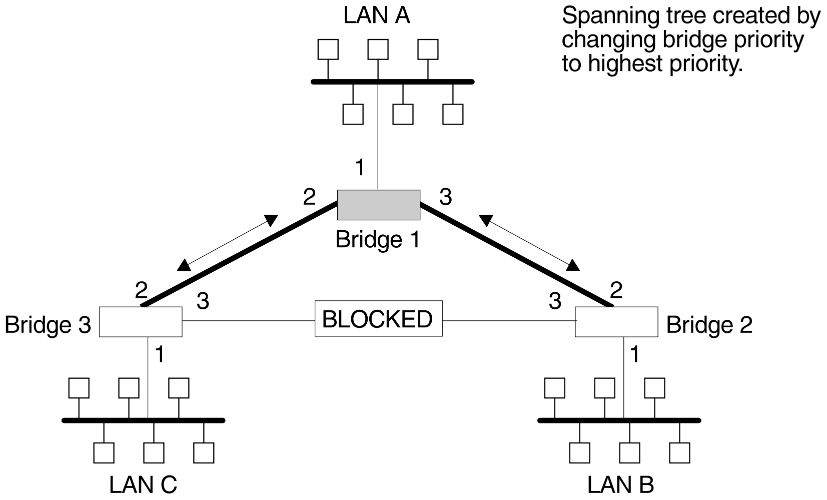

LAN A
Spanning tree created by changing bridge priority to highest priority.
1
2
3
Bridge 1
2
3
BLOCKED
3
2
Bridge 3
Bridge 2
1
1
LAN C
LAN B

Figure 2-3. User-Adjusted Spanning Tree

Spanning Tree Bridges and Ethernet Packet Format Translation

The 2216 Spanning Tree Bridge protocol provides packet forwarding for the bridging routers in accordance to IEEE Standard 802.1D-1990 Media Access Control (MAC) bridges. The protocol also provides appropriate header translation for Ethernet packets.

An Ethernet/IEEE 802.3 network can simultaneously support the Ethernet data link layer and the IEEE 802.2 data link layer, based on the value of the length/type field in the MAC header. The bridge must translate to and from Ethernet format to provide transparency across mixed LAN types. The algorithm used is based on emerging IEEE standards.

The basic approach consists of translating Ethernet packets to IEEE 802.2 Unnumbered Information (UI) packets using the IEEE 802 SNAP SAP. The SNAP Protocol Identifier has the organization-unique identifier (OUI) of 00-00-00, with the last 2 bytes being the Ethernet *type* value.

IBM RT Feature for SNA Traffic

Some IBM personal computers (IBM RT PC running AIX or any PC running OS/2 EE) encapsulate SNA within Ethernet Type 2 packets instead of using IEEE 802.3 Ethernet encapsulation. This requires a special Ethertype header that contains the length of the MAC user data followed by the IEEE 802.2 (LLC) header.

The processing of these frames can be enabled/disabled on a per-port basis. In the enabled mode, the bridge learns the source station’s behavior. When frames are targeted for such stations, the bridge generates the correct frame format. If there is no information about the station’s behavior, (as with multicast or unknown stations), the bridge produces duplicate frames, one in IEEE 802.3 and IEEE 802.2 format, and the other with the IBM-RT header.

UB Encapsulation of XNS Frames

XNS Ethernet frames use Ethertype 0x0600. When translated to token-ring format, these frames get SNAP as specified in IEEE 802.1H. Because some Token-Ring end stations use the Ungermann-Bass OUI in the SNAP for such frames, there is a configuration switch to activate this encapsulation. The switch to activate this encapsulation is set with the **frame token_ring_SNAP** command.

Transparent Bridging and Frame Relay

The Frame Relay interface forwards transparent frames from Ethernet and Token-Ring networks, provided bridging is enabled on the permanent virtual circuit (PVC). IP tunneling does not have to be used.

In a Frame Relay network, Hello BPDUs are generated and transmitted for each PVC configured for transparent bridging. The spanning tree protocol allows Frame Relay PVCs that have not been designated as part of the active data path to be BLOCKED, thereby eliminating loops.

Transparent Bridge Terminology and Concepts

This section reviews the terms and concepts commonly used in transparent bridging.

Aging Time

The aging time parameter determines the length of time (age) before a dynamic entry is removed from the filtering database when the port with the entry is in the forwarding state. If dynamic entries are not referenced by the aging time, they are deleted.

Bridge

A bridge is a protocol-independent device that connects local area networks (LAN). These devices operate at the data link layer, storing and forwarding data packets between LANs.

Bridge Address

The bridge address is the least significant 6-octet part of the bridge identifier used by the spanning tree algorithm to identify a bridge on the network. The bridge address is set to the MAC (media access control) address of the lowest-numbered port by default. You can override the default address by using the **set bridge** configuration command.

Bridge Hello Time

The bridge hello time specifies how often a bridge sends out Hello BPDUs (containing bridge configuration information) when it becomes the root bridge in the spanning tree. This value is useful only for the root bridge because it controls the hello time for all bridges in the spanning tree. Use the **set protocol bridge** command to set the bridge hello time.

Bridge Forward Delay

The bridge forward delay specifies how much time a bridge port spends in the listening state as well as the learning state. The forward delay is the amount of time the bridge port listens in order to adjust the spanning tree topology. It is also the amount of time the bridge spends learning the source address of every packet that it receives while the spanning tree is configuring. This value is useful only for the root bridge because it controls the forward delay for all bridges in the spanning tree.

The root bridge conveys this value to all bridges. This time is set with the **set protocol bridge** command. The procedure for setting this parameter is discussed in the next chapter.

Bridge Identifier

The spanning tree algorithm uses the bridge identifier as a unique ID to determine the spanning tree. Each bridge in the network must have a unique bridge identifier.

The bridge identifier consists of two parts: a least-significant 6-octet bridge address and a most-significant 2-octet bridge priority. By default, the bridge address is set to the MAC (media access control) address of the lowest-numbered port. You can override the default address with the **set bridge** configuration command.

Bridge Maximum Age

The bridge maximum age specifies the amount of time that spanning tree protocol information is considered valid before the protocol discards the information and a topology changes. All the bridges in the spanning tree use this age to time out the received configuration information in their databases. This allows a uniform timeout for every bridge in the spanning tree. Use the **set protocol bridge** command to set the bridge maximum age.

Bridge Priority

The bridge priority is the most significant 2-octet part of the bridge identifier set by the **set protocol bridge** command. This value indicates the chances of each bridge becoming the root bridge of the network. In setting the bridge priority, the spanning tree algorithm chooses the bridge with the highest priority value to be the root bridge of the spanning tree. A bridge with the lowest numerical value has the highest priority value.

Designated Bridge

The designated bridge is the bridge that claims to be the closest to the root bridge on a specific LAN. This closeness is measured according to the accumulated path cost to the root bridge.

Designated Port

The designated port is the port ID of the designated bridge attached to the LAN.

Filtering and Permanent Databases

The bridge's filtering and permanent databases contain information about station addresses that belong to specific port numbers of ports connected to the LAN.

The filtering database is initialized with entries from the permanent database. These entries are permanent and survive power on/off or system resets. You can add or delete these entries through the spanning tree configuration commands. Entries in the permanent database are stored as static random access memory (SRAM) records, and the number of entries is limited by the size of SRAM.

Note: You can also add entries (static) by using the console commands but these ***do not*** survive power on/off and system resets.

The filtering database also accumulates entries learned by the bridge (dynamic entries) which have an aging time associated with them. When entries are not referenced over a certain time period (age time), they are deleted. Static entries are ageless, so dynamic entries cannot overwrite them.

Entries in the filtering and permanent databases contain the following information:

- *Address*. The 6-byte MAC address of the entry
- *Port Map*. Specifies all port numbers associated with that entry
- *Type of Entry*. Specifies one of the following types:
  - Reserved Entries. Reserved by the IEEE 802.1d committee.
  - Registered Entries. Consist of unicast addresses belonging to communications hardware attached to the box or multicast addresses enabled by protocol forwarders.
  - Permanent Entries. Entered by the user in the configuration process. They survive power on/off and system resets.
  - Static Entries. Entered by the user in the console process. They do not survive power on/off and system resets and are ageless.
  - Dynamic Entries. Dynamically learned by the bridge. They do not survive power on/off and system resets and have an associated age.
  - Free. Locations in database that are free to be filled by address entries.

- *Address Age (dynamic entries only)*. Resolution of time period at which address entries are ticked down before being discarded. The user can set this value.

Make changes to the permanent database through the spanning tree configuration commands and make changes to the filtering database through the GWCON console process.

Parallel Bridges

Two or more bridges connecting the same LANs are considered parallel bridges.

Path Cost

Each port interface has an associated path cost which is the relative value of using this port to reach the root bridge in a bridged network. The spanning tree algorithm uses the path cost to compute a path that minimizes the cost from the root bridge to all other bridges in the network topology. The sum total of all the designated costs and the path cost of the root port is called the root path cost.

Port

A port represents the bridge’s connection to each attached LAN or WAN. A bridge must have at least two ports to function as a bridge.

Port ID

The port ID is a 2-octet port identifier. The most-significant octet represents the port priority and the least-significant octet represents the port number. Both port number and port priority are user-assignable. The port ID must be unique within the bridge.

Port Number

The port number is a user-assigned 1-octet part of the port ID whose value represents the attachment to the physical medium. A port number of zero is not allowed.

Port Priority

The port priority is the second 1-octet part of the port ID. This value represents the priority of the port that the spanning tree algorithm uses in making comparisons for port selection and blocking decisions.

Resolution

Resolution is the time factor by which dynamic entries are ticked down as they age within the database. The range is 1 to 60 seconds.

Root Bridge

The root bridge is the bridge selected as the *root* of the spanning tree because it possesses the highest priority bridge ID. This bridge is responsible for keeping the spanning tree intact by regularly emitting Hello BPDUs (containing bridge configuration information). The root bridge is the designated bridge for all the LANs to which it is connected.

Root Port

The root port is the port ID of a bridge's port that offers the lowest cost path to the root bridge.

Spanning Tree

The spanning tree is a topology of bridges such that there is one and only one data route between any two end stations.

Transparent Bridging

This type of bridging involves a mechanism that is *transparent* to end stations applications. Transparent bridging interconnects local area network segments by bridges designated to forward data frames through a spanning tree algorithm.

Source Routing Bridging (SRB)

Source routing is a method of forwarding frames through a bridged network in which the source station identifies the route that the frame will follow. In a distributed routing scheme, routing tables at each bridge determine the path that data takes through the network. By contrast, in a source routing scheme, the source station defines the entire route in the transmitted frame.

The source routing bridge (SRB) provides local bridging over 4 and 16 Mbps token-rings, as shown in Figure 2-4. It can also connect remote LANs through a telecommunications link operating at speeds up to E1.

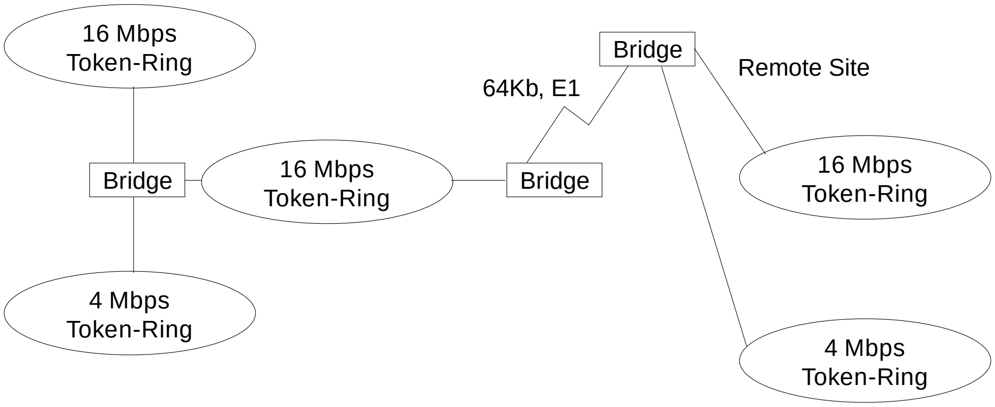


Figure 2-4. Example of Source Routing Bridge Connectivity

Among its features, the source routing bridge provides:

- *Bridge compatibility*. You can use the bridge to connect IBM PC LANs running systems such as OS/2, PC LAN Manager, and NetBIOS. The bridge can also carry IBM SNA traffic between PC LANs and mainframes.
- *Performance and speed*. Because bridging occurs at the data link layer instead of the network layer, packet conversion and address table maintenance are not necessary. This requires less overhead and permits higher speed routing decisions.
- *Bridge Tunneling*. By encapsulating source routing packets, the bridge/router dynamically routes these packets through internetworks to the desired destination end station without degradation or network size restrictions.

Source routing end stations see this path as a single hop, regardless of the network complexity. This helps overcome the usual seven-hop distance limit encountered in source routing configurations. This feature also lets you connect source routing end stations across non-source routing media (for example, Ethernet networks).

Source Routing Bridge Operation

As mentioned, the source station defines the entire route in the transmitted frame in a source routing configuration. The source routing bridge is dynamic. Both end stations and bridges participate in the route discovery and forwarding process. The following steps describe this process:

1. A source station sends out a frame and finds that the frame's destination is not on its own (local) segment or ring.

2. The source station builds a *route discovery* broadcast frame and transmits it onto the local segment.

3. All bridges on the local segment capture the route discovery frame and send it over their connected networks.

   As the route discovery frame continues its search for the destination end station, each bridge that forwards it adds its own bridge number and segment number to the routing information field (RIF) in the frame. As the frame continues to pass through the bridged network, the RIF compiles a list of bridge and segment number pairs describing the path to the destination.

   When the broadcast frame finally reaches its destination, it contains the exact sequence of addresses from source to destination.

4. When the destination end station receives the frame, it generates a response frame including the route path for communication. Frames that wander to other parts of the bridged network (accumulating irrelevant routing information in the meantime) never reach the destination end station and no station ever receives them.

5. The originating station receives the learned route path It can then transmit information across this established path.

Source Routing Frames

As mentioned, bridges interconnect LANs by relaying data frames, specifically MAC frames, between the separate MAC entities of the bridged LANs. MAC frames provide the necessary “Where?” information in the form of source and destination addresses. This information is essential for the successful transmission and reception of data.

In source routing, the data frame forwarding decision is based on routing information within the frame. Before the frame is forwarded, end stations have obtained the route to the destination station by the *route discovery* process. The source station that originates the frame designates the route that the frame will travel by imbedding a description of the route in the routing-information field (RIF) of the transmitted frame. A closer look at the various types of source routing bridge frames will help to further explain how the bridge obtains and transmits this routing information.

Because source routing MAC frames contain routing information necessary for data communication over multi-ring environments, they differ slightly in format from the typical token-ring MAC frames. The presence of a “1” in the RII within the source address field indicates that a RIF containing routing information follows the source address. Figure 2-5 on page 2-12 provides a closer look at the format of the source address field of a source routing frame.

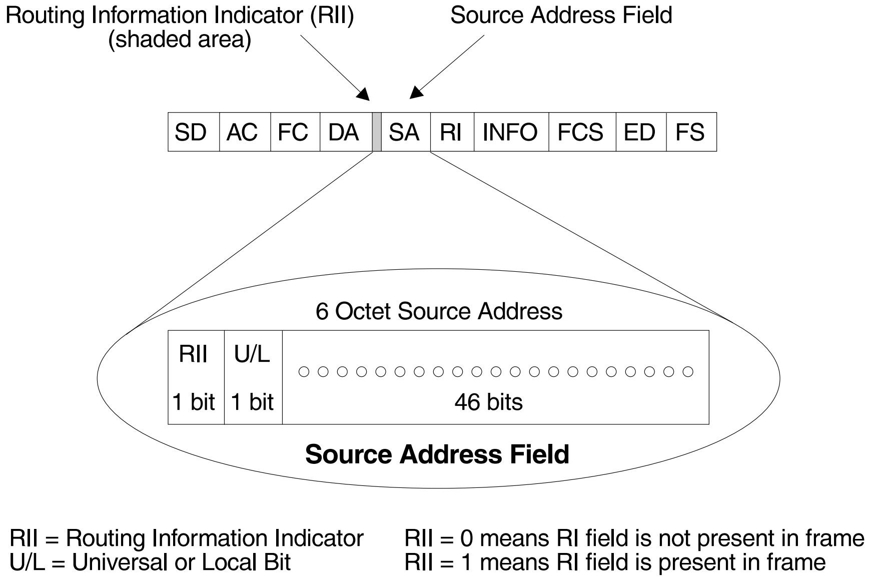

Routing Information Indicator (RII)
(shaded area)
Source Address Field
SD
AC
FC
DA
SA
RI
INFO
FCS
ED
FS
6 Octet Source Address
RII
U/L
1 bit
1 bit
46 bits
Source Address Field
RII = Routing Information Indicator
U/L = Universal or Local Bit
RII = 0 means RI field is not present in frame
RII = 1 means RI field is present in frame

Figure 2-5. 802.5 Source Address Format

When the RII in the source address field is set to 1, a RIF is present after the source address. The RIF is required because it provides route information during source routing. It consists of a 2-octet routing control (RC) field and a series of 2-octet route designator (RD) fields. Figure 2-6 provides a closer look at the format of the Routing Information Field.

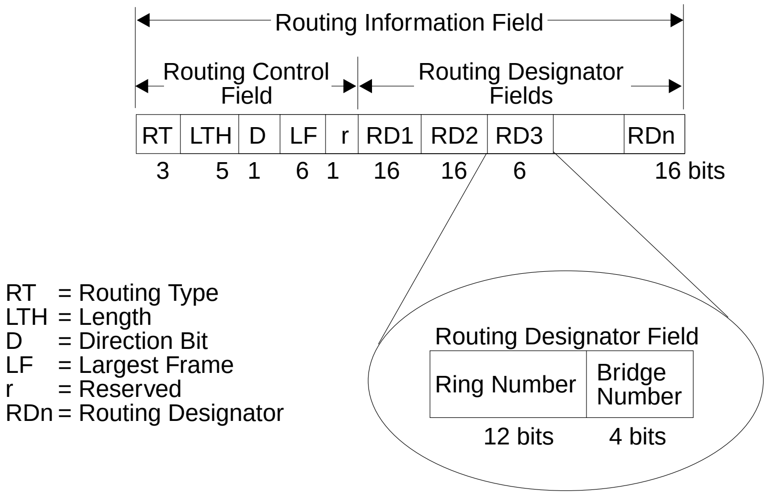


Figure 2-6. 802.5 Routing Information Field

The following information describes each specific field found in the RIF:

- *Routing Type (RT).* Indicates by bit settings if the frame is to be forwarded through the network along a specific route or along a route (or routes) that reaches all interconnected LANs. Depending on the bit settings in this field the source routing frame can be identified as one of the following types:

- All-paths explorer frame (explorer frame)
- Spanning-tree explorer frame (explorer frame)
- Specifically-routed frame (routing frame)
- Spanning-tree routed frame (routing frame)

All-paths explorer frames exist if the RT bits are set to 100. These frames are generated and routed along every non-repeating route in the network (from source to destination). This process results in as many frames arriving at the destination end station as there are different routes from the source end station. This routing type is the response to receiving a route discovery frame sent along the spanning tree to the present originating station using all the routes available. The forwarding bridges add routing designators to the frame.

A *spanning tree explorer frame* exists if the RT bits are set to 110. Only spanning tree bridges relay the frame one network to another. This means that the frame appears only once on every ring in the network and therefore only once at the destination end station. A station initiating the route discovery process uses this frame type. The bridge adds routing designator fields to the frame. It can also be used for frames sent to stations using a group address, which is discussed more fully in the next section.

Specifically routed frames exist if the first RT bit is set to 0. When this is the case, the Route Designator (RD) fields containing specific routing information guide the frame through the network to the destination address. Once the frame reaches its destination and discovers a route path, the destination station returns a specifically routed frame (SRF) to the source station. The source station then transmits its data in a specifically routed frame.

- *Length bits (LTH)*. Indicates the length (in octets) of the RI field.
- *Direction bit (D)*. Indicates the direction the frame takes to traverse the connected networks. If this bit is set to 0, the frame travels the connected networks in the order in which they are specified in the routing information field (for example, RD1 to RD2 to.... to RDn). If the direction bit is set to 1, the frame travels the networks in the reverse order.
- *Largest Frame Bits (LF)*. Indicates the largest frame size of the INFO field that can be transmitted between two communicating end stations on a specific route. The LF bits are meaningful only for STE and ARE frames. In specifically routed frames (SRFs), the bridge ignores the LF bits and can not alter them. A station originating an explorer frame sets the LF bits to the maximum frame size it can handle. Forwarding bridges set the LF bits to the largest value that does not exceed the minimum of:
  - The indicated value of the received LF bits
  - The largest maximum service data unit (MSDU) size supported by the bridge
  - The largest MSDU size supported by the port from which the frame was received
  - The largest MSDU size supported by the port on which the frame is to be transmitted.

  If necessary, the destination station further reduces the LF value to indicate its maximum frame capacity.

  LF bit encoding is made up of a 3-bit base encoding and a 3-bit extended encoding (6 bits total). The SRT bridge (explained in a later section) contains an LF mode indicator allowing the bridge to select either base or extended LF bits. When the LF mode indicator is set to the *base mode*, the bridge sets the LF bits in explorer frames with the largest frame base values. When the LF mode indicator is set to *extended mode*, the bridge sets the LF bits in explorer frames in with the largest frame extended values.

- *Route Designator fields (RDn)* indicates the specific route through the network according to the sequence of the RD fields. Each RD field contains a unique network 12-bit ring number and 4-bit bridge number that differentiates between two or more bridges when they connect the same two rings (parallel bridges). The last bridge number in the routing information field has a null value (all zeros).

The Spanning Tree Explore Option

The spanning tree explore feature lets you select a single route to a destination when your network has two or more bridges connecting the same LANs. With this feature enabled, only the bridges you select receive spanning tree explorer (STE) frames. Not to be confused with the spanning tree protocol, this option allows you to:

- Simulate a spanning tree network
- Balance traffic loads

Simulating a Spanning Tree Network

A spanning tree network contains a single data route between any two end stations. If your network uses two or more parallel bridges, such as those in Figure 2-7, you can manually configure a spanning tree in a network by preventing duplication of discovery frames onto the network. Without spanning tree explore enabled, if Station Q transmits a discovery frame to a Station R, both Bridge A and Bridge B retransmit that frame. Segment 2 then receives two copies of the same frame.

With spanning tree explore enabled, each LAN segment on the network receives only one copy of the transmitted frame. Only the bridges you select can receive STE frames, reducing the creation of redundant frames and lowering network overhead.

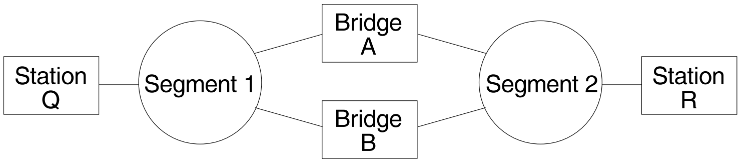


Figure 2-7. Example of Parallel Bridges

Balancing Traffic Loads

You can also use the spanning tree explore option for load balancing. For example, in Figure 2-8 on page 2-15, Bridge A is configured to accept STE frames over the interface connecting Segment 2. Bridge B is configured to accept STE frames over the interface connecting Segment 1. Traffic travels in the direction of the arrows. This configuration allows parallel bridges to share the traffic load.

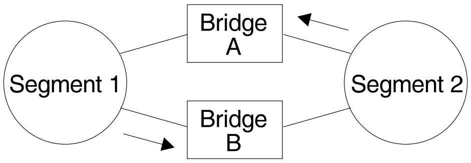


Figure 2-8. Using Spanning Tree Explore for Load Balancing

Note: For source routing to work, some end-node applications such as the IBM PC LAN program require you to enable spanning tree explore on attached interfaces. For parallel bridge configuration, the spanning tree explore option should be enabled only on one of the parallel interfaces. However no serious harm (other than some extra traffic) results from having too many interfaces enabled for the spanning tree.

If you use the spanning tree explore option and any bridge on the single-route path goes down, source routing traffic cannot reach its destination. You must manually reconfigure an alternate path.

Protocol Filtering

A single bridge platform can perform both bridging and routing. Protocol filtering is the process that determines whether the incoming data is routed or bridged. This decision is based on the contents of the destination address field of incoming frames.

Table 2-2 shows how the “Bridge or Route?” question is answered based on the destination address contents.

Table 2-2. Route/Bridge Decision Table

If the Destination MAC Address in the Received Frame Contains:	The Bridge takes this action:
Bridge Address	The bridge passes the frame to the configured protocol that routes the frame.
Multicast or Broadcast Address	If there is a configured protocol in the frame, the bridge routes the frame. If there is no configured protocol in the frame, the frame is dropped.
Unicast	The frame is bridged.

Source Route Bridging and Frame Relay

The Frame Relay interface forwards source-routed frames to and from the bridging forwarder provided source routing bridging is enabled on the permanent virtual circuit (PVC). A destination ring number is configured for each PVC. Some PVCs that are not part of the active data path are BLOCKED in order to maintain the loop-free topology.

Source Routing Bridge Terminology and Concepts

This section reviews the terms and concepts commonly used in source routing bridging.

Bridge Instance

The bridge instance identifies the sequence of a bridge defined in the software. For example, in a bridge with two configured bridges, the bridge instances would be 1 and 2.

Bridge instances within a single bridge are independent and do not communicate. For example, in Figure 2-9, Station A cannot pass data to either station on Bridge Instance 2. It can pass frames only to Station B. In effect, the bridge instance allows you to create two separate networks. These networks do not communicate unless they physically interconnect at some other point.

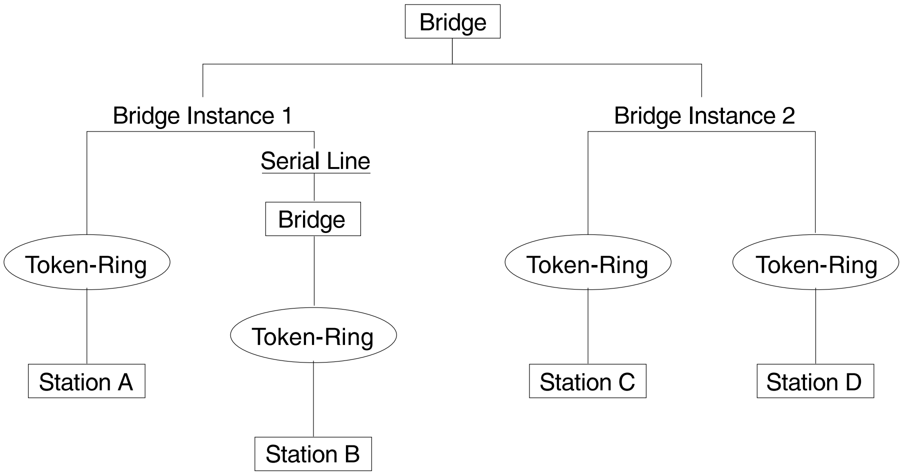


Figure 2-9. Bridge Instances within a Bridge

Bridge Number

The bridge number is a 4-bit hexadecimal value that identifies a bridge. Although bridges which are attached to the same ring can have the same bridge number, parallel bridges (bridges that are connected to the same two rings) must have unique bridge numbers.

Explorer Frames

The source routing bridge adds routing information to an explorer frame as it forwards the frame through the network to its destination end station. The explorer frame is used to discover routes. There are two types of explorer frames: all-routes explorer (ARE) frames and spanning-tree explorer (STE) frames. ARE frames are forwarded by all ports while STE frames are forwarded only by ports assigned to forward them by the spanning tree protocol.

Interface Number

The interface number identifies a “physical” interface within the hardware/product and must be tied to the “logical” interface that is understood by a bridge (that is a port). When you configure the router software, the router/bridge numbers the ports sequentially. To use the source routing bridge, you must use the port numbers to identify the interface that connects each network segment.

Route

The route is a path through a series of LANs and bridges for example, SRB bridges.

Route Discovery

Route discovery is the process by which a route is learned to a destination end station.

Segment Number

The segment number identifies each individual LAN, such as a single token-ring or serial line. A segment connects to the bridge, but can also operate independently.

Source Routing

Source routing is a bridging mechanism that routes frames through a multi-LAN network by specifying in the frame the route it will travel.

Source Routing Transparent (SRT) Bridge

Having worked hard to adopt standardized technologies (Ethernet and token-ring are both defined by IEEE), you may actually be forced back into the proprietary arena when trying to connect them. This is because bridges function differently in token-ring and Ethernet networks.

Aside from the differences such as bit-ordering, packet size, and acknowledgement bits, differences in bridging methods are another obstacle. Ethernet bridges use the transparent bridging method in which the bridges determine the route of the traffic through the network. Token-ring networks use transparent bridging only in some instances, so they generally depend on source routing as the primary bridging method.

Source routing cannot operate in a transparent environment because transparent packets contain no routing information. In this case, the bridge has no way of knowing whether to forward the packet. While transparent bridging can operate in a source routing environment, it does so without any routing information being passed to an end station. Significant information (for example, packet sizing) is missing and can potentially create problems.

IEEE has ratified an extension to the 802.1D transparent bridging standard called source routing transparent (SRT). SRT is a bridging technology that attempts to resolve a large part of the incompatibility issues inherent in bridging token-ring and Ethernet. It saves you the cost of installing multiple bridges and separate links to support the two types of traffic by adding a parallel bridging architecture (rather than an alternative) to the transparent bridging standard.

The following sections describe SRT Bridging in more detail:

- “General Description” on page 2-18
- “Source Routing Transparent Bridge Operation and Architecture” on page 2-18
- “SRT Bridge and Frame Relay” on page 2-19
- “Source Routing Transparent Bridge Terminology” on page 2-19

General Description

A Source Routing Transparent (SRT) bridge is a MAC bridge that performs source routing when source routing frames with routing information are received and that performs transparent bridging when frames are received without routing information. In SRT, all the bridges between Ethernets and token-rings are transparent. The bridges operate at the MAC sublayer of the data link layer and are completely invisible to the end stations.

The SRT bridge distinguishes between the two types of frames by checking the value in the RII field of the frame (see “Source Routing Frames” on page 2-11 for more information). An RII value of 1 indicates that the frame is carrying routing information while a value of 0 in the RII indicates that no routing information is present. With this method, the SRT bridge forwards transparent bridging frames without any conversions to the outgoing media (including token-ring). Source routing frames are restricted to source routing bridging domain.

The spanning tree protocol and algorithm forms a single tree involving all the networks connected by SRT bridges. The SRT-bridged network offers a larger domain of transparent bridging with sub-domain of source routing. Thus, transparent frames are capable of reaching to the farthest side of the SRT- and TB-bridged LAN while source routed frames are limited to only SRT- and SRB-bridged LAN. In the SRT bridging model, source routing and transparent bridging parts use the same spanning tree. In the SRT-bridged domain, end stations are responsible for answering the “Source Routing or Transparent Bridging” question.

Source Routing Transparent Bridge Operation and Architecture

With an SRT bridge, each bridge port receives and transmits frames to and from the attached local area networks using the MAC services provided by the individual MAC entity associated with that port. The MAC relay entity takes care of the MAC-independent task of relaying frames between bridge ports. If the received frame is not source-routed (RII = 0), then the bridge frame is forwarded or discarded using the transparent bridging logic. If the received frame is source-routed (RII = 1), then the frame is handled according to the source routing logic. This process is illustrated in Figure 2-10. The arrows represent the data path.

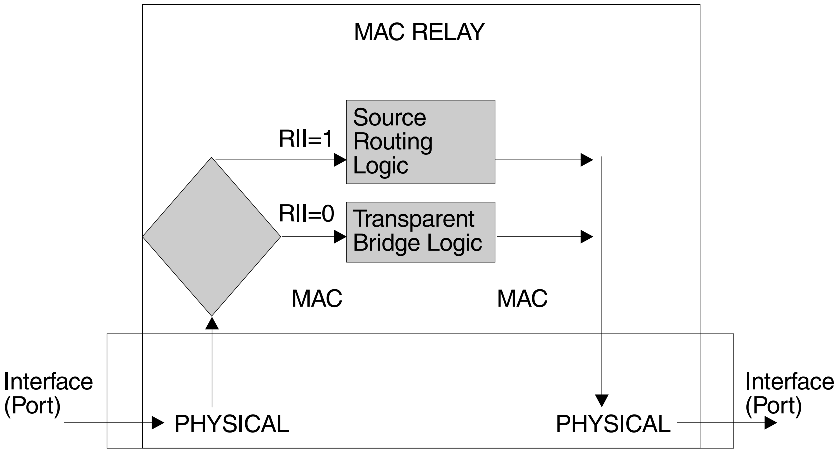


Figure 2-10. SRT Bridge Operation

SRT differentiates between source-routed and non-source-routed traffic on a frame-by-frame basis. If the packet is source-routed, the bridge forwards it as such. If it is a transparent bridge packet, the bridge determines the destination address and processes it as an Ethernet.

SRT Bridge and Frame Relay

The frame-relay interface supports SRT bridging by forwarding all bridged frames to the appropriate bridging forwarder, provided bridging is enabled on the PVC.

Source Routing Transparent Bridge Terminology

This section reviews the terms and concepts commonly used in SRT bridging.

Explorer Frames

The source routing bridge adds routing information to an explorer frame as it forwards the frame through the network to its destination end station. The explorer frame discovers routes. There are two types of explorer frames:

- All-routes explorer (ARE) frames
- Spanning-tree explorer (STE) frames

ARE frames are intended to be forwarded by all ports while STE frames are forwarded only by ports assigned to forward them by the spanning tree protocol.

Routing Information Field (RIF)

In source routing, the data frame forwarding decision is based on routing information within the frame. Before forwarding the frame end stations obtain the route to the destination station by the *route discovery* process. The station that originates the frame (that is, the *source* station) designates the route that the frame will travel by imbedding a description of the route in the Routing Information Field (RIF) of the transmitted frame.

Routing Information Indicator (RII)

Because source routing MAC frames contain routing information necessary for data communication over multi-ring environments, their format differs slightly from the typical token-ring MAC frames. The presence of a 1 in the source address field called the Routing Information Indicator indicates that a Routing Information Field containing routing information follows the source address. The SRT bridge distinguishes between source-routed and non-source-routed frames by checking for a 1 or 0 value in the RII field.

Source Routing

Source routing is a bridging mechanism that routes frames through a multi-LAN network by specifying in the frame the route it will travel.

Spanning Tree

The spanning tree is a topology of bridges in which there is only one data route between any two end stations.

Transparent Bridging

This type of bridging involves a mechanism that is transparent to end stations. Transparent bridging interconnects local area network segments by bridges designated to forward data frames through in a spanning tree algorithm.

ASRT Bridge Overview

The Adaptive Source Routing Transparent (ASRT) bridge is a software collection of several bridging options. The ASRT bridge software combines transparent bridging and source routing so that they function separately or can be combined as single ASRT bridge. This extended function allows communication between a strict source routing end station and a transparent end station via an ASRT bridge. Depending on the set of configuration commands used, the ASRT bridge provides the following bridging options:

- Transparent Bridge (STB)
- Source Routing Bridge (SRB)
- Source Routing Transparent Bridge (SRT)
- Source Routing—Transparent Bridge (SR-TB)

The ASRT bridge is modeled after the Source Routing Transparent bridge described in IEEE 802.5M/Draft 6 (1991) of SRT. Modifications have been built into the ASRT bridge which provide users with extended function that goes beyond compliance with the SRT standard. The ASRT bridge allows compatibility to the installed base of source routing bridges, while still enabling them to link Ethernet and token-ring LANs. ASRT also enhances basic SRT function in some additional, critical ways described in the following sections.

Adaptive Source Routing Transparent Bridge (ASRT) (SR-TB Conversion)

While source routing is still available in the SRT model, it is available only between adjacent source routing token-rings. Source routing-only bridges cannot coexist with SRT bridges that link Ethernet and token-ring LANs. Because a token-ring end node needs to communicate with an Ethernet node, it must be configured to omit RIFs. Also, if the end node is configured to omit RIFs, it cannot communicate through ordinary source routing bridges that require that RIF.

The following sections describe the ASRT bridge in detail:

- “General Description” on page 2-21
- “Source Routing—Transparent Bridge Operation” on page 2-21
- “SR-TB and Frame Relay” on page 2-27
- “Source Routing—Transparent Bridge (SR-TB) Terminology and Concepts” on page 2-27
- “Transparent-Source Routing Compatibility - Issues and Solutions” on page 2-28
- “ASRT Configuration Considerations” on page 2-29

General Description

The Source Routing - Transparent Bridge (SR-TB) option interconnects networks using source routing bridging (source routing domain) and transparent bridging (transparent bridging domain). It transparently joins both domains. During operation, stations in both domains are not aware of the existence of each other or of the SR-TB bridge. From a station's point of view, any station on the combined network appears to be in its own domain.

The bridge achieves this function by converting frames from the transparent bridging domain to source routing frames before forwarding them to the source routing domain (and vice versa). This is accomplished by the bridge maintaining a database of end station addresses each with its Routing Information Field in the source routing domain. The bridge also conducts route discovery on behalf of the end stations present in the transparent bridging domain. The route discovery process is used to find the route to the destination station in the source routing domain. Frames sent to an unknown destination are sent in the spanning tree explorer (STE) format.

The SR-TB bridge anticipates three types of spanning trees:

- A spanning tree formed by transparent bridge domain
- A spanning tree formed by source routing bridge domain
- A special spanning tree of all SR-TB bridges

The next sections discuss the operation of the SR-TB bridge in more detail.

Source Routing—Transparent Bridge Operation

During SR-TB operation, a network is partitioned into a series of two or more separate domains. Each domain is made up of a collection of LAN segments interconnected by bridges all operating under a common bridging method. This allows networks comprised of two types of domains (depending on the bridging method):

- Source routing domains
- Transparent bridging domains

Figure 2-11 on page 2-22 shows an example of these domains. With separate domains, each source routing domain has a single-route broadcast topology set up for its bridges. Only bridges belonging to that source routing *spanning tree* are designated to forward single-route broadcast frames. In this case, frames that carry the single-route broadcast indicator are routed to every segment of the source routing domain. Only one copy of the frame reaches each segment since the source routing spanning tree does not allow multiple paths between any two stations in the domain.

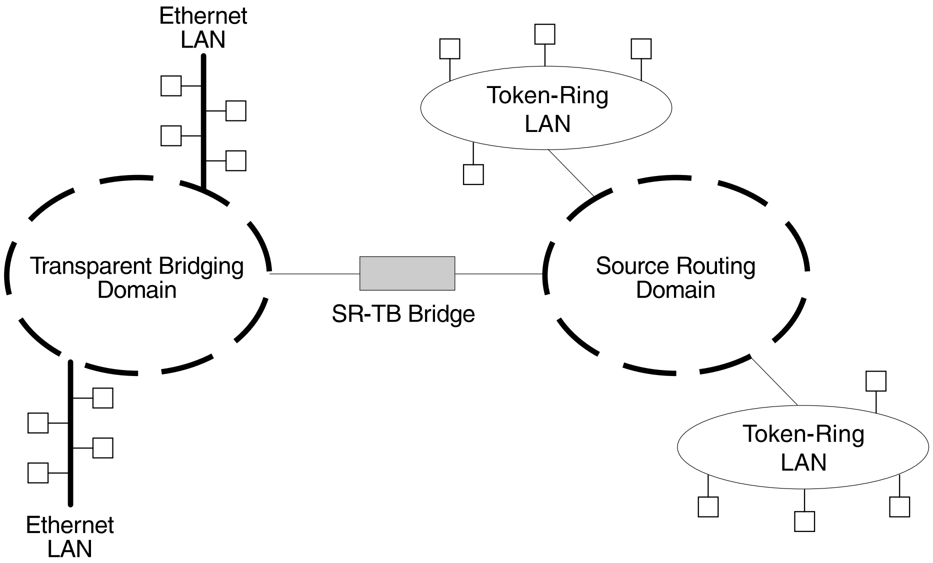


Figure 2-11. SR-TB Bridge Connecting Two Domains

Specific Source Routing and Transparent Bridging Operations

The SR-TB bridge is a *two-port device* with a MAC interface assigned to the LAN segment on the source routing side and another assigned to the LAN segment on the transparent bridging side. Each end station reads the appropriate MAC layer for its LAN segment. This means that bridging functions can be divided into two types of operations:

- Transparent bridging operations
- Source routing bridging operations

On the transparent bridging side, the SR-TB bridge operates the same as any other transparent bridge. The bridge keeps a table of addresses for stations it knows are transparent bridging stations. The SR-TB bridge observes the *inter-bridge* protocols necessary to create and maintain the network spanning tree since more than one SR-TB bridge joins different domains.

The SR-TB bridge forwards the frames received from its transparent bridging station to the source routing side of the bridge only if the destination address carried in the frame is not found in the bridge’s transparent bridging side address table.

On the source routing bridging side, the SR-TB bridge combines the functions of a source routing bridge and a source routing end station in a specific way. As a source routing end station, the bridge maintains an association of destination addresses and routing information on the source routing side. It communicates either as an end station for applications in the bridge itself (for example, network management) or as an intermediary for stations on the transparent bridging side.

The SR-TB bridge forwards the frames received from its transparent bridging station to the source routing side of the bridge only if the destination address carried in the frame is not found in the bridge’s transparent bridging side address table. Frames transmitted by the bridge’s source routing station carry the routing information associated with the bridge, if such information is known and held by the bridge.

As a source routing bridge, the SR-TB bridge participates in the route discovery process and in the routing of frames already carrying routing information. The route designator unique to the SR-TB bridge consists of the LAN number of the individual LAN on its source routing side and the bridge's individual bridge number.

The bridge also maintains a single LAN number representing all of the LANs on the transparent bridging side. The SR-TB bridge treats each case of received and forwarded frames differently as described in Table 2-3.

Table 2-3 (Page 1 of 2). SR-TB Bridge Decision Table

type of frame received	**action taken by SR-TB Bridge**
Non-routed frames received by the source routing station.	Does not copy or forward frames carrying routing information.
All-routes broadcast frame received by the source routing station.	Copies frame and sets A and C bits of the broadcast indicator in the repeated frame. If the destination address is in the transparent bridging table, the bridge forwards the frame without routing information on the transparent bridging network. Otherwise, the frame is not forwarded.
Non-routed frames received by the source routing station.	Does not copy or forward frames carrying routing information.
Single-route broadcast frame received by the source routing station. the bridge is not designated as single-route broadcast bridge.	Does not copy or forward the frame.
All-routes broadcast frame received by the source routing station.	Copies frame and sets A and C bits of the broadcast indicator in the repeated frame. If the destination address is in the transparent bridging table, the bridge forwards the frame without routing information on the transparent bridging network. Otherwise, the frame is not forwarded.
Single-route broadcast frame received by the source routing station. The bridge is designated as single-route broadcast bridge.	Copies frame, sets A and C bits in the broadcast indicator, removes the routing information from the frame, and forwards modified frame to transparent bridging side. Adds its bridge number to the saved routing information field and the LAN number for transparent bridging side. Changes the broadcast indicator to non-broadcast, complements D-bit, and stores this routing information for the source address of the frame.
Single-route broadcast frame received by the source routing station. The bridge is not designated as single-route broadcast bridge.	Does not copy or forward the frame.

Table 2-3 (Page 2 of 2). SR-TB Bridge Decision Table

type of frame received	action taken by SR-TB Bridge
Single-route broadcast frame received by the source routing station. The bridge is designated as single-route broadcast bridge.	Copies frame, sets A and C bits in the broadcast indicator, removes the routing information from the frame, and forwards the modified frame to the transparent bridging side. Adds its bridge number to the saved routing information field and the LAN number for transparent bridging side. Changes the broadcast indicator to non-broadcast, complements D-bit, and stores this routing information for the source address of the frame.
Non-broadcast frame received by the source routing station.	If frame carries specific route, bridge examines the routing information. If SR-TB bridge is part of the route and appears between the LAN number for the source routing side and LAN number for transparent bridge side, bridge copies frame and sets A and C bits in the repeated frame. Forwards frame to the transparent bridging side without routing information. If bridge does not already have a permanent route for the source address, it saves a copy of the routing information, complements D-bit, and stores saved routing information for the source address of the frame.
Frame received from the Transparent bridging side.	To forward frame to the source routing side, bridge first determines if it has routing information associated with the destination address carried in the frame. If yes, bridge adds routing information to the frame, sets the RII to 1, and queues the frame for transmission on the source routing side. If no, bridge adds a routing control field to the frame containing an indicator for single-route broadcast and two route designators containing the first two LAN numbers and its own individual bridge number.

SR-TB Bridging: Four Examples

The SR-TB bridge interconnects source routing domains with transparent bridging domains by transparently joining the domains. During operation, stations in both domains are not aware of the existence of each other or of the SR-TB bridge. From the end station’s point of view, any station on the combined network appears to be in its own domain.

The following sections provide specific examples of frame forwarding during SR-TB bridging. These examples assume that the SR-TB bridge is designated as a single-route broadcast bridge. Figure 2-12 on page 2-25 provides the following information to accompany the situations described in each section:

- Q is the bridge’s own bridge number
- X is the LAN number for the LAN on the source routing side
- Y is the LAN number for the LAN on the transparent bridging side
- A, B, C, and D represent end stations

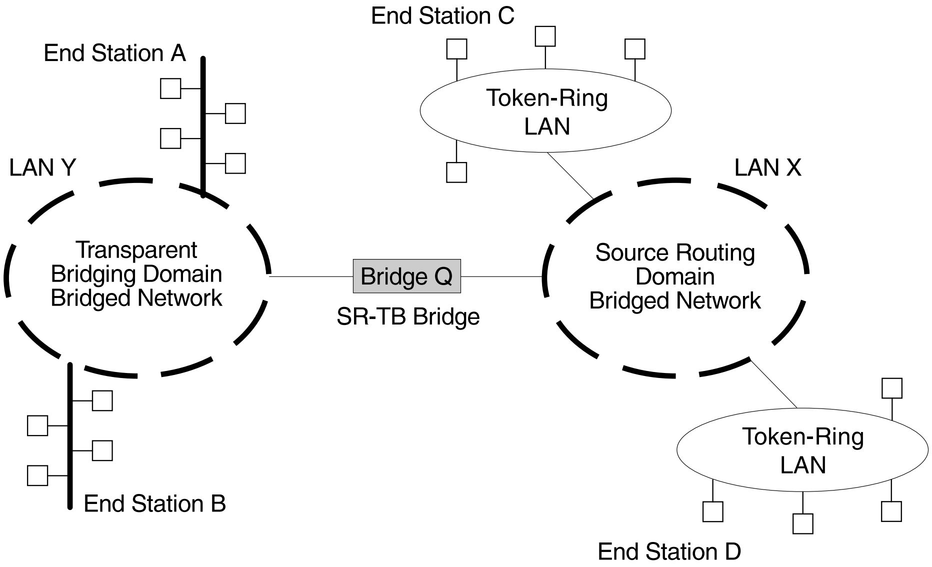


Figure 2-12. SR-TB Bridging Examples

Example 1: Frame Sent from End Station A to End Station B

When the SR-TB bridge receives a frame with a source address of end station A and a destination address of end station B, it enters end station A's address into its transparent bridging side address table. This table contains the addresses of stations known to be on the transparent bridging side of the bridge which is the normal process for transparent bridging.

If end station B's address is in the transparent bridging side's address table, the SR-TB bridge does not forward the frame. If end station B's address is not in the transparent bridging side's address table and not in the source routing side's address table, its location is not known to the SR-TB bridge. In this case, the frame is forwarded on the source routing side as a single-route broadcast with no request for route explorer return. Any frame sent by end station B (regardless of its destination) causes its address to be added to the transparent bridging address table. This prevents future forwarding of frames addressed to end station B to the source routing side.

Example 2: Frame Sent from End Station A to End Station C

In this example, end station A's address is treated the same as the previous example. Since end station C's address will definitely not be in the transparent bridge address table, the SR-TB bridge will forward the frame on the source routing side.

The bridge then looks for end station C's address in its source routing address table. This table contains all known addresses with related routing information for stations known to be on the source routing side of the bridge. If C's address is in the source routing table, the bridge forwards the frame using the routing information in the address table. If C's address is not in the source routing table (or if it appears but has null routing information), the bridge forwards the frame on the source routing side as a single-route broadcast with no request for route explorer return.

When end station C receives this frame, it enters end station A’s address in its source routing table together with the reverse direction of the route built from the SR-TB bridge and marks it as a temporary entry. When end station C later tries to send a frame to end station A, it will use this specific route, and because the route is marked as temporary, the frame will be sent as a non-broadcast route *with* a request for route explorer return.

When the returning frame arrives at the SR-TB bridge, it is forwarded on the transparent bridge side without routing information but will cause the route to end station C to be entered in the source routing table as a temporary route. This further causes the network management entity to send a route-explorer frame with an all-routes broadcast setting back to end station C. This lets end station C select the optimal routing for frames addressed to end station A to be entered as a permanent route in the SR-TB bridge's source routing table.

Example 3: Frame Sent from End Station C to End Station D

If the frame is sent as a non-broadcast and crosses over the segment to which the SR-TB bridge is attached, the bridge scans the RII field for the routing sequence (LAN X to Bridge Q to LAN Y). It cannot find the sequence and so will not forward the frame.

If the frame is sent as a single-route broadcast, the bridge will discard the frame if end station D is already known to be on the source routing side. If end station D is not known to be on the source routing side, the bridge forwards the frame to the transparent bridging side (minus the routing information), and adds “Q to Y” to the routing information. Finally, it saves the routing information for end station C as a temporary route in the source routing table with a non-broadcast indicator and the direction bit complemented.

If the frame is sent as an all-routes broadcast, the SR-TB bridge discards the frame (because end station D's address is not present in the transparent bridging address table) and makes sure that end station C's address is in the source routing table.

Example 4: Frame Sent from End Station C to End Station A

If the frame is sent non-broadcast, the bridge scans the RII field for the routing sequence (X to Q to Y). When it finds it, it forwards the frame to the transparent bridging side. It also stores the routing information for end station C.

If the frame is sent as a single-route broadcast, the bridge forwards the frame (minus the routing information) to the transparent bridging side and adds “Q to Y” to the routing information. It also sets the non-broadcast indicator, complements the direction bit, and enters the routing information for C's address in its source routing table.

If a temporary entry for end station C already exists in the source routing table, the SR-TB bridge updates the routing information. If the frame is sent as an all-routes broadcast, the bridge discards the frame but makes sure that end station C's address is in the source routing table.

SR-TB and Frame Relay

The frame-relay interface supports SR-TB bridging by forwarding all bridged frames to the appropriate bridging forwarder as long as bridging has been enabled on the PVC.

Source Routing—Transparent Bridge (SR-TB) Terminology and Concepts

This section reviews the terms and concepts used in SR-TB bridging.

All Routes Broadcast

The process of sending a frame through every non-repeating route in the bridged LAN.

All Stations Broadcast

The process of addressing a frame (placing all ones in the destination address) so that every station on the ring the frame appears on copies the frame.

Bridge

A bridge is a protocol-independent device that connects local area networks (LAN). Bridges operate at the data link layer, storing and forwarding data packets between LANs.

Bridge Number

The unique number identifying a bridge. It distinguishes between multiple bridges connecting the same two rings.

Explorer Frames

The source routing bridge adds routing information to an explorer frame as it forwards the frame through the network to its destination end station. The explorer frame discovers routes. There are two types of explorer frames: all-routes explorer (ARE) frames and panning-tree explorer (STE) frames. ARE frames are forwarded by all ports while STE frames are forwarded only by ports assigned to forward them by the spanning tree protocol.

Ring Number

The unique number identifying a ring in a bridged network.

Route

A path through a series of LANs and bridges (for example, source routing bridges).

Route Designator

A ring number and a bridge number in the Routing Information Field used to build a route through the network.

Route Discovery

The process of learning a route to a destination end station.

Segment Number

The segment number identifies each individual LAN, such as a single token-ring or serial line. A segment connects to the bridge, but can also operate independently.

Single Route Broadcasting

The process of sending a frame through a network such that exactly one copy of the frame appears on each ring in the network.

Source Routing Bridging

Source routing is a bridging mechanism that routes frames through a multi-LAN network by specifying in the frame the route it will travel.

Spanning Tree

The spanning tree is a topology of bridges such that there is only one data route between any two end stations.

Transparent Bridging

This type of bridging involves a mechanism that is *transparent* to end station applications. Transparent bridging interconnects local area network segments by bridges designated to forward data frames in a spanning tree algorithm.

Transparent-Source Routing Compatibility - Issues and Solutions

First, the ASRT bridge provides transparent bridge compatibility with ordinary source routing bridges through source routing bridge conversion (SR-TB). SR-TB was originally proposed as part of the 802.5 specification. This implementation is similar to and can interoperate with IBM's 8209 conversion bridge.

SR-TB converts transparent bridging frames to source routing frames and vice versa. In other words, instead of just checking to see whether an RIF is present in a packet and forwarding it to a like destination, the ASRT bridge can translate the packet into either format; it functions as either a transparent bridge or a source routing bridge by inserting or removing an RIF as necessary. With this function, packets can move between Ethernet and SRT token-ring LANs and still be compatible with an installed base of source routing token-ring LANs.

Elimination of Packet Size Problems

SR-TB also eliminates packet sizing problems in token rings being bridged together across an Ethernet domain. In this configuration, end stations use the source routing protocol which allows them to dynamically determine that there is a network with a 1518-byte maximum frame size between them. The end station automatically honors this limit without a manual reconfiguration. In the reverse situation, bridging Ethernets across a token-ring domain, packet size is not an issue because the token-ring packet size allowance is much larger.

Hardware Address Filtering

Another key feature provided by the ASRT bridge is hardware address filtering. Hardware address filtering solves the conflict in packet acknowledgement methods that exists in the Ethernet and token-ring LAN technologies. It occurs in the MAC layer and is the only technique that accurately sets acknowledgment bits based on the destination MAC address. The ASRT bridge uses content-addressable memories (CAMs) to implement hardware address filtering. This technology effectively gives the bridge a higher level of intelligence by providing instantaneous lookup of MAC addresses without creating any performance penalty.

Bit Ordering in STB (802.3) and SRB (802.5) Bridges

As bridges are continually being built to connect LANs with different MAC address types, bit ordering during data transmission affects the inter-operability of these technologies.

In administering MAC addresses, IEEE assigns addresses known as 48-bit IEEE globally assigned unique MAC addresses. These addresses are supported by 802.3, 802.4, and 802.5 LANs. Due to the lack of standards at the time this addressing scheme was developed, two different situations have arisen:

- 802.3 (Ethernet) and 802.4 LANs transmit source and destination addresses with the group bit first and LLC data fields transmitted least-significant bit (LSB) first.
- 802.5 (token-ring) LANs transmit source and destination addresses with the group bit first and LLC data fields transmitted most-significant bit (MSB) first.

Note: For simplicity sake, 802.3 and 802.4 bridges and LANs will now be referred to as LSB bridges and LANs. 802.5 bridges and LANs will be referred to as MSB bridges and LANs.

The difference in the bit transmission standard means that a bridge from LSB to MSB LANs has to reverse the bit order of the destination and source MAC addresses at the start of the MAC frame. This is because the different LAN types use the same bit order for the MAC address (that is, group bit first) and yet use a different bit order for the user data (either LSB or MSB first).

The misinterpretation of addresses due to reversed bit ordering is compounded by the fact that some of the higher level communications protocols misinterpret MAC addresses altogether. Protocols such as IP and Novell IPX interpret bridging addresses incorrectly because at the time of their initial development, there was no standard representation of MAC addresses.

The bit order differential is best resolved by combining bridging technology (data link layer technology) with routing technology (network layer technology). Rather than ask the user to “reverse engineer” today’s communications protocols and configure each bridge to “flip” or reverse addresses on a case-by-case basis, the problem is more easily solved by routing these protocols.

Routing eliminates the bit order and protocol addressing problems by accessing the detailed packet addresses running at the higher layer. Routing alone is not a complete solution, because other protocols such as IBM Frames and NetBIOS cannot be routed, and SNA routing is limited. Therefore, it is important to implement SRT in a device where bridging and routing work hand-in-hand.

ASRT Configuration Considerations

The ASRT bridge uses the spanning tree protocol and algorithm described in the IEEE 802.1D bridge standard over all interfaces. It is possible that more than one spanning tree will form in an environment where different types of bridges exist. For example a spanning tree of all bridges practicing IEEE 802.1d protocol (for example, STB and SRT) existing with another tree of IBM 8209 bridges. The loops forming from this configuration require you to correct the situation.

TCP/IP Host Services support SDLC relay. When running as a pure bridge, and not as an IP router, functions usually associated with an IP router are not available. For example, there is no BootP forwarder function or any ARP subnet routing capabilities.

ASRT Configuration Matrix

With an ASRT Bridge, the collection of configuration parameters for the bridge and all connected interfaces produces a *bridge personality* for that bridge. The following matrix provides a guide to the configuration settings needed for each interface type to produce the desired bridge personality to handle your network.

		Interface Type & Bridging Method Setting		
Bridge Personality	**SR <-> TB Conversion Enabled?**	**Token Ring**	**Ethernet**	**Serial Line or Tunnel**
STB	No	TB	TB	TB
SRB	No	SR	--	SR
STB & SRB	No	SR	TB	TB or SR
SR TB	Yes	SR	TB	TB
SR TB	Yes	SR	TB	SR
SRT	No	SR & TB	TB	SR & TB
ASRT	Yes	SR & TB	TB	SR & TB
ASRT	Yes	SR	TB	SR & TB
ASRT	Yes	SR or TB	TB	SR & TB

Bridge Personality Key:

STB = Transparent (Spanning Tree) Bridge SRB = Source Routing Bridge SR TB = Source Routing Transparent Conversion Bridge SRT = Source Routing Transparent Bridge ASRT = Source Routing Transparent Bridge

Bridging Method Key:

SR = Source Routing TB = Transparent Bridging

Chapter 3. Bridging Features

This chapter describes bridging features that are available with the Adaptive Source Routing Transparent (ASRT) bridge. The chapter includes the following sections:

- “Bridging Tunnel”
- “TCP/IP Host Services (Bridge-Only Management)” on page 3-3
- “Bridge-MIB Support” on page 3-3
- “NetBIOS Name Caching” on page 3-3
- “Duplicate Frame Filtering” on page 3-4
- “NetBIOS Name and Byte Filters” on page 3-4
- “Multiple Spanning Tree Protocol Options” on page 3-7
- “Threading (Router Discovery)” on page 3-8

Bridging Tunnel

The bridge tunnel (encapsulation) is another feature of the ASRT bridge software. By encapsulating packets in industry-standard TCP/IP packets, the bridging router can dynamically route these packets through large IP internetworks to the destination end-stations.

End stations see the IP path (the tunnel) as a single hop, regardless of the network complexity. This helps overcome the usual 7-hop distance limit encountered in source routing configurations. It also lets you connect source routing end-stations across non-source routing media, such as Ethernet networks.

The bridging tunnel also overcomes several limitations of regular source routing including:

- Distance limitations of seven hops

- Large amounts of overhead that source routing causes in wide area networks (WANs)

- Source Routing’s sensitivity to WAN faults and failures (if a path fails, all systems must restart their transmissions)

With the bridge tunnel feature enabled, the software encapsulates packets in TCP/IP packets. To the router, the packet looks like a TCP/IP packet. Once a frame is encapsulated in an IP envelope, the IP forwarder is responsible for selecting the appropriate network interface based on the destination IP address. This packet can be routed dynamically through large internetworks without degradation or network size restrictions. End-stations see this path or tunnel as a single hop, regardless of the complexity of the internetwork. Figure 3-1 on page 3-2 shows an example of an IP internetwork using the tunnel feature in its configuration.

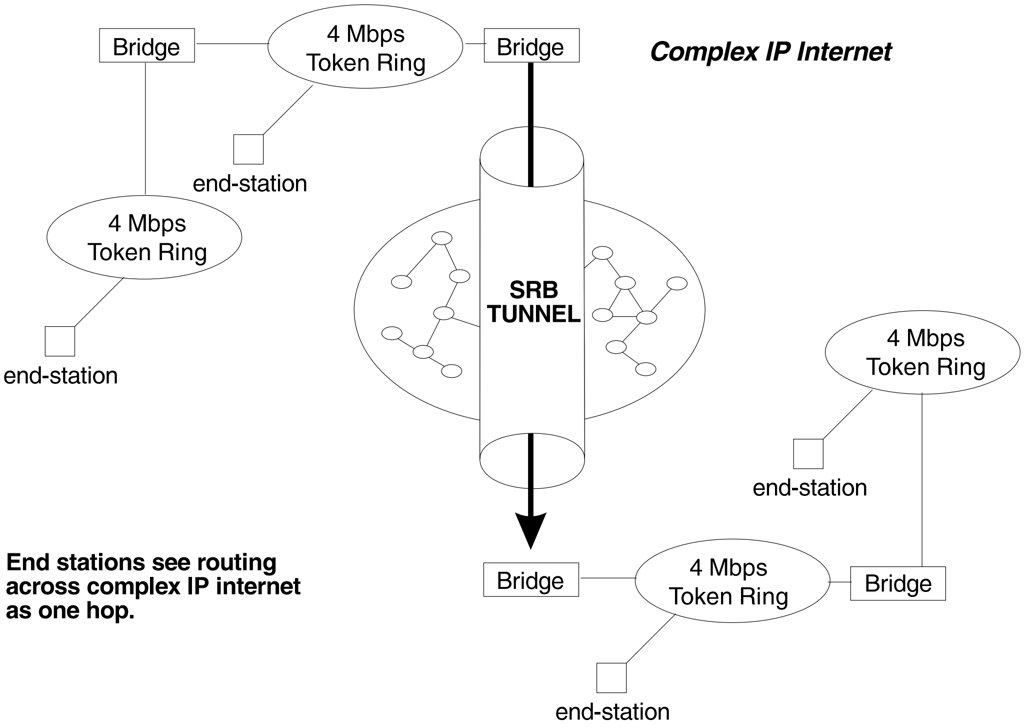


Figure 3-1. Example of the Bridge Tunnel Feature

The tunnel is transparent to the end stations. The bridging routers participating in tunneling treat the IP internet as one of the bridge segments. When the packet reaches the destination interface, the TCP/IP headers are automatically removed and the inner packet proceeds as a standard source routing packet.

Encapsulation and OSPF

A major benefit of the encapsulation feature is the addition of the OSPF dynamic routing protocol to the routing process. OSPF offers the following benefits when used with encapsulation:

- *Least-Cost Routing*. OSPF accesses the fastest path (tunnel) with the fewest delays allowing network administrators to distribute traffic over the least expensive route.
- *Dynamic Routing*. OSPF looks for the least-cost path as well as detects failures and reroutes traffic with low overhead
- *Multi-Path Routing*. Load sharing makes more efficient use of available bandwidth.

With OSPF, tunnels automatically manage paths inside the internetwork. If a line or bridge fails along the path then the tunnel bridge automatically reroutes traffic along a new path. If a path is restored, the tunnel automatically updates to the best path. This rerouting is completely transparent to the end-stations. For more information on OSPF, see the configuration and monitoring chapters beginning at Chapter 23, “Using and Configuring OSPF” on page 23-1.

TCP/IP Host Services (Bridge-Only Management)

The bridging router also supports TCP/IP Host services which let you configure and monitor a bridge when routing functions are disabled. This option gives you the following capabilities:

- Management through SNMP
- Telnet server function
- Downloading and uploading of configuration through the TFTP protocol
- TFTP neighbor boot function
- IP diagnostic tools of ping and trace route
- Control of the device through SNMP sets and the telnet client

When viewed from the bridge's console interface, TCP/IP Host Services is handled as a new protocol having its own configuration and monitoring consoles. These prompts are accessed via the **protocol** command in the `Config>` and + (GWCON) consoles.

Bridge-only management function is activated by assigning an IP address to the bridge and enabling TCP/IP Host Services (see Chapter 13, "Configuring TCP/IP Host Services" on page 13-1). This IP address is associated with the bridge as a whole, instead of being associated with a single interface. When booting over the network, the bridge's IP address and a default gateway can be learned automatically through the ROMCOMM interface with the boot PROMs. Default gateway assignments can also be user-configured.

TCP/IP host services is available whenever bridging is an option in the router software load. These services coexist with the IP routing function but do not require IP routing be present.

Bridge-MIB Support

For Bridge Management via SNMP, the IBM Nways Multiprotocol Access Services supports the management information bases (MIBs) as specified by RFC 1493 and RFC 1525, *except* for the following MIBs:

- dot1dStaticTable
- dot1dTpFdbTable
- dot1dPortPairTable

For additional information on the MIBs, see Appendix A, "MIB Support" on page A-1.

NetBIOS Name Caching

The NetBIOS name caching feature enables the bridging router to significantly reduce the number of Name-Query frames that leave an originating ring and are forwarded through a bridge. Configuring for NetBIOS name caching is part of the NetBIOS configuration. Details are in "NetBIOS Name Caching and Route Caching" on page 9-12.

Duplicate Frame Filtering

Three frame types are typically sent in groups of six:

- Name-Query
- Add-Name
- Add-Group-Name

Duplicate frame filtering uses a timer to allow only one instance of each type of frame to be forwarded through the bridge in the amount of time set by the user.

This process uses a separate database from the one used in Name Caching. Duplicate frame database entries contain the client's MAC address and three time stamps, one for each of the mentioned frame types. Duplicate-frame filtering is processed before name caching. Details are in "Duplicate Frame Filtering" on page 9-5.

NetBIOS Name and Byte Filters

NetBIOS filtering is a feature that allows you to enhance the performance of ASRT Bridging. This feature lets you configure specific filters using the router configuration process. NetBIOS filters are sets of rules applied to NetBIOS packets to determine if the packets should be bridged (forwarded) or filtered (dropped).

Types of NetBIOS Filtering

There are two types of NetBIOS filtering, *host name* and *byte*:

Host name You implement host name filtering using fields in NetBIOS packets that let you select packets with specific NetBIOS host names to be bridged or filtered. Host name filters are for bridging only. You can use them based on NetBIOS source or destination names, depending on frame type.

Name filters apply to NetBIOS traffic that is being bridged or data link switched.

Byte You implement byte filtering using bytes (arbitrary fields) in NetBIOS packets that allow you to specify certain NetBIOS packets to be bridged or filtered.

There are no thresholds or timers associated with these filters and they remain active until you either disable or remove them. A NetBIOS filter is made up of three parts, the actual filter, filter lists, and filter items (described in more detail at "Building a Filter" on page 3-6).

Configuration and monitoring of NetBIOS is described at Chapter 9, "Using, Configuring, and Monitoring NetBIOS" on page 9-1. The remainder of this section describes NetBIOS host name filtering and NetBIOS byte filtering.

NetBIOS Host Name Filtering

NetBIOS filtering using host names lets you select packets with specific NetBIOS host names to be bridged or filtered. When you specify that packets with a particular NetBIOS host name (or set of NetBIOS host names) should be bridged or filtered, the source name or destination name field of the following NetBIOS packet types are examined:

- ADD_GROUP_NAME_QUERY (source)
- ADD_NAME_QUERY (source)
- DATAGRAM (destination)
- NAME_QUERY (destination)

Host name filter lists specify NetBIOS names that should be compared with source or destination name fields in the four different types of NetBIOS packets. The result of applying a host name filter list to a NetBIOS packet that is not one of those four types is *Inclusive*.

When configuring NetBIOS Filtering using host names, you specify which ports the filter is applied to and whether it is applied to input or output packets on those ports. Only NetBIOS Unnumbered Information (UI) packets are considered for filtering. Filtering is applied to NetBIOS packets that arrive at the router for either source route bridging (all RIF types) or transparent bridging.

When specifying a NetBIOS host name in a filter, you can indicate the 16th (last) character of the name, as a separate argument, in its hexadecimal form. If you do this, the first 15 bytes of the name are taken as specified and the 16th byte (if any is specified) is determined by the final argument. If you specify fewer than 16 characters (and no 16th byte), then the name is padded with ASCII blank characters up to the 15th character and the 16th character is treated as a wildcard.

When a specific NetBIOS host name is evaluated, that name is compared with only certain fields of certain NetBIOS packets. NetBIOS host names in filter items may include a wildcard character (?) at any point in the NetBIOS host name, or an asterisk (*) as the final character of a NetBIOS host name. The ? matches against any single character of a host name. The * matches against any one or more characters at the end of a host name.

NetBIOS Byte Filtering

Another filtering mechanism, byte filtering, lets you specify which NetBIOS packets should be bridged or filtered based on fields in the NetBIOS packets that relate to the MAC address. In this case, all NetBIOS packets are examined to determine if they match the configured filtering criteria.

To build a byte filter, you specify the following filter items:

- An offset from the beginning of the NetBIOS header
- A byte pattern to match on
- An optional mask to apply to the selected fields of the NetBIOS header

The length of the mask, if present, must be of equal length to the byte pattern. The mask specifies bytes that are to be logically added with the bytes in the NetBIOS header before the router compares the header bytes with the hex pattern for equality. If no mask is specified, it is assumed to be all ones. The maximum length for the hex pattern (and hence the mask) is 16 bytes (32 hexadecimal digits).

When configuring NetBIOS Filtering using specific bytes, you also specify which ports the filter is applied to and whether it is applied to input or output packets on those ports.

Building a Filter

Each filter is made up of one or more filter lists. Each filter list is made up of one or more filter items. Each filter item is evaluated against a packet in the order in which the filter items were specified.

When a match is found between a filter item and a packet, the router:

- Bridges the packet if the filter list is specified as *Inclusive*
- Drops the packet if the filter list is specified as *Exclusive*

If no filter items in the filter list produce a match, the router:

- Forwards the packet if the filter as a whole is specified as *Inclusive*
- Drops the packet if the filter as a whole is specified as *Exclusive*

A filter item is a single rule applied to a particular field of a NetBIOS packet. The result of the application of the rule is either an Inclusive (bridge) or an Exclusive (filter) indication. The following filter items can be configured with NetBIOS Filtering (the first two items are host name filters, the last two items are byte filters):

- Include NetBIOS host name optional 16th character (hex)
- Exclude NetBIOS host name optional 16th character (hex)
- Include decimal byte offset into NetBIOS hdr hex pattern starting at that offset hex mask
- Exclude decimal byte offset into NetBIOS hdr hex pattern starting at that offset hex mask

Part of the specification of a filter indicates whether packets that do not match any of the filter items in the filter list should be bridged (included) or filtered (excluded). This is the default action for the filter list. The default action for a filter list is initially set to Include, but this setting can be changed by the user.

Simple and Complex Filters

A simple filter is constructed by combining one filter list with a router port number and an input/output designation. This indicates that the filter list should be applied to all NetBIOS packets being received or transmitted on the given port. If the filter list evaluates to Inclusive, then the packet being considered is bridged. Otherwise, the packet is filtered.

A complex filter can be constructed by specifying a port number, an input/output designation, and multiple filter lists separated by one of the logical operators *and* or *or*. The filter lists in a complex filter are evaluated strictly left to right, and each filter list in the complex filter is evaluated. Each inclusive filter list result is treated as a true and each exclusive filter list result is treated as a false. The result of applying all the filter lists and their operators to a packet is a true or false, indicating that the packet is bridged or filtered. Each combination of input/port or output/port can have at most one filter.

Multiple Spanning Tree Protocol Options

The ASRT bridge lets you extend Spanning Tree protocol options to cover as many configuration options as possible. The next sections provide information on these features.

Background: Problems with Multiple Spanning Tree Protocols

Bridging technology employs different versions of spanning tree algorithms to support different bridging methods. The common purpose of each algorithm is to produce a loop-free topology.

In the spanning tree algorithm used by Transparent Bridges (TB), Hello BPDUs and Topology Change Notification (TCN) BPDUs are sent in a transparent frame to well known group addresses of all participating media (Token-Ring, Ethernet, and so on). Tables are built from this exchanged information and a loop free topology is calculated.

Source routing bridges (SRB) transmit spanning tree explorer (STE) frames across other SRB bridges to determine a loop-free topology. The algorithm sends Hello BPDUs in a transparent frame to well known functional addresses. Since TCN BDPUs are not used by SRB bridges, the port state setting created as a result of this spanning tree algorithm does not affect all route explorer (ARE) frame and specifically routed frame (SRF) traffic.

In bridging configurations using IBM 8209 Bridges, a different spanning tree method is used to detect parallel 8209 bridges. This algorithm uses Hello BPDUs sent as STE frames to IEEE 802.1d group addresses on the token ring. On the Ethernet, Hello BPDUs sent as transparent frames to the same group address are used. This method allows 8209s to build spanning trees with transparent bridges and other IBM 8209 bridges. It does not participate in the SRB spanning tree protocol, however, and Hello BPDUs sent by SRBs are filtered. Consequently, there is no way to prevent the 8209 from becoming the root bridge. The flip side of this situation is that if the 8209 bridge is selected as the root then traffic between two Transparent Bridge domains may have to pass through token-ring/SRB domains.

As you can see, running multiple spanning tree protocols can cause compatibility problems with the way algorithm creates its own loop-free topology.

STP/8209

The STP/8209 bridging feature is available to allow you to further extend the Spanning Tree protocol. Previously, SRBs allowed only manual configuration of a loop-free tree over the token-ring. This was the only mechanism to prevent loops in the case of parallel SR-TB bridges. With the addition of the STP/8209 feature the following spanning tree algorithm combinations are possible:

- Pure Transparent Bridge (TB) - IEEE 802.1d Spanning Tree protocol is used.
- Pure Source Routing Bridge (SRB) - SRB Spanning Tree protocol is used.
- Transparent and Source Routing Bridges as separate entities - IEEE 802.1d Spanning Tree protocol is used for TB and manual configuration (no Spanning Tree protocol) is used for SRB.
- SR-TB Bridge - IEEE 802.1d Spanning Tree protocol is used for TB ports and IBM 8209 BPDUs on SRB ports are used to form a single tree of TBs and

SR-TBs. SRB Hello BPDUs are allowed to pass on the SR domain but are not processed. IBM 8209 bridges filter such frames but this is allowed as it is a two-port bridge with the other port being a TB port.

- Pure SRT Bridge - *Only* IEEE 802.1d Spanning Tree protocol is used. SRB Hello BPDUs and IBM 8209 BPDUs are allowed to pass but are not processed.
- ASRT Bridge - IEEE 802.1d Spanning Tree protocol is used to make a tree with TBs and SRT bridges. “8209-like” BPDUs are also generated on all SR interfaces. These BPDUs are processed as soon as they are received. This causes two BPDUs to be generated and received on all SR interfaces. Because both BPDUs carry the same information, there will be no conflict of port information. This lets the ASRT bridge create a spanning tree with IBM 8209 and SR-TB bridges along with other TBs and SRT bridges.

Threading (Router Discovery)

Threading is a process used by a token-ring end station protocol (for example, IP, IPX, or AppleTalk) to discover a route to another end station through a source-routing bridged network.

The details of the threading process vary according to the end station protocol. The following sections describe the threading process for IP, IPX, and AppleTalk.

IP Threading with ARP

IP end-stations use ARP REQUEST and REPLY packets to discover a RIF. Both IP end-stations and the bridges participate in the route discovery and forwarding process. The following steps describe the IP threading process.

1. An IP end-station maintains an ARP table and a RIF table. The MAC address in the ARP table is used as a cross reference for the destination RIF in the RIF table. If a RIF does not exist for that specific MAC address, the end-station transmits an ARP REQUEST packet with an ARE (all routes explorer) or an STE (spanning tree explorer) onto the local segment.
2. All bridges on the local segment capture the ARP REQUEST packet and send it over their connected networks.

   As the ARP REQUEST packet continues its search for the destination end-station, each bridge that forwards it adds its own bridge number and segment number to the RIF in the packet. As the frame continues to pass through the bridged network, the RIF compiles a list of bridge and segment number pairs describing the path to the destination.

   When the ARP REQUEST packet finally reaches its destination, it contains the exact sequence of bridge and segment numbers from source to destination.
3. When the destination end-station receives the frame, it places the MAC address and its RIF into its own ARP and RIF tables. If the destination end-station should receive any other ARP REQUEST packets from the same source, that packet is dropped.
4. The destination end-station then generates an ARP REPLY packet including the RIF and sends it back to the source end-station.

5. The source end-station receives the learned route path. The MAC address and its RIF are then entered into the ARP and RIF tables. The RIF is then attached to the data packet and forwarded onto the destination.
6. Aging of RIF entries is handled by the IP ARP refresh timer.

IPX Threading

IPX end-stations check each packet they receive for a RIF. If the RIF does not exist in the table, they add the RIF to the table and designate that route as *HAVE_ROUTE*. If the RIF indicates that the packet came from an end-station on the local ring, the route is designated as *ON_RING*.

If the end-station needs to send out a packet and there is no entry in RIF table for the MAC address, the end-station transmits the data as an STE.

When the RIF timer expires, the entry in the table is cleared and will not be reentered until another packet arrives containing a RIF for that entry.

AppleTalk 2 Threading

AppleTalk end-stations use ARP and XID packets to discover a route. Both the AppleTalk end-stations and the bridges participate in the route discovery process and forwarding. The following steps describe the AppleTalk threading process.

1. If a RIF does not exist for a specific MAC address, the end-station transmits an ARP REQUEST packet with an ARE (all routes explore) onto the local segment.
2. All bridges on the local segment capture the ARP REQUEST packet and send it over their connected networks. As the ARP REQUEST packet continues its search for the destination end-station, each bridge that forwards it adds its own bridge number and segment number to the RIF in the packet. As the frame continues to pass through the bridged network, the RIF compiles a list of bridge and segment number pairs describing the path to the destination.
3. When the destination end-station receives the frame, it places the MAC address and its RIF into its own ARP and RIF tables and the state of the entry is designated as *HAVE_ROUTE*. If the destination end-station should receive any other ARP REQUEST packets from the same source, that packet is dropped.
4. The destination end-station then generates an ARP REPLY packet including the RIF and sends it back to the source end-station with the direction bit in the RIF flipped.
5. The source end-station receives the learned route path. The MAC address and its RIF are then entered into the ARP and RIF tables and the state is designated as *HAVE_ROUTE*. If the RIF indicates that the packet came from an end-station on the local ring, the route is designated as *ON_RING*.
6. If the RIF timer expires, an XID is sent out with an ARE and the state is changed to *DISCOVERING*. If no XID reply is received, the entry is discarded.

Chapter 4. Using the Boundary Access Node (BAN) Feature

This chapter describes the Boundary Access Node (BAN) feature on the 2216. BAN provides a reliable, low-cost way for attached PU Type 2.0 and 2.1 end stations to communicate with the SNA environment across wide-area links. This chapter includes the following sections:

- “About the Boundary Access Node Feature”
- “Using the BAN Feature” on page 4-4
- “Using Multiple DLCIs for BAN Traffic” on page 4-7
- “Checking the BAN Configuration” on page 4-8
- “Enabling Event Logging System (ELS) Messages for BAN” on page 4-9.

About the Boundary Access Node Feature

BAN can be used to attach to any of these SNA node types:

- End nodes
- Network nodes
- Subarea nodes.

The IBM Network Control Program (NCP) is an example of a subarea node and, in conjunction with VTAM, a composite APPN network node.

The BAN feature is an enhancement of the Frame Relay, DLSw, and Adaptive Source Route Bridging (ASRT) capabilities of the 2216 software. This feature enables IBM Type 2.0 and 2.1 end stations connected to an 2216 to make a direct connection via Frame Relay to a SNA node supporting the RFC 1490 Bridged 802.5 (Token-Ring) Frame format. The BAN feature provides a better, less costly way of communicating with the IBM SNA environment. IBM has modified the IBM Network Control Program (NCP) software in the IBM 3745 to support this enhancement.

When using BAN, end stations function as if they are directly connected to an SNA node via a Token-Ring, Ethernet, or SDLC line as shown in Figure 4-1. Though their data actually passes through an 2216 and over a Frame-Relay network, this is transparent to the end stations.

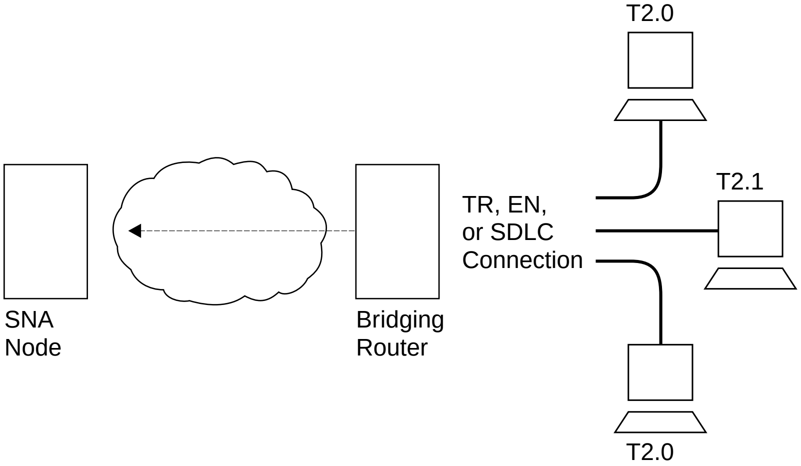


Figure 4-1. Direct Connection of End Stations to an SNA Node Using BAN

Benefits of BAN

Designed to meet the needs of customers who do not require a full DLSw implementation, BAN provides an economical method for connecting to IBM environments. Offering a path to full DLSw capability, BAN provides three major benefits to customers who need to internetwork with the IBM environment:

1. Ability to bridge Ethernet or Token-Ring traffic directly to the SNA node without frame conversion by another DLSw router. This can save capital equipment costs by eliminating the need for another router and a host at the central site.
2. No architectural limit to the number of multiplexed LLC Type 2 (*LLC2*) connections over a single frame-relay data link connection identifier (DLCI). In contrast, the existing NCP Frame Relay Boundary Node (BN) support limits the number of LLC2 connections per DLCI to 127. This can save significantly on frame-relay DLCI provider costs.
3. Eliminates the need to configure end station addresses on the DLSw router that is local to the end stations. This makes it easier to configure and manage the BAN setup.

   Note: You can use a BAN DLCI for IP traffic. This allows you to manage the router (via SNMP) over the same DLCI you are using for SNA (via BAN).

How BAN works

The BAN feature in the router works by filtering the frames sent by Type 2.0 or 2.1 end stations. Each BAN frame is modified by the router to comply with Bridged 802.5 (Token-Ring) Frame format. The router examines each frame, and allows only those with the BAN DLCI MAC address to pass over a DLCI to the mainframe. The destination MAC address in the bridged 802.5 frame is replaced with the Boundary Node Identifier in frames destined for the SNA node.

With BAN, only one DLCI ordinarily is needed. However, BAN may use many DLCI connections between the router and the IBM environment. In some cases, you may want to set up more than one DLCI to handle BAN traffic. See “Setting up Multiple DLCIs” on page 4-8 for more information.

There are two ways to use the BAN feature:

- Straight bridging using the 2216’s bridging capability
- DLSw terminated, in which BAN terminates the LLC2 connection at the router running DLSw.

The sections that follow explain how to configure each method.

Bridged Versus DLSw BAN

You can implement BAN in two ways: straight bridging and DLSw terminated. With straight bridging, you configure BAN to bridge LLC2 frames from Type 2.0 or Type 2.1 end stations straight into the SNA node. With DLSw Terminated, BAN terminates the LLC2 connection at the router running DLSw. In this discussion, we refer to straight bridging as *BAN Type 1* and DLSw Terminated as *BAN Type 2*.

Figure 4-2 on page 4-3 shows a BAN Type 1 (Bridged) connection. In this figure, notice that the router does not terminate the LLC2 traffic received from attached end stations. Instead, the router converts the frames it receives to Bridged Token-Ring format (RFC 1490) frames, and bridges directly to the SNA node.

SDLC devices can be supported using BAN Type 1 connections via DLSw. In this case the LLC2 traffic on the BAN connection is terminated by DLSw.

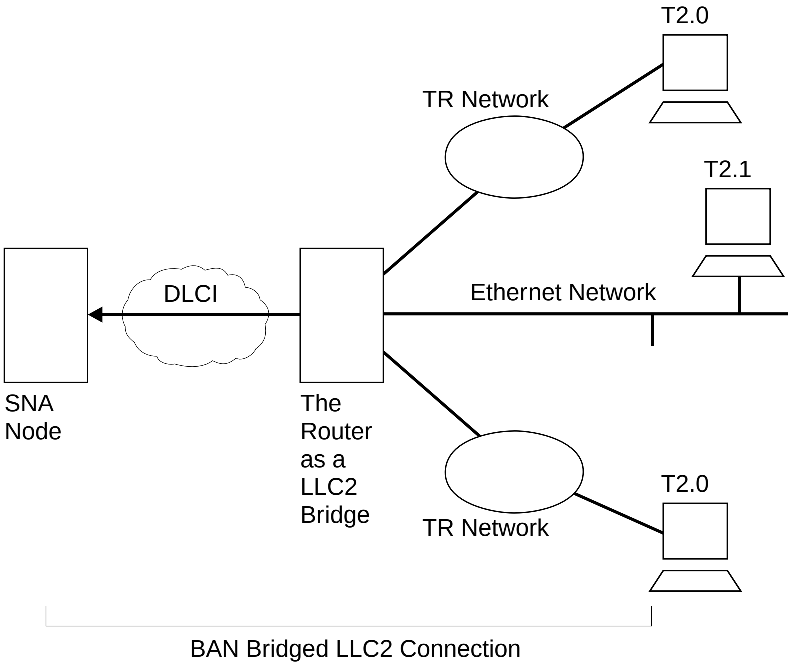


Figure 4-2. BAN Type 1: The Router as an LLC2 Bridge

In this case, the router acts as a bridge between the SNA node and the end stations. DLSw does not terminate LLC2 sessions at the router, as does BAN Type 2. End-station frames can be Token-Ring, SDLC (not pictured), or Ethernet format, provided the bridge is configured to support that type of frame.

Figure 4-3 on page 4-4 shows a BAN Type 2 (Virtual BAN DLSw) connection. In this figure, notice that the DLSw router does not function as a bridge. The router terminates the LLC2 traffic received from attached end stations. At the same time, the router establishes a new LLC2 connection to the SNA node over the frame-relay network. Thus, though two LLC2 connections exist within the transaction, the break between them is transparent both to the SNA node and the end stations. The result is a virtual LLC2 connection between the SNA node and the end stations.

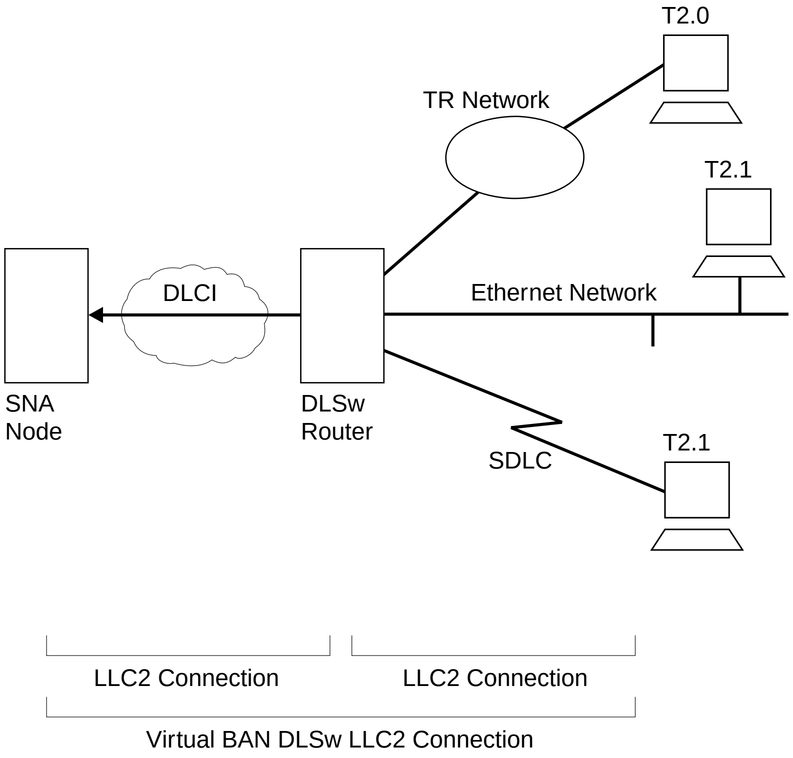


Figure 4-3. BAN Type 2: Local DLSw Conversion

The SDLC session is terminated in the router, and a separate LLC2 session exists between the router and the SNA node. The SDLC station appears to the SNA node as a Frame-Relay attached station.

Remote DLSw is supported for both types of BAN. Either BAN Type 1 or Type 2 connections can be used by routers functioning as DLSw partners to connect Type 2.0 or 2.1 end stations to an SNA node.

Which Method Should You Use?

Straight bridging of frames (BAN Type 1) is generally preferred because it provides fast delivery of data with minimal network overhead. However, there are exceptions. If usage on a DLCI is too high, session timeouts might occur in a bridged configuration. Conversely, session timeouts rarely occur in a DLSw configuration (BAN Type 2) since this type of configuration terminates and then recreates LLC2 sessions at the local (DLSw) router.

Using the BAN Feature

When you are configuring BAN, the system prompts you for information. Often, the system provides default values, which you accept by pressing **Return**.

To use the BAN feature, you must:

1. Configure the router for frame relay (FR)
2. Configure the router for Adaptive Source Route Bridging (ASRT)
3. Configure the router for BAN
4. Open service access points (SAPs) on the frame-relay and LAN Interfaces.

These steps are documented in the example that follows. The example assumes that you are setting up a single DLCI to carry BAN traffic. Depending on your circumstances and needs, you may want to set up multiple DLCIs for redundancy or for increased total bandwidth to the IBM environment. In this case, the BAN DLCI MAC address of the 2216 must be identical to the BAN DLCI MAC address of the ISDN backup 2216. Also, the value of the internal bridge segment of the 2216 must be different from the value of the internal bridge segment of the backup 2216. See “Setting up Multiple DLCIs” on page 4-8 for more information.

Step 1: Configure the 2216 for Frame Relay

To access the frame-relay configuration prompt, type **network** *interface#* at the `Config>` prompt as shown in the following example. (*Interface#* is the number of the frame-relay interface.)

```
Config>network 2
Frame Relay user configuration
FR Config>
```

At the `FR Config>` prompt, add a permanent circuit as shown in the following example. The router will prompt you for:

- The circuit number. This is the DLCI number.
- A committed information rate.

```
FR Config>add permanent
Circuit number  [16]? 20
Committed Information Rate in bps  [64000]?
Committed Burst Size(Bc) in bits (64000)?
Excess Burst Size (Be) in bits(0)?
Assign circuit name []? 20-ncp10
Is circuit required for interface operation [N]?
FR Config>
```

The DLCI you create becomes the PVC that connects the 2216 and the SNA node when BAN is used. The next step consists of configuring this PVC as a bridge port.

Note: If you want to set up multiple BAN DLCIs connected to the same or different SNA nodes, you must configure frame relay separately for each DLCI. See “Setting up Multiple DLCIs” on page 4-8 for more information.

Step 2: Configure the Router for Adaptive Source Route Bridging

Next, you must configure the PVC as a bridge port. To do this, use the **protocol** command at the `Config>` prompt as shown:

```
Config>protocol asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>
```

At the `ASRT Config>` prompt, add a port as shown. The router will prompt you for an interface number. The number you assign will be the FR interface number on the bridge. You will be prompted for a port number and a circuit number. The circuit number same as the name used when configuring the router for bridging over frame relay in Step 1.

```
ASRT config>add port
Interface Number [0]? 2
Port Number [5]?
Assign circuit number [16]? 20
ASRT config>
```

Next, enable source routing and define source-routing segment numbers for the frame-relay port:

```
ASRT config>enable source routing
Port Number [3]? 5
Segment Number for the port in hex (1 - FFF) [1]? 456
Bridge Number in hex (1-9, A-F) [1]? 789
ASRT config>
```

Last, disable transparent bridging on the bridge port as shown:

```
ASRT config>disable transparent bridging
Port Number [3]? 5
ASRT config>
```

The next step consists of configuring the router for BAN.

Step 3: Configure the Router for BAN

You must configure the router for BAN from the ASRT config> prompt. The addition of a BAN port on the router will not be verified until you restart the router. Note that, as in steps 1 and 2, bridge port 5 is the port used throughout this step.

```
Config>protocol asrt
ASRT config>ban
BAN (Boundary Access Node) configuration
BAN config>
```

At the BAN config> prompt, add the port number (5) on which you want to enable the BAN feature. You will be prompted to enter a BAN DLCI MAC address and the Boundary Node Identifier address as shown:

```
BAN config>add 5
Enter the BAN DLCI MAC Address []? 400000000001
Enter the Boundary Node Identifier MAC Address [4FFF00000000]?
```

In this example, 400000000001 is the MAC address of the DLCI. This is the address to which attached end stations will send data. (See Figure 4-2 on page 4-3 and Figure 4-3 on page 4-4). The other address, 4FFF00000000, is the default boundary node identifier address. To accept it, press **Enter**.

Note: The boundary node identifier corresponds to the destination MAC address placed in the bridged 802.5 frames sent from the 2216 to the SNA node. The default of 4FFF00000000 matches the default used by the IBM Network Control Program (NCP). The NCP address is specified in the NCP definition by the LOCADD keyword of the LINE statement that defines the physical Frame Relay port. For other SNA nodes that support bridged 802.5 frames over frame relay, the boundary node identifier must be set to the MAC address that the SNA node has configured for this virtual circuit.

Specifying the BAN Connection Type: The next prompt asks you to specify which type of BAN connection you want to add: bridged or DLSw terminated. These two methods are described in preceding sections as BAN Type 1 and BAN Type 2. Type 1, straight bridging, is the default. You should accept the default unless you want inbound traffic to be terminated at the router.

After you enter **b** or **t**, the router informs you that the BAN port has been added.

```
Do you want the traffic bridged (b) or DLSw terminated (t) (b/t) [b]?
BAN port record added.
```

Step 4: Open Service Access Points (SAPs)

To use the BAN feature, you must open the service access points (SAPs) associated with the FR interface and the LAN interface. If you fail to open these SAPs, you will not be able to use BAN. Failure to open all SAPs is often the cause of configuration problems.

Open the SAPs from the `DLSw config>` prompt as shown in this example:

```
DLSw config>open
Interface # [0]?
Enter SAP in hex (range 0-ff) [0]? 4
DLSw config>
```

Issuing the **open** command for interface 0 opens the SAP on the LAN interface. Issue the same command to open the SAP on the FR interface. Note that in each case, you enter the number **4** to open a SAP.

```
DLSw config>open
Interface # [2]?    [open on the FR interface]
Enter SAP in hex (range 0-ff) [0]? 4
DLSw config>
```

Using Multiple DLCIs for BAN Traffic

While one DLCI is usually sufficient to handle BAN traffic to and from the IBM environment, setting up two or more DLCIs may prove useful in some circumstances.

Scenario 1: Setting up a Fault-Tolerant BAN Connection

Redundant connections to multiple SNA nodes protect against a single SNA node failure. In addition, sharing BAN traffic among several DLCIs reduces the chance of one SNA node becoming overloaded. In a redundant DLCI configuration, PU Type 2.0 and 2.1 end stations can pass BAN traffic to different SNA nodes, as shown in Figure 4-4.

Note: Each DLCI is configured on a separate FR ASRT bridge port with the same DLCI MAC address.

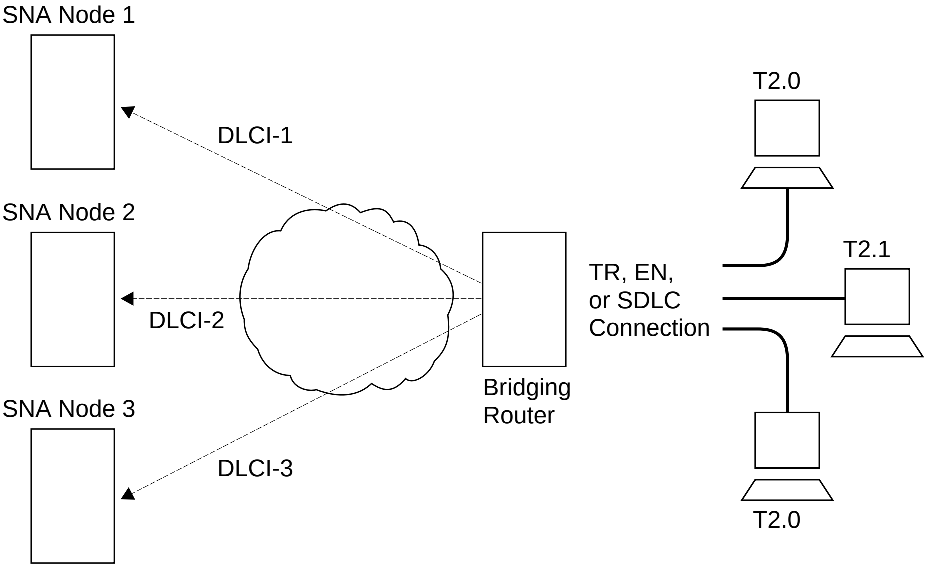


Figure 4-4. BAN Configuration with Multiple DLCIs to Different SNA Nodes

Scenario 2: Increasing Bandwidth to the IBM Environment

Multiple connections to the same SNA node increase the total bandwidth available for communicating with the IBM environment. This reduces the possibility of congestion on a single DLCI.

You may want to set up two or more DLCIs if you have a large amount of BAN traffic and another FR connection at your disposal. A second DLCI can provide greater total bandwidth to the SNA node, and protect you against unexpected failures.

Setting up Multiple DLCIs

Setting up multiple DLCIs is simple, particularly if you do this during the initial BAN configuration. When setting up multiple connections, remember that each frame-relay DLCI corresponds to a specific SNA node in the IBM environment. To pass BAN frames to that SNA node, you must specify the correct circuit number when establishing the Frame Relay connection. Your frame relay provider can give you the circuit number for each of your connections.

To set up DLCI connections to different SNA nodes (“Scenario 1: Setting up a Fault-Tolerant BAN Connection” on page 4-7), you must:

1. Take one of the following actions:
   - **In the ASRT configuration**, add a bridge port for that DLCI.
   - **In the frame-relay configuration**, define another frame-relay DLCI on a second bridge port.
2. Configure the bridge port for BAN, as shown in “Step 3: Configure the Router for BAN” on page 4-6.

To set up a second DLCI connection to the same SNA node (see “Scenario 2: Increasing Bandwidth to the IBM Environment”) follow the same steps. In “Scenario 2: Increasing Bandwidth to the IBM Environment,” the circuit number provided for the second frame-relay port will differ from the first. However, each circuit number identifies a different DLCI and a distinct path to the IBM environment.

Checking the BAN Configuration

When you restart the router, the router will validate that the BAN bridge port is a frame-relay bridge port with source-routing behavior. You should check the BAN configuration with the list command as shown here:

```
BAN config>list

bridge    BAN                  Boundary                 bridged or
port      DLCI MAC Address     Node Identifier          DLSw terminated
------    ----------------     ---------------          ---------------
5         40:00:00:00:00:01    4F:FF:00:00:00:00        bridged

BAN config>
```

As this example shows, the **list** command displays each aspect of the BAN configuration, giving the bridge port (5 in this case), the MAC address of the DLCI and the boundary node identifier for the SNA node, and whether the port is bridged or DLSw terminated.

To verify that BAN has initialized properly on startup, you can use GWCON as follows:

```
+ protocol asrt
ASRT>ban
BAN (Boundary Access Node) console

BAN>list
bridge  BAN                Boundary           bridged or
port    DLCI MAC Address   Node Identifier    DLSw terminated   Status
----    ----------------   ---------------    ---------------   -------
5       40:00:00:00:00:01  4F:FF:00:00:00:00  bridged           Init Fail

BAN>
```

GWCON provides three status messages:

- A status of `Init Fail` indicates that a configuration problem exists.
- A status of `Down` indicates that the DLCI is not running.
- A status of `Up` indicates that the frame-relay DLCI is up and running as intended.

If you receive a status other than `Up`, you should check the router's ELS messages to diagnose the problem. "Enabling Event Logging System (ELS) Messages for BAN" explains how to enable ELS messages.

Enabling Event Logging System (ELS) Messages for BAN

After initial BAN configuration and restart, it is a good idea to enable ELS messages to see whether the configuration is working as planned. You can enable BAN-specific messages from the `Config> prompt` as shown:

```
Config>ev
Event Logging System user configuration
ELS config>display subsystem ban all
ELS config>
```

Entering this command displays all BAN subsystem messages. This will cause ELS to notify you of all BAN-related behavior. After running BAN for a while, you may want to turn off some messages. You can turn off specific ELS BAN messages by using the **nodisplay** command and the specific message number. This example illustrates how to turn off the ban.9 message:

```
ELS config>nodisplay event ban.9
```

For a list and explanation of all BAN-related messages, refer to the Event Logging System Messages Guide.

Chapter 5. Basic Bridging Configurations

This chapter describes how to create basic configurations for the Adaptive Source Routing Transparent (ASRT) Bridge using the ASRT configuration commands. The chapter includes the following sections:

- “Accessing the ASRT Configuration Environment”
- “Basic Bridging Configuration Procedures”

If you need more information about the ASRT bridge configuration commands, refer to the Chapter 7, “Configuring Bridging.”

For an introduction to modification of ASRT bridging, see “NetBIOS Name and Byte Filters” on page 3-4.

For examples of setting up NetBIOS filtering, see “NetBIOS Host Name and Byte Filtering Configuration Procedures” on page 9-15.

Accessing the ASRT Configuration Environment

For information on how to access the ASRT configuration environment, see “Getting Started” in the *Software User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

Basic Bridging Configuration Procedures

The ASRT bridge allows you to perform basic bridging configurations using as few commands as possible. For example, using the **enable bridge** command begins this process by letting all properly configured devices participate in transparent bridging. In addition, all default values for the spanning tree algorithm are enabled.

Bridging function beyond transparent bridging is then enabled on a “per port” basis. When source routing is enabled, user input such as segment number, bridge number, and so on, is still required and must be entered beyond the basic commands that are explained.

Bridging Interfaces

The interfaces over which bridging is supported include combinations of one or more of the following:

- Ethernet
- Token-Ring
- Serial Line

The Ethernet interfaces typically support transparent bridging while token-ring interfaces can support source routing and transparent bridging.

The serial line interface provides point-to-point connectivity for transparent and source routing traffic. It is important to note that a bridge configuration over a serial line should be consistent at both end points. This means that both end points should be configured as follows:

- Transparent-to-transparent
- Source routing-to-source routing
- Source routing/transparent-to-source routing/transparent

It is best if the serial line is configured for both bridging methods if mixed bridging is desired. Another suggested guideline is to make sure that bridging routers are consistent in their bridging method or in their routing of particular protocols.

The information immediately following outlines the initial steps required to enable the bridging options offered by the ASRT bridge. Details on making further configuration changes will be covered in the command sections of this chapter. After completing these tasks, you must restart the router for the new configuration to take effect.

Enabling the Transparent Bridge

Use the following commands to enable transparent bridging:

- **Enable bridge** to enable transparent bridging on all Local Area Network (LAN) interfaces. Wide Area Network (WAN) interfaces (such as serial lines) can be included by using the **add port** command.
- **Disable transparent** *port#* to exclude specified token-ring interfaces from participating in transparent bridging. Repeat the command for all interfaces you want excluded from the transparent bridging configuration.

Enabling the Source Routing Bridge

Use the following commands to enable source-routing bridging:

- **Enable bridge** to enable bridging on all local area network interfaces. WAN interfaces (for example, serial lines) can be included by using the **add port** command.
- **Disable transparent** *port#* to disable transparent bridging on all ports.
- **Enable source-routing** *port# segment# [bridge#]* to enable source routing for given ports. When source routing is enabled on more than two ports, an additional segment number is required to assign an internal virtual segment needed for 1:N SRB configurations.

If source routing is the only feature desired, transparent bridging on the interfaces should be disabled.

Note: You should be careful to ***not*** include interfaces that traditionally do not support source routing. For example, if transparent bridging is disabled and source routing is enabled on an Ethernet port, the bridging facility is disabled for this port.

Enabling the SR-TB Bridge

Use the following commands to enable SR-TB bridging:

- **Enable bridge** to enable bridging on all local area network interfaces. WAN interfaces (for example, serial lines) can be included by using the **add port** command.
- **Disable transparent** *port#* to disable transparent bridging on all underlying source routing interfaces.
- **Enable source routing bridge** *port# segment# [bridge#]* to enable source routing for given ports. When source routing is enabled on more than two ports, an additional segment number is required to assign an internal virtual segment needed for 1:N SRB configurations.
- **Enable sr-tb-conversion** *segment#* to enable conversion of source-routed frames to transparent frames and vice versa. You are also required to assign a domain segment number and a domain MTU size to represent the entire transparent (Ethernet) bridging domain.

After completing any of the procedures just described, it is advised that you use the **list bridge** command to display the current bridge configuration. This lets you verify and check your configuration.

For more information on all of the commands just mentioned, refer to Chapter 7, “Configuring Bridging.”

Chapter 6. Overview of Routing Over ATM

Note: See the glossary for definitions of the acronyms and terms used in this chapter. This chapter describes the routing and bridging functions of the router.

Overview of Routing

The routing overview presented in this section is short because the relationships between LAN Emulation (LE), Classical IP (CIP), and the supported routing protocols are simple. The router supports IP and IPX routing as illustrated in Figures 6-1 and 6-2.

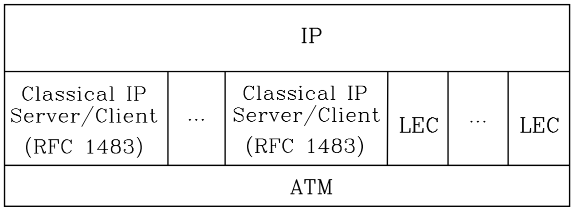

IP
Classical IP Server/Client (RFC 1483)
...
Classical IP Server/Client (RFC 1483)
LEC
...
LEC
ATM

Figure 6-1. IP Routing in the router

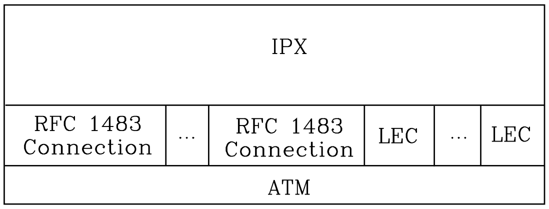


Figure 6-2. IPX Routing in the router

IP routing is supported between arbitrary combinations of Classical IP (CIP) and LAN emulation (LE) subnets, whereas IPX routing is supported over emulated LAN interfaces and RFC 1483[1] connections to other routers. These protocols treat emulated interfaces implemented by LAN emulation (LE) clients just like real Ethernet and Token-Ring interfaces. When an LE client is created, it is assigned a unique interface number.

All other routing protocols, for example, Appletalk 2 and DECnet, are supported only in LAN Emulation Mode.

Overview of Bridging

The 2216 supports bridging over emulated Ethernet and token-ring interfaces as illustrated in Figure 6-3 on page 6-3. The operational characteristics of bridging over these emulated interfaces are identical to those over legacy LAN interfaces. Emulated interfaces have interface numbers and bridge port numbers. Each bridge port has a particular behavior, and the bridge has an overall behavior.

[1] J. Heinanen, “Multiprotocol Encapsulation over ATM Adaptation Layer 5,” RFC 1483, Telecom Finland, July, 1993.

Three port-level bridging modes are available: transparent bridging (TB), source route (SR) bridging, or source route-transparent (SRT) bridging, which supports both TB and SR simultaneously. Emulated Ethernet ports support only TB. Emulated token-ring ports support all three modes.

Bridging Behaviors

There are six bridging behaviors: Pure transparent bridging (TB), pure source route bridging (SR), SR and TB, SR-TB, source route transparent (SRT), and adaptive source route transparent (ASRT). For each bridging behavior, LAN destination addresses are resolved via LE_ARP_REQUESTS as described in the following paragraphs.

Pure TB

Pure TB behavior is activated when all bridge ports are in TB mode. In Pure TB mode, the router acts only as a transparent bridge and the IEEE 802.1d spanning tree algorithm is used.

Pure SR

Pure SR behavior is activated when all bridge ports are in SR mode. In Pure SR mode, the router acts only as a source-route bridge and the IBM source-route bridging spanning tree algorithm is used.

SR and TB

SR and TB behavior is activated when at least one bridge port is in SR mode, at least one bridge port is in TB mode, no bridge ports are in SRT mode, and SR-TB translation is disabled. In SR and TB mode, the router acts as both a transparent bridge and a source-route bridge simultaneously, but the two types of bridges do not work together and are thus isolated from each other. The IEEE 802.1d and IBM source-route spanning tree algorithms are used independently in SR and TB mode.

SR-TB

The SR-TB bridging mode differs from SR and TB in that SR-TB translation is enabled. In SR-TB mode, the bridges are no longer independent and frames are translated between the two bridges. SR-TB translation is only supported for protocols that use IEEE 802.2 Logical Link Control (such as SNA and NetBIOS).

SRT

SRT behavior is activated when at least one bridge port is in SRT mode and SR-TB translation is not enabled. In SRT mode, the bridges are independent, as in SR and TB mode, but only the IEEE 802.1d spanning tree algorithm is used.

ASRT

The ASRT bridging mode differs from SRT in that SR-TB translation is enabled in ASRT but not in SRT.

Resolving LAN Destination Addresses via LE_ARP_REQUESTS

Given the port behaviors, only one additional configuration parameter, for enabling or disabling SR-TB translation, is required to determine the server-level bridging behavior.

When an LE client is acting as a bridge port, it joins the ELAN as a proxy and registers its MAC address, regardless of the bridging mode. If the port-level bridging mode is SR or SRT, a route descriptor is also registered with the LES. The LE client will always answer LE_ARP_REQUESTs for LAN Destinations that it registered. Additionally, if transparent bridging is enabled on the port, the LE client will respond to LE_ARP_REQUESTs when the target MAC address is in the TB database.

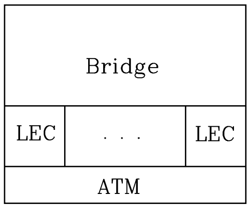


Figure 6-3. Bridging Over the Emulated LAN Interface

Overview of RFC 1483 Support

RFC 1483 (Multiprotocol Encapsulation over ATM Adaptation Layer 5) provides the details about the encapsulation of bridged and router frames. Routing of IP and IPX traffic is supported. The software also provides the full range of bridging capabilities, allowing bridged traffic to be transmitted natively over ATM.

RFC 1483 specifies LLC/SNAP encapsulation for carrying multiprocol traffic over ATM. A LLC value of 0xAA-AA-03 is specified to indicate the presence of a SNAP header. The OUI portion of the SNAP header is 0x00-00-00 for routed protocols, and 0x00-80-C2 for bridged protocols.

Overview of RFC 1483 Support for Routing

Classical IP uses the LLC/SNAP format for routed protocols defined in RFC 1483. The router also supports connections to IPX routers that use LLC/SNAP encapsulation. This IPX support is modeled after the Classical IP approach.

RFC 1483 Support for IPX Routing

IPX routers use routing information protocol (RIP) and service advertising protocol (SAP) to propagate routing and server information tables. On LANs or emulated LANs, these protocols use broadcast frames to propagate information to interested parties. The router will also propagate the routing and server information to and from all RFC 1483 connections with other IPX routers.

The 2216, like other routers that support RFC 1483 LLC/SNAP encapsulation on ATM, can be interconnected in full or partial meshes using manually configured RFC 1483 connections.

In a *fully meshed* network, every router has a direct connection to every other router. In a *partially meshed* network, not every router has a direct connection to every other router; however, there exists enough connectivity for any router to reach any other router, directly or through another router. In the partially meshed network, some routers must perform intermediate routing. An intermediate router provides connectivity between routers that are not directly connected to one another.

Both permanent virtual circuits (PVCs) and configured switched virtual circuits (SVCs) are supported. However, virtual channel connections (VCCs) to IPX routers must be dedicated to IPX; they cannot be shared with other protocols, such as IP.

The router supports a single IPX network per ATM interface. This statement implies that there is a single ATM ARP client per interface for IPX, which must be explicitly configured. Therefore, all interconnected routers on the ATM interface must be part of the same IPX network.

IPX ATM addresses must be unique among all components using RFC 1483 encapsulation, including Classical IP components. The end system identifier (ESI) and the selector portions of IPX ATM addresses are configured in the same manner as Classical IP ATM addresses. When the router does not initiate the SVC, then at least the selector should be explicitly specified in order to provide a fixed address that can be configured at the calling router.

IPX protocol addresses have two parts: a 4-byte network number and a 6-byte host number (or host ID). Network numbers must be unique within IPX routing domains, and host numbers must be unique within a given network. The router sets the IPX host number to the ESI component of the associated ATM address. Whenever you do not explicitly configure the ESI, it defaults to the MAC address that is burned into the ATM interface hardware.

Destination IPX host numbers can be specified during VCC configuration or learned dynamically using InATMARP. You must manually configure the IPX host numbers of destination routers that do not support InATMARP. The router also periodically uses InATMARP to refresh its knowledge of the partner router’s IPX host number.

Routers that are interconnected in a partial mesh and are providing intermediate routing between routers on the same ATM interface should disable IPX split-horizon on the ATM interface. Doing this ensures that RIP and SAP properly inform the interconnected routers of all available routes and services. Routers that are interconnected in a full mesh need not disable split-horizon.

The router implementation of RFC 1483 support for IPX routing requires minimal configuration. The IPX network number and the IPX host number (IPX ATM ARP client) are the only pieces of information that are required. If you wish to open a connection to a remote IPX router, you must additionally configure the desired virtual channel connections (VCCs). Although the combination of RFC 1483 encapsulation and InATMARP has not been has not yet been standardized, the combination is specified for IPX over Frame Relay in RFC 1490.[2]

[2] T. Bradley, C. Brown, and A. Malis, “Multiprotocol Interconnect Over Frame Relay,” RFC 1490, Wellfleet Communications Inc. and Ascom Timeplex Inc., July 1993.

Chapter 7. Configuring Bridging

This chapter describes how to configure the Adaptive Source Routing Transparent (ASRT) Bridge protocol and how to use the ASRT configuration commands. The chapter includes the following sections:

- “Accessing the ASRT Configuration Environment”
- “ASRT Configuration Commands”
- “Tunnel Configuration Commands” on page 7-34
- “BAN” on page 7-12
- “Frame-Relay Commands” on page 7-39

Accessing the ASRT Configuration Environment

To access the ASRT configuration environment, enter the **protocol asrt** command at the `Config>` prompt:

```
Config>protocol asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>
```

ASRT Configuration Commands

This section summarizes and then explains the ASRT configuration commands. The ASRT configuration commands allow you to specify network parameters for the ASRT bridge and its network interfaces. These commands also allow you to enable and configure the bridge IP Tunnel, NetBIOS, and ATM interface features.

The router must be restarted for the new configuration to take effect.

Enter the ASRT configuration commands at the `ASRT config>` prompt. Access the commands as follows:

- Enter configuration commands for IP tunnels at the `TNL config>` prompt. The tunnel prompt is a subset of the major ASRT commands and is accessed by entering the ASRT **tunnel** command explained later in this chapter.
- Enter configuration commands for NetBIOS at the `NetBIOS config>` prompt. The NetBIOS prompt is a subset of the major ASRT commands and is accessed by entering the ASRT **netbios** command explained later in this chapter.
- Enter configuration commands for NetBIOS Filtering at the `NetBIOS Filter config>` prompt. This prompt is a subset of the NetBIOS commands.
- Enter bridging configuration commands for ATM at the `ASRT config>` prompt.

Table 7-1 on page 7-2 shows the ASRT configuration commands.

Table 7-1. ASRT Configuration Command Summary

Command	Function
? (Help)	Lists all of the ASRT configuration commands, or lists the options associated with specific commands.
Add	Adds station address entries to the permanent database, specific address mapping, LAN/WAN ports, protocol filters, and a tunnel between end stations across an IP internetwork.
Ban	Allows access to the boundary access node (BAN) configuration prompt so that BAN configuration commands can be entered.
Change	Allows the user to change bridge and segment numbers.
Delete	Deletes station address entries, specific address mapping, LAN/WAN ports, protocol filters, and a tunnel between end stations across an IP internetwork.
Disable	Disables the following functions: • Bridging • Duplicate frames • Mapping between group and functional addresses • Propagation of Spanning Tree Explorer Frames • Source routing on a given port • Reception of spanning tree explorer frames over a tunnel • Conversion of source routed to transparent frames • Transparent (spanning tree) bridging function on a given port • Tunnel between bridges
Enable	Enables the following functions: • Bridging • Duplicate frames • Mapping between group and functional addresses • Propagation of Spanning Tree Explorer Frames • Source routing on a given port • Reception of spanning tree explorer frames over a tunnel • Conversion of source routed to transparent frames • Transparent (spanning tree) bridging function on a given port • Tunnel between bridges
List	Displays information about the complete bridge configuration or about selected configuration parameters.
NetBIOS	Displays the NetBIOS configuration prompt.
Set	Sets the following parameters: • Aging time for dynamic address entries • Bridge address • Maximum frame size for tunneling • Largest Frame (LF) bit encoding • Maximum frame size • Spanning tree protocol bridge and port parameters • Route Descriptor (RD) values • Filtering database size
Tunnel	Allows access to the tunnel configuration prompt so that tunnel configuration commands can be entered.
Exit	Exits the ASRT configuration process and returns to the CONFIG environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADD
BAN
CHANGE
DELETE
DISABLE
ENABLE
LIST
NETBIOS
SET
TUNNEL
EXIT
```

Example: **set ?**

```
AGE
BRIDGE
FILTERING
LF-BIT-INTERPRETATION
PORT
MAXIMUM-PACKET-SIZE
PROTOCOL BRIDGE
PROTOCOL PORT
PROTOCOL
ROUTE-DESCRIPTOR-LIMIT
```

Add

Use the **add** command to add the following information to your bridging configuration:

- Station address entries to the permanent database
- Specific address mapping for a given protocol
- LAN/WAN ports
- Protocol filters that selectively filter packets based on their protocol type
- IP tunnel between end-stations and across IP network segments

For the bridge's IP tunnel feature, the **add** command lets you create an IP tunnel between end-stations across an IP internetwork. This tunnel is counted as only one hop between the end stations no matter how complex the path through the IP internet.

Syntax: add address . . .
mapping . . .
port . . .
prot-filter . . .
tunnel . . .

address *addr-value*

Adds unique station address entries to the permanent database. These entries are copied into the filtering database as permanent entries when the bridge is restarted. The *addr-value* is the MAC address of the desired entry. It can be an individual address, multicast address, or broadcast address.

You are also given the option to specify the outgoing forwarding port map for each incoming port. Permanent database entries are not destroyed by the power off/on process and are immune to the aging settings. Permanent entries cannot be replaced by dynamic entries.

Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Default Value: none

Example: `add address`

```
Address (in 12-digit hex) []? 123456789013
    Exclude destination address from all ports?(Yes or [No]):
    Use same output port mapping for all input Ports?(Yes or [No]):
    Output port mapping:
       Input Port Number [1]?
       Bridge to all ports?(Yes or [No]):
       Bridge to  port 1  Yes or [No]:
       Bridge to  port 2  Yes or [No]:
       Bridge to  port 3  Yes or [No]:
       Bridge to  port 4  Yes or [No]:
       Bridge to  port 5  Yes or [No]:
       continue to another input port? (Yes or [No]): y
       Input Port Number [2]? 3
       Bridge to all ports?(Yes or [No]): y
       continue to another input port? (Yes or [No]): y
       Input Port Number [4]?
       Bridge to all ports?(Yes or [No]):
       Bridge to  port 1  Yes or [No]:
       Bridge to  port 2  Yes or [No]:
       Bridge to  port 3  Yes or [No]:
       Bridge to  port 4  Yes or [No]:
       Bridge to  port 5  Yes or [No]:
       continue to another input port? (Yes or [No]): n
    Source Address Filtering Applies? (Yes or No): y
    ASRT config>
```

Note: For any “Yes or No” question in the prompts, “No” is the default value. Press **Return** to accept the default value.

Exclude destination address ... This prompt lets you set destination address filtering for that entry. Answering “Yes” to the prompt causes filtering of any frames that contain this address as a destination address no matter which port it came from.

Use same output mapping... Answering “Yes” to this prompt lets you create one outgoing port map for all incoming ports rather than allowing for mapping to only specific ports. Answering “No” to this prompt causes further prompting (`Input Port Number [1]?`) to select each input port. From that specific input port prompt you can then create a unique port map for that input port.

Input Port 1, Port 2 Answering “No” to the previous prompt causes input port-by-input port prompting (`Input Port Number [1]?`) to select each input port and its associated outgoing bridge ports.

Bridge to all ports? Answering “Yes” to this prompt creates an outgoing port map which includes all ports. Thus, when a frame with this address as the destination address is received, it is forwarded to all outgoing forwarding ports except for the incoming port. The following are examples of how this is done according to the port map:

If a frame is received on *port 1* and the port map indicates 1 (for port 1), the frame is filtered.

If the same frame is received on *port 2* and the port map indicates 1 (for port 1), the frame is forwarded to port 1. If a frame is received on port 1 and the matching address entry's port map indicates 1, 2, or 3, the frame is forwarded to ports 2 and 3.

If the port map indicates no port (NONE/DAF), the frame is filtered. This is known as destination address filtering (DAF).

If no address entry is found to match the received frame, it is forwarded to all the forwarding ports except for the source port.

Bridge to Port 1, Port 2, etc. This prompt lets you associate an address entry with that specific bridge port. Entering “Y” (for yes) after the prompt maps the address to the specified port so that the port is included in that address entry's port map. Entering “N” skips address mapping for that port.

continue to another bridge port? This prompt lets you select the next input port to be configured.

Source address filtering This allows for port-specific address filtering. When SAF is applied (“yes” is entered at the prompt), frames received with source addresses that match address entries in the filtering database that have source address filtering enabled will be discarded. This mechanism allows a network manager to isolate an end station by prohibiting its traffic to be bridged.

The following sections present specific examples of how the **add address** command is used to manage address entries:

Enabling Destination Address Filtering For Entry

This example shows how to answer the command prompts to select destination address filtering for an entry:

```
ASRT config>add address 000000334455
Exclude destination address from all ports?(Yes or [No]): y
Source Address Filtering Applies? (Yes or [No]): y
ASRT config>
```

After adding the address entry, you can verify its status by using the **list range** command. The following example shows that no port map exists for that entry (in bold) and that destination address filtering (DAF) has been turned on.

```
ASRT config>list range
Start-Index [1]?
Stop-index [3]?
ADDRESS                 ENTRY TYPE      PORT MAP
=======                 ==========      ========
01-80-C2-00-00-00       REGISTERED      Input Port:  ALL PORTS
                                        Output ports:

00-00-00-22-33-44       PERMANENT       Input Port:  3
                                        Output ports: 1, 2
                                        Input Port:  4
                                        Output ports:  1, 2

00 00 00 33 44 55       PERMANENT       NONE/DAF
```

Output Port Map Created For Address Entry Having More Than One Input Port

This example shows how to answer the command prompts to create separate output port maps for an address entry that will have more than one input port.

```
ASRT config> add address 000000123456
Exclude destination address from all ports?(Yes or [No]):
Use same output port mapping for all input Ports?(Yes or [No]):
Input Port Number [1]? 1
Bridge to all ports ?(Yes or [No]):
Bridge to port 1 - Yes or [No]: y
Bridge to port 2 - Yes or [No]: y
Bridge to port 3 - Yes or [No]:
continue to another input port ? (Yes or [No]): y
Input Port Number [2]?
Bridge to all Ports?(Yes or [No]):
Bridge to Port 1 - Yes or [No]:
Bridge to port 2 - Yes or [No]:
Bridge to port 3 - Yes or [No]: y
continue to another input port ? (Yes or [No]):
Source Address Filtering Applies? (Yes or [No]):
ASRT config>
```

After adding the address entry, you can verify its status by using the **list range** command. The following example shows an entry (in bold) that has ports 1 and 2 as input ports and has separate port maps for both input ports. Source address filtering (SAF) has also been enabled.

```
ASRT config> list range
Start-Index [1]?
Stop-index [3]?
ADDRESS                 ENTRY TYPE      PORT MAP
=======                 ==========      ========
01-80-C2-00-00-00       REGISTERED      Input Port:  ALL PORTS
                                        Output ports:

01-80-C2-00-00-01       RESERVED        NONE/DAF

00-00-00-12-34-56       PERM/SAF        Input Port:  1
                                             Output ports: 1, 2
                                             Input Port:  2
                                             Output ports: 3
```

Single Output Port Map Created All Incoming Ports Associated With Address Entry

This example shows how to answer the command prompts to create a single output port map for all incoming ports associated with an address entry.

```
ASRT config> add address 000000556677
Exclude destination address from all ports?(Yes or [No]):
Use same output port mapping for all input Ports?(Yes or [No]): y
   Bridge to all ports?(Yes or [No]): n
   Bridge to  port 1 - Yes or [No]: y
   Bridge to  port 2 - Yes or [No]: y
   Bridge to  port 3 - Yes or [No]:
Source Address Filtering Applies? (Yes or [No]): y
ASRT config>
```

After adding the address entry, you can verify its status by using the **list range** command. The example below shows an entry (in bold) that has a single port map for all incoming ports. Source address filtering (SAF) has also been enabled.

```
ASRT config> list range
Start-Index [1]?
Stop-index [3]?
ADDRESS                ENTRY TYPE      PORT MAP
=======                ==========      ========
01-80-C2-00-00-00      REGISTERED      Input Port:  ALL PORTS
                                       Output ports:

01-80-C2-00-00-01      RESERVED        NONE/DAF

00-00-00-55-66-77      PERM/SAF        Input Port:  ALL PORTS
                                       Output ports:  1, 2
```

mapping *dlh-type type-field ga-address fa-address*

Adds specific functional address to group address mapping for a given protocol identifier. The address mapping is converted only on destination addresses crossing Token Ring to Ethernet or vice versa.

Note: For every Ether-type mapped value, the corresponding SNAP-type value should be added. This is necessary for bidirectional mapping.

dlh-type (data-link-header type) is a choice for DSAP, Ether-type, or SNAP.

type-field Protocol type field.

Destination Service Access Point (DSAP) protocol type is entered in a range of 1 – FE (hexadecimal).

DSAP Valid Values: X'1' to X'FE'

Common values are:

Protocol - SAP (hexadecimal value)

- Banyan SAP - BC (used only for 802.5)
- Novell IPX SAP - E0 (used only for 802.5)
- NetBIOS SAP - F0
- ISO Connectionless Internet - FE

DSAP Default Value: 1

Ethernet (Ether) protocol type is entered in a range of 5DD–FFFF (hexadecimal).

Ethernet Valid Values: X'5DD' to X'FFFF'

Protocol - Ethernet type (hex value)

- IP - 0800
- ARP - 0806
- CHAOS - 0804
- Maintenance Packet Type - 7030
- DECnet MOP Dump/Load - 6000
- DECnet MOP Remote Console - 6002
- DECnet- 6003
- DEC LAT - 6004
- DEC LAVC - 6007
- XNS - 0600
- Apollo Domain - 8019 (Ethernet)
- Novell NetWare IPX - 8137 (Ethernet)
- AppleTalk Phase 1 - 809B
- Apple ARP Phase 1 - 80F3
- Loopback assistance - 9000

Ethernet Default Value: 1

Subnetwork Access Protocol (SNAP) protocol type is entered in 10-digit hexadecimal format.

SNAP Valid Values: X'00 0000 0000' to X'FF FFFF FFFF'

Common values are:

• AppleTalk Phase 2	08-00-07-80-9B
• Apple ARP Phase 2	00-00-00-80-F3

SNAP Default Value: 00 0000 0800

ga-address 6-byte (12-digit hexadecimal) group/multicast address.

Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Default Value: none

fa-address Functional address in noncanonical format. Functional addresses are locally administered group addresses. These are most commonly used in token-ring networks.

Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Default Value: none

Example:

```
ASRT config> add mapping dsap
```

```
Protocol Type in hex (1 - FE) [1]?
Group-Address (in 12-digit hex) [ ]?
Functional address (in noncanonical format) [ ]?
```

Example: `ASRT config>` **`add mapping ether`**

```
Protocol Type in hex (5DD - FFFF) [0800]?
Group-Address (in 12-digit hex) []?
Functional address (in noncanonical format) [ ]?
```

Example: `ASRT config>` **`add mapping snap`**

```
Address (in 10-digit hex) [0000000800]?
Group-Address (in 12-digit hex) []?
Functional address (in noncanonical format) [ ]?
```

port *interface* *port-num*

Adds a LAN/WAN port and its associated DLCI to the bridging configuration. This command associates a port number with the interface number and enables that port's participation in transparent bridging.

Port Number Valid Values: 1 to 254

Port Number Default Value: none

Example 1: add a port

```
ASRT config> add port
Interface Number [0]?
Port Number [5]?
```

prot-filter snap ether dsap

Allows the bridge to be configured so that it can selectively filter packets based on their protocol type. Filters can be applied to all ports or only selected ports.

This parameter specifies protocol identifiers for which the received frames of that specific protocol are discarded exclusively without applying bridge logic. ARP packets for this protocol type will also be discarded. The protocol filter is applied only on the received packets. The protocol filters available include the following:

SNAP Packets Subnetwork Access Protocol with protocol type entered in 10-digit hexadecimal format.

Ether Packets Ethernet Type with the protocol type entered in a range of 5DD–FFFF (hexadecimal).

DSAP Packets Destination Service Access Point protocol with the protocol type entered in a range of 0–FE (hexadecimal).

The routing protocols that are enabled in the router (that is, the ones that are displayed by the configuration command in GWCON) cannot be added for filtering. Common protocol filters and their respective values are as follows.

DSAP Types

Protocol	SAP (hexadecimal value)
Banyan SAP	BC (used only for 802.5)
Novell IPX SAP	E0 (used only for 802.5)
NetBIOS SAP	F0
ISO Connectionless Internet	FE

SNAP Protocol Identifiers

Protocol	*SNAP OUI/IP (10-digit)*
AppleTalk Phase 2	08-00-07-80-9B
Apple ARP Phase 2	00-00-00-80-F3

Ethernet Types

Protocol	*Ethernet type (hex value)*
IP	0800
ARP	0806
CHAOS	0804
Maintenance Packet Type	7030
DECnet MOP Dump/Load	6000
DECnet MOP Remote Console	6002
DECnet	6003
DEC LAT	6004
DEC LAVC	6007
XNS	0600
Apollo Domain	8019 (Ethernet)
Novell NetWare IPX	8137 (Ethernet)
Apple ARP Phase 1	80F3
Loopback assistance	9000

Example: `ASRT config>` **`add prot-filter dsap`** (used for DSAP packets)

```
Protocol Type in hex (0 - FE) [1]?
   Filter packets arriving on all ports?(Yes or [No]):
   Filter packets arriving on  port 1 - Yes or [No]:
   Filter packets arriving on  port 2 - Yes or [No]:
   Filter packets arriving on  port 3 - Yes or [No]:
Port assignment Required, filter not added
ASRT config>
```

Example: `ASRT config>` **`add prot-filter ether`** (used for Ethernet packets)

```
Protocol Type in hex (5DD - FFFF) [0800]?
   Filter packets arriving on all ports?(Yes or [No]):
   Filter packets arriving on  port 1 - Yes or [No]:
   Filter packets arriving on  port 2 - Yes or [No]:
   Filter packets arriving on  port 3 - Yes or [No]:
Port assignment Required, filter not added
ASRT config>
```

Example: **`add prot-filter snap`** (used for SNAP packets)

```
Address (in 10-digit hex) [0000000800]?
Protocol Type in hex (5DD - FFFF) [0800]?
   Filter packets arriving on all ports?(Yes or [No]):
   Filter packets arriving on  port 1 - Yes or [No]:
   Filter packets arriving on  port 2 - Yes or [No]:
   Filter packets arriving on  port 3 - Yes or [No]:
Port assignment Required, filter not added
ASRT config>
```

tunnel *port#*

Creates the user-defined IP tunnel to a bridge port. This tunnel provides a passage for a bridged frame through an IP internetwork. This tunnel is counted as only one hop between the bridges no matter how complex the path through the IP internet. To use the tunnel feature, the IP forwarder must be enabled.

The tunnel bridge allows source route bridge domains or transparent bridge domains to communicate across an IP network.

To allow IBM LAN and terminal traffic to merge with non-IBM traffic (that is, Novell) across a single backbone, the Source Routing Bridge Tunnel and SDLC (Synchronous Data Link Control) Relay features of the bridging router software encapsulate IBM traffic within industry-standard TCP/IP packets. The bridging router then routes these packets using an IP path or *tunnel* through large IP internetworks. The benefit is increased functionality and network utilization as well as higher network availability and increased ease of use.

End-stations see the IP path (the tunnel) as a single hop, regardless of the network complexity. This helps overcome the usual 7-hop distance limit encountered in source routing configurations. It also lets you connect source-routing end-stations across non-source-routing media, such as Ethernet networks.

The bridging tunnel also overcomes several limitations of regular source routing including:

- Distance limitation of seven hops
- Large amounts of overhead that source routing causes in wide-area networks (WANs)
- Source-Routing's sensitivity to WAN faults and failures (if a path fails, all systems must restart their transmissions)

With the bridge tunnel feature enabled, the software encapsulates packets in TCP/IP packets. To the router, the packet looks like a TCP/IP packet. Once a frame is encapsulated in an IP envelope, the IP forwarder is responsible for selecting the appropriate network interface based on the destination IP address. This packet can be routed dynamically through large internetworks without degradation or network size restrictions. End-stations see this path, or tunnel, as a single hop regardless of the complexity of the internetwork.

The tunnel is transparent to the end stations. The bridging routers participating in tunneling treat the IP internet as one of the bridge segments. When the packet reaches the destination interface, the TCP/IP headers are automatically removed and the inner packet proceeds as a standard source routing packet.

Add Tunnel creates the user-defined IP tunnel to a bridge port. This tunnel is counted as only one hop between the bridges no matter how complex the path through the IP internet. To use the tunnel feature, the IP forwarder must be enabled.

Only one tunnel can be added. It is required that for the *Port Number*, you use one that is not used for any other LAN port. Internally, the interface number 255 is ascribed to mark that interface as connected as a "virtual" interface.

Transparent bridging is enabled on this port by default. Source routing can be enabled, however, by using the **Enable Source-Routing** option.

Example: **add tunnel 3**

```
Port Number   [1] ? 3
```

Port Number A unique port number that is not being used by the bridge.

BAN

Use the **ban** command to access the boundary access node (BAN) configuration prompt. BAN commands are entered at the BAN configuration prompt (`BAN config>`) See “BAN” for an explanation of each of these commands.

Syntax: ban

Example: **ban**

```
BAN (Boundary Access Mode) configuration
BAN config>
```

Change

Use the **change** command to change source routing bridge and segment numbers in the bridging configuration.

Syntax: change bridge . . .
segment . . .

bridge *new-bridge#*

Changes bridge numbers in the bridging configuration.

Example: **change bridge 3**

segment *old-segment# new-segment#*

Changes bridge numbers in the bridging configuration.

Example: **change segment 2 3**

Delete

Use the **delete** command to delete the following information from your bridging configuration:

- Station address entries to the permanent database
- Specific address mapping for a given protocol
- LAN/WAN ports
- Protocol filters that selectively filter packets based on their protocol type

For the IP tunnel feature, the **delete port** command with the corresponding port number for the tunnel removes the tunnel between bridges across an IP internetwork.

Syntax: delete address
mapping . . .
port . . .
prot-filter . . .

address *addr-value*

Deletes an address entry from the permanent database. The address is the MAC address of the desired entry. Enter the addr-value (in 12-digit hexadecimal format) of the entry to be deleted and press **Return**. Reserved multicast addresses cannot be deleted. If you attempt to delete an address entry that does not exist, you will receive the message

```
Record matching that address not found
```

Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Default Value: none

Example: `delete address`

mapping *dlh-type type-field ga-address*

Deletes specific address mapping for given protocol.

dlh-type	(data-link-header type) is a choice for DSAP, Ether-type, or SNAP.
type-field	Protocol type field.
	Destination service access point (DSAP) protocol type is entered in a range of 1 – FE (hexadecimal).
	Valid Values: X'1' to X'FE'
	Common values are:
	Protocol - SAP (hexadecimal value)
	Default Value: 1
	Ethernet (Ether) protocol type is entered in a range of 5DD–FFFF (hexadecimal).
	Valid Values: X'5DD' to X'FFFF'
	Default Value: 1
	Subnetwork Access Protocol (SNAP) protocol type is entered in 10-digit hexadecimal format.
	Valid Values: X'00 0000 0000' to X'FF FFFF FFFF'
	Common values are:
	Default Value: 00 0000 0800
ga-address	6-byte (12-digit hexadecimal) group/multicast address.
	Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'
	Default Value: none

Example: `delete mapping DSAP FE <group address>`

port *port#*

Removes a port from a bridging configuration. Because the **enable bridge** command by default configures all LAN devices to participate in bridging, this command allows you to customize which devices should or should not participate in the bridging. The port number value normally is one greater than the interface number.

This command followed by the IP tunnel port# removes an IP tunnel from a bridging configuration.

Example: `delete port 2`

prot-filter snap ether dsap

Deletes previously specified protocol identifiers used in filtering. You can delete filters for all ports or selected ports. These filters include the following:

SNAP Packets	Subnetwork Access Protocol with protocol type entered in 10-digit hexadecimal format.
Ether Packets	Ethernet Type with the protocol type entered in a range of 5DD – FFFF (hexadecimal).
DSAP Packets	Destination service access point protocol with the protocol type entered in a range of 0–FE (hexadecimal).

Example: `ASRT config>` **`delete prot-filter snap`** (used for SNAP packets)

```
Address (in 10-digit hex) [0000000800]?
   Delete filter on all ports?(Yes or [No]):
   Delete filter on  port 1 - Yes or [No]:
   Delete filter on  port 2 - Yes or [No]:
   Delete filter on  port 3 - Yes or [No]:
```

Example: `ASRT config>` **`delete prot-filter ether`** (used for Ethernet packets)

```
Protocol Type in hex (5DD - FFFF) [0800]?
   Delete filter on all ports?(Yes or [No]):
   Delete filter on  port 1 - Yes or [No]:
   Delete filter on  port 2 - Yes or [No]:
```

Example: `ASRT config>` **`delete prot-filter dsap`** (used for DSAP packets)

```
Protocol Type in hex (0 - FE) [1]?
   Delete filter on all ports?(Yes or [No]):
   Delete filter on  port 1 - Yes or [No]:
   Delete filter on  port 2 - Yes or [No]:
   Delete filter on  port 3 - Yes or [No]:
```

Disable

Use the **disable** command to disable the following bridge functions:

- Bridging function entirely
- Creation of duplicate frames for mixed bridging environments (network traffic management)
- Mapping between group address and functional address
- Propagation of Spanning Tree Explorer Frames
- Source routing on a given port
- Reception of spanning tree explorer frames over a tunnel
- Conversion of source routed frame to transparent frame and vice versa
- Transparent (spanning tree) bridging function on a given port

For the tunnel feature, the disable command disables a tunnel between end stations across an IP internetwork.

Syntax:

```
disable   bridge
          dls
          duplicate . . .
          ethertype-ibmrt-pc
          fa-ga-mapping
          ibm8209_Spanning_Tree
          spanning-tree-explorer  . . .
          source-routing  . . .
          sr-tb-conversion
          stp
          transparent  . . .
          tree
          ub-encapsulation
```

bridge

Disables bridging function entirely. This command does not remove previously configured bridging values, however.

Example: **`disable bridge`**

dls

Disables the operation of DLSw on the bridge. (The router running DLSw appears as a bridge to the end stations.) See Chapter 27, “Using and Configuring DLSw” on page 27-1 for more details.

Example: **`disable dls`**

duplicate *frame-type*

Disables the creation of duplicate frames present in mixed bridging environments. When the SR-TB bridging feature is enabled on an 802.5 interface (with source routing and transparent bridging enabled), there are inconsistencies created when bridging frames to an unknown (or multicast) destination. It is not known to the bridge whether the destination is behind a source routing (only) or transparent bridge.

To remedy this situation, the bridge sends out duplicates of these frames (by default). One frame has source routing fields present (a spanning tree explorer RIF) and the other is formatted for transparent bridging (no RIF is present). The **disable duplicate** command lets you eliminate this duplication by allowing you to disable the creation of one of these types of frames. The **disable duplicate** command will not allow you to disable simultaneously both types of frames.

Entering **STE** after the command tells the bridge to refrain from sending out spanning tree explorer frames created for the source routing environment. Entering **TSF** after the command tells the bridge to refrain from sending out transparent spanning frames for the transparent bridging environment. In both cases, it is a situation where normally both types of frames would be sent out. Disabling transparent bridging on the interface also disables the creation of transparent frames.

Example: **`disable duplicate TSF`**

```
Port Number [1]?
```

ethertype-ibmrt-pc

Disables translation of SNA frames to Ethernet Type 2 format as used by IBM RTs running OS/2 EE.

Example: **`disable ethertype-ibmrt-pc`**

```
Port Number [1]?
```

fa-ga-mapping

Disables group address-to-functional address (and vice versa) mapping. You might under certain circumstances want to disable the mapping between group address and functional address globally.

Example: **`disable fa-ga-mapping`**

ibm8209_Spanning_Tree

Removes bridges from participating in spanning tree protocols with IBM 8209 bridges.

Example: `disable IBM8209_spanning_tree`

spanning-tree-explorer *port#*

Disables a port from allowing propagation of spanning tree explorer frames if source routing is enabled. This command is used only if transparent bridging is not enabled on the port. In that case, it is automatically known in conformance with the transparent spanning tree.

Example: `disable spanning-tree-explorer 2`

source-routing *port#*

Disables source routing on a given port. This command is used to have an already-participating bridge interface discontinue source routing.

Example: `disable source-routing 2`

sr-tb-conversion

Disables conversion of source routed frame to transparent frame and vice versa.

Example: `disable sr-tb-conversion`

stp

Disables the Spanning Tree Protocol on the bridge. The default is enabled.

Example: `disable stp`

transparent *port#*

Disables transparent bridging function on the given port. This command is useful for cases where an alternative communication method such as source routing is desirable.

Note: This command might bring about an absurd configuration if not used properly. For instance, using it on an ethernet interface will result in disabling bridging function for that interface. This command is used to bring about SRB and SR-TB bridge function.

Example: `disable transparent 2`

tree *port#*

Disables STP participation for the bridge on a per-port basis.

Example: `disable tree 1`

Note: Disabling STP on a per-port basis can produce network loops because of the existence of parallel bridges.

ub-encapsulation

Disables Ungermann-Bass OUI encapsulation of XNS frames. XNS frames are forwarded to both Ethernet and Token Ring using SNAP encapsulation with an OUI of all zeros.

Example: `disable ub-encapsulation`

Enable

Use the **enable** command to enable the following bridging functions:

- Bridging function (entire bridging function)
- Creation of duplicate frames for mixed bridging environments (network traffic management)
- Mapping between group address and functional address
- Propagation of spanning tree explorer frames
- Source routing on a given port
- Reception of spanning tree explorer frames over a tunnel
- Conversion of source routed frame to transparent frame
- Transparent (spanning tree) bridging function on a given port

For the IP tunnel feature, the **enable** command enables a tunnel between end stations across an IP internetwork.

Syntax: enable bridge . . .
dls
duplicate
ethertype-ibmrt-pc
fa-ga-mapping
ibm8209_Spanning_Tree
spanning-tree-explorer . . .
source-routing . . .
sr-tb-conversion
stp
transparent . . .
tree
ub-encapsulation

bridge

Enables transparent bridging function on all the LAN devices (interfaces) configured in the bridging router. The port numbers are assigned to each interface as the previous interface number plus 1. For example, if interface 0 is a LAN device its port number will be 1.

Example: `enable bridge`

dls

Enables the operation of DLSw on the bridge. The router running DLSw appears as a bridge to the end stations. See Chapter 27, “Using and Configuring DLSw” on page 27-1 from more information.

Example: `enable dls`

duplicate *frame-type*

Enables the generation of duplicate STE (spanning tree explorer) or TSF (transparent spanning frames) frames. This command is available to offset the **disable duplicate** command. Duplicate frame generation is enabled by default. The **enable duplicate** command may be followed by a frame type of **TSF** or **STE** to specifically enable one of the frame types, or by the frame type **BOTH**, which yields the same behavior as not specifying a frame type for this parameter.

Example: **`enable duplicate STE`**

```
Port Number [1]?
```

ethertype-ibmrt-pc

Enables translation of SNA frames to Ethernet Type 2 as used by IBM PC RTs running OS/2 EE. This will result in SNA frames being duplicated into both 802.3/802.2 and IBM-RT formats to unknown hosts on an Ethernet.

Example: **`enable ethertype-ibmrt-pc`**

```
Port Number  [4]?
```

fa-ga-mapping

Enables group address to functional address (and vice versa) mapping. This mapping is conducted when frames are forwarded between token ring and other media (except serial line). In the token-ring arena, functional addresses are more popular even though they are locally assigned group addresses due to restrictions in hardware. On other media, group addresses are widely used. Under normal circumstances group address to functional address mapping is inevitable.

Mapping is enabled by default if mapping addresses have been added. The enable/disable mapping lets users have a choice when it comes to deleting added map records.

Example: **`enable fa-ga-mapping`**

IBM8209_Spanning_Tree

Allows bridges to participate in spanning tree protocols with IBM 8209 bridges.

Example: **`enable IBM8209_spanning_tree`**

spanning-tree-explorer *port#*

Enables the port to allow propagation of spanning tree explorer frames if source routing is enabled. This command is valid on token-ring and WAN ports only. This feature is enabled by default when source routing is configured on the port.

Example: **`enable spanning-tree-explorer 2`**

source-routing *port# segment# [bridge#]*

Enables source routing for a given port. This command is typically used when source routing on part of the bridge is desired. If source routing is the only feature desired, transparent bridging on the interface should be disabled. For the first instance of the command, entering the bridge number is required. For subsequent times, this input is not required.

port# Valid port participating in the bridge configuration.

Valid Values: X'0' to X'FFF'

Default Value: 1

segment#	12-bit number that represents the LAN/WAN to which media is attached. All the media on other bridges attached to this LAN/WAN must be configured with the same value. For correct operation of the source routing function, it is very important that all the bridges attached to this LAN/WAN have the same perspective of the LAN/WAN identification value.
bridge#	4-bit value unique among all the bridges attached to the same LAN/WAN. This value is required when source routing is enabled on the first interface. For later interfaces, this input is optional. It is recommended that the bridge# be unique on the segment. **Valid Values**: X'0' to X'F' **Default Value**: 1

Note: If the configuration is a situation where two segments have already been configured (that is a 1:N SRB configuration), you will be prompted for an additional *virtual-segment#* parameter.

Example: `enable source-routing 2 1 1`

sr-tb-conversion

Allows for compatibility between source routing and transparent bridging domains. When this feature is enabled, the bridge lets source-routed frames be accepted into a transparent domain by stripping off the RIF field and converting them into transparent frames.

The bridge also gathers routing information concerning source routing stations from the passing source routing frames. This is obtained from the RIF. This RIF information is then used to convert a transparent frame to a source-routed frame. If an RIF is not available for a station, then the frame is sent out as a spanning tree explorer frame in the source routing domain.

In order for the conversion function to operate properly, the transparent bridging domain must be given a segment number. All SR-TB bridges that are connected to this domain should also be configured with the same segment number.

TB-Domain Segment Number Valid Values: X'1' - X'FFF'

TB-Domain Segment Number Default Value: 1

The maximum transmission unit (MTU) is the number of octets per frame of data that can be transferred across a given physical network. When an IP datagram travels from one host to another, it can cross different physical networks. Some physical networks may have this set MTU which will not allow long IP datagrams to be placed in on physical frame. Fragmentation will occur when you attempt to transmit frames larger than that which the physical network can handle.

TB-Domain MTU Valid Values: 576 to 18000 bytes

TB-Domain MTU Default Value: 2048

Example: `enable sr-tb-conversion`

```
TB-Domain Segment Number in hex(1 - FFF) [1]? 2
Bridge Virtual Segment Number in hex[1 - FFF]? aa
TB-Domain's MTU [1470]? 1455
TB-Domain's MTU is adjusted to 1350
```

stp

Enables the spanning tree protocol on the bridge. This is the default.

Example: `enable stp`

transparent *port#*

Enables transparent bridging function on the given port. Under normal circumstances, this command is not necessary.

Example: `enable transparent`

```
Port Number [1]?
```

tree *port#*

Enables STP participation for the bridge on a per-port basis.

Example: `enable tree 1`

ub-encapsulation

Causes XNS Ethernet Type 2 frames to be translated into Token-Ring frames using the Ungermann-Bass OUI in the SNAP header. Token-Ring frames containing the UB OUI header will be forwarded to Ethernets as type 0x0600 Ethernet Type 2 frames rather than as 802.3/802.2 frames.

Example: `enable ub-encapsulation`

List

Use the **list** command to display information about the complete bridge configuration or to display information about selected configuration parameters.

Syntax: list
address
bridge
filtering . . .
mapping . . .
permanent . . .
port . . .
prot-filter . . .
protocol
range . . .

address *addr value*

Reads an address entry from the permanent database. The addr value is the MAC address of the desired entry. It can be an individual address, multicast address, or broadcast address. Permanent database es are not destroyed by the power off/on process and are immune to the aging settings. Permanent entries cannot be replaced by dynamic entries.

Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Default Value: none

Example: `list address 000000123456`

```
0000-00-12-34-56    PERMANENT  Input Port:  1
                               Output ports:  1, 2
                               Input port:  2
                               Output ports:  3
ASRT config>
```

Address Address entry in 12-digit hexadecimal format.

Entry Type

Permanent
Indicates that the entry is permanent in nature and will survive power on/offs or system resets.

Reserved
Indicates that the entry is reserved by the IEEE 802.1d committee for future use. Frames destined to reserved addresses are discarded.

Registered
Indicates that the entry is meant for the bridge itself.

SAF
Appears after the entry type if source address filtering has been configured.

Input Port Displays the numbers of the input port or ports associated with that address entry.

Output Port Displays the numbers of the output port or ports associated with that address entry. Displays “NONE/DAF” to indicate that destination address filtering applies because no ports have been selected to be associated with that address entry.

bridge

Lists all general information regarding the bridge.

Example: **`list bridge`**

```
Source Routing Transparent Bridge Configuration
===============================================

Bridge:  ENABLED                        Bridge Behavior: ADAPTIVE SRT
                 +----------------------------+
-----------------| SOURCE ROUTING INFORMATION |---------------------------
                 +----------------------------+
Bridge Number:         0A               Segments:         2
Max ARE Hop Cnt:       14              Max STE Hop cnt:  14
1:N SRB:               Active           Internal Segment:  0xFF6
LF-bit interpret:      Extended
                 +-------------------+
-----------------| SR-TB INFORMATION |-----------------------------------
                 +-------------------+
SR-TB Conversion:       Enabled
TB-Virtual Segment:     0x107               MTU of TB-Domain:   1470
                 +------------------------------------+
-----------------| SPANNING TREE PROTOCOL INFORMATION |------------------
                 +------------------------------------+
Bridge Address:        00-00-00-00-00-06    Bridge Priority:   32768/0x8000

STP Participation:     IEEE802.1d and IBM-8209
                 +-------------------------+
-----------------| TRANSLATION INFORMATION |-----------------------------
                 +-------------------------+
FA<=>GA Conversion:    Enabled                UB-Encapsulation:  Disabled
DLS for the bridge:    Enabled
```

```
                 +------------------+
-----------------| PORT INFORMATION |------------------------------------
                 +------------------+
Number of ports added:  3
Port:   1      Interface:     0      Behavior:   STB only    STB:  Enabled
Port:   2      Interface:     1      Behavior:   STB & SRB   STB:  Enabled
Port:   3      Interface:     2      Behavior:   STB & SRB   STB:  Enabled
```

Bridge
Indicates current state of bridge. Values are ENABLED or DISABLED.

Bridge Behavior
Indicates method of bridging being used by that bridge. The values include STB for Transparent, SRB for Source Routing, and ADAPTIVE SRT for Source-Routing Transparent conversion bridging.

Bridge address
Bridge address specified by the user (if set).

Bridge priority
A high-order 2-octet bridge address found in the Bridge Identifier, either the MAC address obtained from the lowest-number port or the address set by the Set Bridge command.

Source Routing Bridge Number
The unique number identifying a bridge. It is used to distinguish between multiple bridges connecting the same two rings.

Number of Source Routing Segments
Indicates the number of Source Routing bridge segments configured for the Source Routing domain.

SRB: Max ARE/STE Hop cnt
The maximum hop count for frames transmitting from the bridge for a given interface associated with source routing bridging.

SR-TB Conversion
Indicates whether the source routing/transparent bridge frame conversion function is enabled or disabled.

TB-Virtual Segment
Indicates the segment number of the transparent bridging domain.

MTU for TB-Domain
Specifies the maximum frame size (maximum transmission units) the transparent bridge can transmit and receive.

1:N Source Routing
Indicates the current state of 1:N Source Routing as ACTIVE or NOT ACTIVE.

Internal Virtual Segment
Displays the virtual segment number configured for 1:N SRB bridging.

SRB LF-bit interpretation
Indicates the largest Frame (LF) bit encoding interpretation mode if source routing is enabled in this bridge. This is listed as either BASIC or EXTENDED.

FA-GA conversion
Indicates whether FA-GA conversion is enabled or disabled.

Spanning Tree Protocol Participation
Displays the types of spanning tree protocols that the bridge participates in.

DLS for the bridge
Indicates if the Data Link Switch protocol is enabled or disabled in the bridge.

Number of ports added
Number of bridge ports added to the bridging configuration.

Port Number
A user-defined number assigned to an interface by the Add Port command.

Interface Number
Identifies devices connected to a network segment through the bridge. You must add at least two interfaces to participate in bridging. An interface number of 255 is used for bridging.

Port Behavior
Indicates method of bridging being used by that port. The values include STB for Transparent, SRB for Source Routing, and SR-TB for Source Routing Transparent conversion bridging.

filtering *datagroup-option*

The following general data groups can be displayed under the **list filtering** command:

All Displays all filtering database entries.

Ethertype Displays Ethernet protocol type filter database entries.

SAP Displays SAP protocol filter database entries.

SNAP Displays SNAP protocol identifier filter database entries.

The following examples illustrate each of the **list filtering** display options.

Example 1: `list filtering all`

```
Ethernet type 0800 is routed on ports 1
IEEE 802.2 destination SAP 42 is routed on ports 1
IEEE 802 SNAP PID 00-00-00-08-00 is routed on ports 2-3
```

Descriptors used in explaining how packets are communicated include the following:

Routed
Describes packets passed to routing forwarder to be forwarded.

Filtered
Describes packets which are administratively filtered by the user setting protocol filters.

Bridged and routed
This describes a protocol identifier for which there is a protocol entity within the system which is not a forwarder. For example a link level echo protocol. Unicast packets from this protocol are bridged or locally processed if being sent to a registered address. Multicast packets are forwarded and locally processed for a registered multicast address.

All of these descriptors also apply to ARP packets with this Ethertype.

Example 2: `list filtering ethertype`

```
Ethernet type (in hexadecimal), 0 for all [0]? 0800
Ethernet type 0800 is routed on ports 1
```

Example 3: `list filtering sap`

```
SAP (in hexadecimal), 100 for all [100]? 42
IEEE 802.2 destination SAP 42 is routed on ports 1
```

Example 4: `list filtering snap`

```
SNAP Protocol ID, return for all [00-00-00-00-00]?
IEEE 802 SNAP PID 00-00-00-08-00 is routed on ports 2-3
```

mapping *add-type type-field*

Lists specific address mapping for a given protocol.

Example: `list mapping SNAP`

```
PROTOCOL TYPE          GROUP ADDRESS          FUNCTIONAL ADDRESS
=============          =============          ==================
123456-7890            12-34-56-78-90-12      12:34:56:78:90:12
```

add-type
Choice of either DSAP, Ether (Ethernet), or SNAP.

type-field
Protocol type field:

- Destination Service Access Point (DSAP) protocol type is entered in a range of 1–FE (hexadecimal).
- Ethernet (Ether) protocol type is entered in a range of 5DD–FFFF (hexadecimal).
- Subnetwork Access Protocol (SNAP) protocol type is entered in 10-digit hexadecimal format.

permanent

Displays the number of entries in the bridge's permanent database.

Example: `list permanent`

```
Number of Entries in Permanent Database:  17
```

port *port#*

Displays port information related to ports that are already configured. Port# selects the port you want to list. Specifying no number selects all ports.

Example: `list port`

```
Port Id (dec)    : 128: 5, (hex): 80-05
Port State       : Enabled
STP Participation: Enabled
Port Supports: NO Bridging
Assoc Interface : 1
Path Cost       : 0
Port ID (dec)    : 128:02, (hex): 80-02
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Path Cost        : 0
```

Port ID
The ID consists of two parts: the port priority and the port number. In the example, 128 is the priority. 1, 2, and 3 are the port numbers. In

hexadecimal format, the low-order byte denotes the port number and the high-order byte denotes the priority.

Port state
Displays current state of the specified port or ports. This can be either ENABLED or DISABLED.

Port supports
Displays bridging method supported by that port (for example, transparent bridging, source routing bridging).

SRB
Displayed only when SRB is enabled and lists source routing bridging information. This includes the SRB segment number (in hex), the Maximum Transmission Unit size, and whether the transmission of spanning tree explorer frames is enabled or disabled.

Duplicate Frames Allowed
Displays a breakdown and count of the types of duplicate frames allowed.

Assoc interface
Displays interface number associated with the displayed port.

Path Cost
Cost associated with the port which is used for possible root path cost. The range is 1 to 65535.

prot-filter *port#*

Reads a current list of the filter protocol types. Filters can be listed selectively by port or all ports can be displayed at once. Port# selects the bridge port that you want to list.

Example: **`list prot-filter 1`**

```
PORT 1
Protocol Class   :  DSAP
Protocol Type    :  01
Protocol State:  :  Filtered
Port Map         :  1, 2, 3
```

Port Number Port number is displayed for each port if all ports are selected to be displayed.

Protocol Class Displays protocol class either SNAP, Ether, or DSAP.

Protocol Type Displays protocol ID in hexadecimal format.

Protocol State Denotes that protocol is being filtered for selected port.

Port Map Displays the numbers of the ports where this type of protocol filter is present.

protocol

Displays bridge information related to the spanning tree protocol.

Example: **`list protocol`**

```
Bridge Identifiers: 32768/00000000003
Bridge-Max-Age (in seconds): 20
Bridge-Hello-Time (in seconds): 2
Bridge-Forward-Delay (in seconds): 15
```

Note: Each of these bridge related parameters is also described in detail in the previous chapter.

Bridge Identifier
8-byte value in ASCII format. If you did not set the bridge address prior to displaying this information, the low order 6 bytes will be displayed as zero, denoting that the default MAC address of a port is being used. When a bridge has been selected as the root bridge, the bridge max age and bridge hello time are transmitted by it to all the bridges in the network via the HELLO BPDUs.

Bridge-Max-Age
Maximum age (period of time) that should be used to time out spanning tree protocol related information.

Bridge-Hello-Timer
Time interval between HELLO BPDUs.

Bridge-Forward-Delay
Time interval used before changing to another state (should this bridge become the root).

range *start-index* *stop-index*

Reads a range of address entries from the permanent database. To do this, first determine the size of the database by using the **list permanent** command. From this value you can then determine a “start index” value for your entry range. The start index is in the range from 1 to the size of the database. You can then choose a “stop index” for displaying a limited number of entries. This input is optional. If the stop index is not provided the default value is the size of the database.

Address entries contain the following information:

Example: **list range**

```
Start-Index [1]? 1
Stop-index [17]? 6
ADDRESS                 ENTRY TYPE      PORT MAP

=======                 ==========      ========
01-80-C2-00-00-00       REGISTERED      Input Port:  ALL PORTS
                                        Output ports:

01-80-C2-00-00-01       RESERVED        NONE/DAF
01-80-C2-00-00-02       RESERVED        NONE/DAF
01-80-C2-00-00-03       RESERVED        NONE/DAF
01-80-C2-00-00-04       RESERVED        NONE/DAF
01-80-C2-00-00-05       RESERVED        NONE/DAF
```

Address
6-byte MAC address of the entry.

Type of Entry
Specifies one of the following types:

- Reserved - entries reserved by the IEEE 802.1d committee
- Registered - entries consist of unicast addresses belonging to proprietary communications hardware attached to the box or multicast addresses enabled by protocol forwarders
- Permanent - entries entered by the user in the configuration process which survive power on/offs or system resets
- Static - entries entered by the user in the console process which do not survive power on/offs or system resets and are ageless
- Dynamic - entries “learned” by the bridge “dynamically” which do not survive power on/offs or system resets and which have an “age” associated with the entry

- Free - locations in database that are free to be filled by address entries

Port Map
Displays outgoing port map for all incoming ports.

NetBIOS

Displays the NetBIOS configuration prompt. Enter **netbios** at the `ASRT config>` prompt to display the NetBIOS configuration prompt. See “NetBIOS Commands” on page 9-24 for an explanation of each of the NetBIOS configuration commands.

Syntax: netbios

Example: **netbios**

```
NetBIOS Support User Configuration
NetBIOS config>
```

Note: If you have not purchased the NetBIOS filtering feature, you will receive the following message if you use this command:

```
NetBIOS Filtering is not available in this load.
```

Set

Use the **set** command to set certain values, functions, and parameters associated with the bridge configuration. These include the following:

- Aging time for dynamic address entries in the filtering database
- Bridge address
- Largest Frame (LF) bit encoding interpretation for source routing
- MAC service data unit (MSDU) size
- Spanning tree protocol bridge and port parameters
- Route Descriptor (RD) limit
- Size of the bridge filtering database.

Syntax: set

```
set     age
        bridge
        filtering
        lf-bit-interpretation   . . .
        maximum-packet-size  . . .
        port
        protocol bridge
        protocol port  . . .
        route-descriptor-limit  . . .
        super-elan-id
```

age *seconds resolution*

Sets the time for aging out dynamic entries in the filtering database when the port with the entry is in the forwarding state. This age is also used for aging RIF entries in the RIF table in the case of an SR-TB bridge personality.

Enter the desired value after each prompt and press **Return**.
It can be an individual address, multicast address, or broadcast address. Permanent database es are not destroyed by the power off/on process and are immune to the aging settings. Permanent entries cannot be replaced by dynamic entries.

Aging Time Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Aging Time Default Value: none

The resolution value specifies how often dynamic entries in the filtering database should be scanned to determine if they have expired their age limit as set by the aging timer.

Resolution Valid Values: 1 to 60 seconds

Resolution Default Value: 5 seconds

Example: **set age**

```
seconds [300]  ?  400
resolution [5] ?  6
```

bridge *bridge-address*

Sets the bridge address. This is the low-order 6-octet bridge address found in the bridge identifier. By default, the bridge-addr-value is set to the medium access control (MAC) address of the lowest-numbered port at initialization time. You can use this command to override the use of the default address and enter your own unique address.

Note: Each bridge in the network must have a unique address for the spanning tree protocol to operate properly.

Caution: In cases where a serial line interface (or tunnel) is the lowest numbered port, it is mandatory to use this command so that the bridge will have a unique address when restarted. This process is necessary because serial lines do not have their own MAC address.

At the prompt, enter the bridge address in 12-digit hexadecimal format and press Return.

If you enter the address in the wrong format you will receive the message `Illegal Address`. If you enter no address at the prompt you will receive the message `Zero length address supplied` and the bridge will maintain its previous value. To return the bridge address to the default value, enter an address of all zeroes.

Valid Values: 12 hexadecimal digits

Do not use dashes or colons to separate each octet. Each bridge in the network must have a unique address for the spanning tree protocol to operate properly.

Default Value: 000000000000

Example: **set bridge**

```
Bridge Address (in 12-digit hex)[]?
```

filtering *database-size*

Sets the number of entries that can be held in the bridge filtering database.

Default Value: 1024 times the number of bridge ports.

For more information, see the **list filtering** command on page 7-24.

Example: **set filtering**

```
database-size [2048]?
```

lf-bit-interpretation *encode-mode*

Sets the Largest Frame (LF) bit encoding interpretation if source routing is enabled in this bridge.

Example: `set lf-bit-interpretation basic`

Encode-mode	Entered as either **basic** or **extended**. In the basic mode only 3 bits of the routing control field are used. This is the common practice in source routing bridges that exist today. In extended mode, 6 bits of the routing control field are used to represent the maximum data unit that the bridge supports. The default value is **extended**. Extended and Basic nodes are compatible.

maximum-packet-size *port# msdu-size*

Sets the largest MAC service data unit (MSDU) size for the port, if source routing is enabled on this port. The MSDU value setting has no implication on traditionally transparent media. An MSDU value greater than the packet size configured in the router will be treated as an error.

If this parameter is not set, the default value used is the size configured as the packet size for that interface.

Valid Values: Specify an integer in the range of 16 - 65535

Default Value: packet size set for the port

Example: `set maximum-packet-size 1 4399` port *block* or *disable*

Begins the port’s participation in the spanning tree protocol. This is done by entering a status value of “block.” This places the port in the “blocked” status as a starting point. The actual state of the port will later be determined by the spanning tree protocol as it determines its topology. Entering a status value of “disable” removes the port from participating in the spanning tree.

Example: **set port block**

```
Port Number [1]?
```

protocol *bridge* or *port*

Modifies the spanning tree protocol bridge or port parameters for a new configuration, or tunes the configuration parameters to suit a specific topology.

Enter “bridge” as the option to modify bridge parameters. The bridge-related parameters that can be modified with this command are described below.

Enter **srb** or **tb** to specify whether the source routing bridge (srb) or transparent bridge (tb) spanning tree protocol parameters are to be affected.

When setting these values, make sure that the following relationships exist between the parameters or the input will be rejected:

2 X (Bridge Forward Delay - 1 second) ≥ Bridge Maximum Age
Bridge Maximum Age ≥ 2 X (Bridge Hello Time + 1 second)

Example: **set protocol bridge tb**

```
Bridge Max-Age [20]  25
Bridge Hello Time [2]   3
Bridge Forward Delay [15] 20
Bridge Priority [32768]    1
```

Bridge Maximum Age

Maximum age (period of time) that should be used to time out spanning tree protocol related information.

When this bridging router is selected as the root bridge in a spanning tree, the value of this parameter specifies how long other active bridges are to store the configuration bridge protocol data units (BPDUs) they receive. When a BPDU reaches its maximum age limit without being replaced, the active bridges in the network discard it and assume that the root bridge has failed. A new root bridge is then selected.

Dependencies

The setting of this parameter may be affected by the setting of the Bridge Hello Time parameter. In addition, the setting of this parameter may affect the setting of the Bridge Forward Delay parameter.

Valid Values: 6 to 40 seconds

Default Value: 20 seconds

Bridge Hello Timer

Time interval between HELLO BPDUs.

When this bridging router is selected as the root bridge in a spanning tree, this parameter specifies how often this bridge transmits configuration bridge protocol data units (BPDUs). BPDUs contain information about the topology of the spanning tree and reflect changes to the topology.

Dependencies

The setting of this parameter may affect the setting of the Max age parameter.

Valid Values: 1 to 10 seconds

Default Value: 2 seconds

Bridge Forward Delay

Time interval used before changing to another state (should this bridge become the root).

When this bridging router is selected as the root bridge in a spanning tree, the value of this parameter specifies how long active ports in all bridges remain in a *listening state*. When the forward delay time expires, ports in the listening state go into the *forwarding state*. State changes occur as a result of changes in the topology of the spanning tree, such as when an active bridge fails or is shut down.

The root bridge conveys this value to all bridges. This process ensures that all bridges are consistent between changes.

Dependencies

The setting of this parameter may be affected by the setting of the SRB Bridge Max Age parameter.

Valid Values: 4 to 30 seconds

Default Value: 15

Bridge Priority

A high-order 2-octet bridge address found in the Bridge Identifier - either the MAC address obtained from the lowest-numbered port or the address set by the **Set Bridge** command.

The bridge priority indicates the chances that this bridge will become the root bridge of the spanning tree. The lower the numerical value of the bridge priority parameter, the higher the priority of the bridge and the more likely it is to be chosen. The spanning tree algorithm chooses the bridge with the lowest numerical value of this parameter to be the root bridge.

Valid Values: 0 to 65535

Default Value: 32768

Enter “port” as the option to modify the spanning tree protocol port parameters. Enter the desired value at each prompt and press **Return**.

Example: **`set protocol port`**

```
Port Number [1] ?
Port Path-Cost (0 for default) [0] ?   1
Port Priority [128] ?   1
```

Port Number

Bridge port number; selects the port for which the path cost and port priority will be changed.

Path Cost

Cost associated with the port which is use for possible root path cost.

Each port interface has an associated path cost which is the relative value of using the port to reach the root bridge in a bridged network. The spanning tree algorithm uses the path cost to compute a path that minimizes the cost from the root bridge to all other bridges in the network topology.

This parameter specifies the cost associated with passing frames through this port interface, should this bridging router become the root bridge. Factor this value in when determining spanning tree routes between any two stations. A value of 0 instructs the bridging router to automatically calculate a path cost for this port using its own formula.

Valid Values: 1 to 65535

Default Value: 0 (means the cost will be calculated automatically)

Port Priority

Identifies port priority for the specified port sed by the spanning tree algorithm in making comparisons for port selection (which port offers the lowest cost path to the root bridge) and blocking decisions.

Valid Values: 0 to 255

Default Value: 128

route-descriptor-limit *limit-type* *RD-limit-value*

Allows the user to associate a maximum Route Descriptor (RD) length for all route explorer (ARE) or spanning tree explorer (STE) frames forwarded by the bridge if source routing is enabled.

Example: **`set route-descriptor-limit ARE 14`**

Limit-type

Entered either as ARE or STE depending on whether the RD-limit-value is applied to all route explorer (ARE) or spanning tree explorer (STE) frames.

RD-limit-value

Specifies the maximum number of RDs that might be contained in the routing information field (RIF) of the frame type specified by the RD limit type.

The hop count for each frame is the number of bridges through which the frame has traveled so far. One RD is added to the Routing Information Field each time the frame passes through a bridge. Therefore, the number of RDs equals the number of hops. When the number of RDs (hops) exceeds the number of hops allowed by this parameter, the frame is discarded.

Valid Values: 0 to 14

Default Value: 14

Tunnel

Use the **tunnel** command to access the Tunnel configuration prompt for a specific tunnel. Tunnel configuration commands are entered at this prompt. See “Tunnel Configuration Commands” for an explanation of each of these commands.

Syntax: tunnel *tunnel-id*

Example: `tunnel 2`

Once a port is configured, all other commands that need a port number as a parameter can function with this port.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Tunnel Configuration Commands

This section summarizes and then explains the Tunnel configuration commands. The Tunnel configuration commands allow you to specify network parameters for specified tunnels that transmit bridging frames over IP.

Configuration commands for specifically defined tunnels are entered at the `TNL config>` prompt. This prompt is accessed by entering the **tunnel** command at the `ASRT config>` prompt. Table 7-2 on page 7-35 shows the tunnel configuration commands.

Table 7-2. Tunnel Configuration Commands

Command	Function
? (Help)	Lists all of the Tunnel configuration commands, or lists the options associated with specific commands.
Add	Adds the IP address of destination bridges participating in an IP unicast or multicast addressing configuration for bridging over IP.
Delete	Deletes the IP address of a destination bridge participating in an IP unicast or multicast addressing configuration for bridging over IP.
Join	Configures the router as a member of one or more multicast groups.
Leave	Removes the router as a member of multicast groups.
List	Displays the IP addresses of end-stations participating in an IP unicast or multicast addressing configuration for bridging over IP. Also displays the size (in number of bytes) of bridging packets being routed through an IP tunnel and whether or not multicast addressing is enabled or disabled.
Set	Sets a base multicast IP address for multicast tunneling on the router.
Exit	Exits the tunnel configuration process and returns to the ASRT environment.

Tunneling and Multicast Packets

For tunnel configurations where multicast packets are involved, the source address of the multicast packets must lie on a network segment that is capable of the Internet Group Management Protocol (IGMP).

IGMP is not defined on ATM configurations. This means that when you run multicast applications on the router (for example, the MOSPF tunnel), you must ensure that one of the following conditions exists:

- The source is one of the LAN segment addresses
- The source is the internal IP address

The first condition can be ensured by using the IP **set router-id** configuration command. The second condition can be ensured by using the IP **set internal-ip-address** configuration command.

In all cases, the second option is preferred and the first should be used only if some of the routers in the network do not like host addresses (this would happen in mixed vendor networks).

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example:
```
?
```

or

```
list ?
```

Add

Use the **add** command to add the IP address of end stations participating in a unicast or multicast IP addressing configuration.

For IP unicast addressing, the tunneling configuration requires that you supply IP addresses of destination bridges. This record will be used by the router software to convert the segment number in the routing information field (RIF) in a source routed frame to the corresponding IP address of the destination bridge. For transparent bridging frames, it identifies the other endpoint of the tunnel.

For IP multicast addressing, the tunneling configuration requires only the IP multicast address reserved for tunneling. Encapsulation uses three groups of IP multicast addresses. The first group is for sending all-routes explorer (ARE) frames, the second group for sending spanning tree explorer (STE) frames, and the third group for specifically routed frames (SRF).

Note: The bridging router software transparently differentiates between unicast and multicast addresses.

Syntax: add address *IP-address*

Valid Values: a valid IP address

Default Value: none

Example: `add address 128.185.144.37`

Delete

Use the **delete** command to delete the IP address of bridges participating in a unicast or multicast IP addressing configuration.

Syntax: delete address *IP-address*

Valid Values: a valid IP address

Default Value: none

Example: `delete address 128.185.144.37`

Join

Use the **join** command to establish the router as a member of one or more multicast groups. A tunnel group may be one of three types: peer, client, or server. The tunnel group is defined by an integer tag. A bridge can belong to only one group type for each tag. A bridge cannot belong to both *peer group 1* and *server group 1*, for example.

Syntax: join client-group *group-number*
peer-group *group-number*
server-group *group-number*

client-group *group-number*

Joins the client group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `join client-group 3`

peer-group *group-number*

Joins the peer group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `join peer-group 5`

server-group *group-number*

Joins the server group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `join server-group 7`

Leave

Use the **leave** command to remove the router as a member of multicast groups.

Syntax: leave
server-group *group-number*
client-group *group-number*
peer-group *group-number*

server-group *group-number*

Leaves the server group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `leave server-group 7`

client-group *group-number*

Leaves the client group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `leave client-group 3`

peer-group *group-number*

Leaves the peer group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `leave peer-group 5`

List

Use the **list** tunnel command to display the IP addresses of bridges participating in an IP unicast or multicast addressing configuration for tunneling over IP. This command can also be used to display the current size of IP packets being sent through the tunnels and displays whether or not IP is enabled or disabled.

Syntax: list address
all

address

Lists the IP addresses of bridges participating in an IP unicast or multicast addressing configuration for tunneling over IP.

Example: **`list address`**

```
IP Tunnel Addresses
128.185.179.51      128.185.170.51      128.185.142.39
128.185.143.39       224.0.0.5
```

all

Lists all unicast IP addresses, configured multicast addresses, and the tunnel packet size.

Example: **`list all`**

```
IP Tunnel Addresses
128.185.179.51      128.185.170.51      128.185.142.39
128.185.143.39       224.0.0.5
Frame size for the tunnel 2120
```

Set

Use the **set** command to set the base multicast address of the router.

Syntax: set base-multicast-address

base-multicast-address

Sets the base multicast IP address for multicast tunneling.

Valid Values: any valid IP address

Default Value: none

Example: **`set base-multicast-address 224.10.0.0`**

Exit

Use the **exit** command to return to the previous ASRT prompt level.

Syntax: exit

Example: **`exit`**

Frame-Relay Commands

To enable bridging over the frame-relay interface, you must associate a DLCI number (also called a circuit number) with a bridge port.

Once a bridge port is configured, all the function associated with the bridge ports, including protocol filtering and address filtering are available.

At the `ASRT config>` prompt, use the following command to enable bridging on the frame-relay interface:

add port *interface#* *port#* *circuit-number*

interface# The interface number of the Frame Relay interface.

port# The unique bridge-specific number associated with the permanent virtual circuit (PVC).

circuit-number The DLCI (circuit) number on which bridging is being enabled.

The command associates a port number with a frame-relay PVC that is identified by the *circuit number*, and enables that circuit's participation in transparent bridging.

Note: The *port#* must be unique to the router, and there cannot already be a record for the *interface#* and *circuit number*.

Once a port is configured, all other commands that need a port number as a parameter can function with this port. When you enter the command **list bridge**, the following information is displayed for Frame Relay bridge ports:

```
Port number: 101 Interface number: 2    Port behavior: STB only
Circuit number: 16
```

When you enter the command **list port**, the following information is displayed for Frame Relay bridge ports:

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Port ID (dec)         : 128, 101, (hex) : 80-01
Port State            : ENABLED
Port Supports         : Transparent Bridging
Assoc Interface #/name : 2/FR/0  Circuit number:  16
Path Cost             : 0
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

List

Use the **list** command to display the following information, specific to the frame-relay circuits with bridging enabled.

list adaptive-bridge config

```
Adaptive bridge:            Disabled
Translation database size:  0
Aging time:                 300 seconds
Aging granularity:          5 seconds

Port  Segment  Interface       State   MTU
2      3         FR/0:16         Up     0
```

In the example, 16 is the number of the circuit over which bridging is enabled.

list database

```
MAC address          MC*      Entry Type         Age
Port(s)
00-00-C2-10-25-93             Perm Dest Filter
101(FR/0:16)
```

In the example, 16 is the DLCI of the circuit over which bridging is enabled.

For each console command where the interface number is displayed in the form FR/0, the frame-relay circuit is shown in the form FR/instance#:circuit#, where:

instance# The instance of the FR interface (for example, FR/0 represents the first instance of a FR interface)

circuit# The circuit number/DLCI over which bridging is enabled

Example: **`list source counters port 101`**

```
SRT> list source counters port 101
Counters for port 101, segment 013, interface FR/0:16
SRF frames received
```

Chapter 8. Monitoring Bridging

This chapter describes how to monitor the ASRT (Adaptive Source Routing Transparent) Bridge and how to use the ASRT console commands. Console commands for the bridging router's tunnel and NetBIOS features are also included as part of the general ASRT console command set. The chapter includes the following sections:

- "Accessing the ASRT Console Environment"
- "ASRT Console Commands"
- "BAN Console Commands" on page 8-19

Accessing the ASRT Console Environment

To access the ASRT console environment, enter the **protocol asrt** command at the + (GWCON) prompt:

```
+protocol asrt
ASRT>
```

ASRT Console Commands

This section summarizes and then explains the ASRT console commands. These commands allow you to view and modify parameters from the active console. Information you modify with the console commands is reset to the SRAM configuration when you restart the bridging router.

You can use these commands to temporarily modify the configuration without losing configuration information in the bridge memory. The ASRT> prompt is displayed for all ASRT console commands.

Monitoring commands for NetBIOS are entered at the NetBIOS> console prompt. The NetBIOS prompt is a subset of the major ASRT commands and is accessed by entering the ASRT **netbios** command explained later in this chapter.

Monitoring commands for NetBIOS are entered at the NetBIOS> console prompt. The NetBIOS-filtering prompt is a subset of the major ASRT commands.

Monitoring and dynamic reconfiguration VLANS commands are entered at the VLAN> console prompt. The VLAN> command is accessed by entering the **VLANS** command explained later in this chapter.

Note: For commands requiring you to enter MAC Addresses, the addresses can be entered in the following formats:

IEEE 802 canonical bit order 00-00-00-12-34-56
IEEE 802 canonical bit order (shorthand format) 000000123456
IBM Token-Ring native bit order (noncanonical) 00:00:00:12:34:56

Table 8-1 on page 8-2 shows the ASRT console commands.

Table 8-1. ASRT Console Commands Summary

Command	Function
? (Help)	Lists all the ASRT console commands or lists the options associated with specific commands.
Add	Adds permanent (static) address entries to the bridging router's permanent database.
BAN	Allows you to access the boundary access node (BAN) console prompt for entering specific BAN console commands. See “BAN Console Commands” on page 8-19 for a detailed description.
Cache	Displays cache entries for a specified port.
Delete	Deletes MAC addresses entries from the bridging router database.
Flip	Flips MAC address from canonical to 802.5 (noncanonical or IBM) bit order.
List	Displays information about the complete bridge configuration or about selected configuration options.
NetBIOS	Displays the NetBIOS monitoring prompt.
VLANS	Displays the VLAN console prompt.
Exit	Exits the ASRT console process and returns to the GWCON environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: `?`

or

```
list ?
```

Add

Use the **add** command to add static address entries and destination address filters to the bridging router’s database. These additions to the database are lost when you restart the router.

Syntax: add destination-address-filter
static-entry . . .

destination-address-filter *mac_address*

Adds a destination address filter to the bridging router’s permanent database. Enter the command followed by the MAC address of the entry.

Example: **add destination-address-filter**

```
Destination MAC address [00-00-00-00-00-00]?
```

static-entry *mac_address* *input_port* *[output_ports]*

Adds static address entries to the bridging router’s permanent database. Enter the command followed by the MAC address of the static entry and the input port number (an optional output port number may also be entered). To create a static entry with multiple port maps (1 per input port), use this command several times.

Example: `add static-entry`

```
MAC address [00-00-00-00-00-00]?  400000012345
Input port, 0 for all [0]? 2
Output port, 0 for none [0]? 3
Output port, 0 to end [0]?
```

BAN

Use the **ban** command to access the BAN (Boundary Access Node) console prompt. Enter the **ban** command from the `ASRT>` prompt.

Syntax: ban

Example: `ASRT>ban`

```
BAN>
```

Once you access the BAN console prompt, you can begin entering specific console commands. To return to the `ASRT>` prompt at any time, enter the **exit** command.

Cache

Use the **cache** command to display the contents of a selected bridging-port routing cache. If the port does not possess a cache you will see the message `Port X does not have a cache`.

Syntax: cache *port#*

Example: `cache`

```
Port number [1]? 3

MAC Address    MC*   Entry Type       Age  Port(s)

00-00-93-00-C0-D0   PERMANENT         0  3 (TKR/1)
00-00-00-11-22-33   STATIC            0  3 (TKR/1)
```

MAC Address 6-byte MAC address of the entry.

Entry Type Specifies one of the following address entry types:

Reserved - entries reserved by the IEEE802.1D Standard.

Registered - entries consist of unicast addresses belonging to proprietary communications hardware attached to the box or multicast addresses enabled by protocol forwarders.

Permanent - entries entered by the user in the configuration process which survive power on/offs or system resets.

Static - entries entered by the user in the console process which do not survive power on/offs or system resets and are not effected by the aging timer.

Dynamic - entries "learned" by the bridge "dynamically" which do not survive power on/offs or system resets and which have an "age" associated with the entry.

Free - locations in database that are free to be filled by address entries.

Unknown - entry types unknown to the bridge. May be possible bugs and/or illegal addresses.

Age Age in seconds of each dynamic entry. Age is decremented at each resolution intervals.

Port(s) Specifies the port number associated with that entry and displays the interface name (this will always be that of the interface having the cache).

Delete

Use the **delete** command to delete station (including MAC) address entries from the router's permanent database.

Syntax: delete *MAC-address*

Example: `delete 00-00-93-10-04-15`

Flip

Use the **flip** command to view specific MAC addresses in the canonical and noncanonical format by "flipping" the address bit order. This command is useful for translating IEEE 802.5 addresses in their typical noncanonical format to the canonical format universally used by the bridge console and ELS (and vice versa).

Syntax: flip *MAC-address*

Example: `flip`

```
MAC address [00-00-00-00-00-00]? 00-00-00-33-44-55
IEEE 802 canonical bit order:    00-00-00-33-44-55
IBM Token-Ring native bit order: 00:00:00:CC:22:AA
```

List

Use the **list** command to display information about the bridging router configuration or to display information about selected configuration or bridging options.

Syntax: list
adaptive-bridge . . .
bridge . . .
conversion . . .
database . . .
filtering . . .
port
super-elan-bridge cache
source-routing . . .
spanning-tree-protocol . . .
transparent . . .
tunnel . . .

adaptive-bridge *datagroup-option [sub-option]*

Lists all general information regarding the SR-TB bridge which converts between types of bridging. There are a number of general datagroup options which may be displayed under **list adaptive-bridge**. These include the following:

- Config - Displays general information regarding the SR-TB bridge.
- Counters - Displays all SR-TB bridge counters.
- Database - Displays contents of the SR-TB bridge RIF database.

The following examples illustrate each of the **adaptive-bridge** display options.

Example: **`list adaptive-bridge config`**

```
      Adaptive bridge:            Enabled
      Translation database size:  0
      Aging time:                 320 seconds
      Aging granularity           5 seconds

   Port  Segment  Interface   State     MTU
      1  001      TKR/1      Enabled   2052
      -  001      Adaptive   Enabled   1470
```

Conversion bridge Shows the current state of the SR-TB conversion bridge. This value is displayed as either Enabled or Disabled.

Translation database size Displays the current size of the SR-TB database, which contains MAC addresses and associated RIFs for the source-routing domain.

Aging time Displays the aging timer setting in seconds. All SR-TB RIF database entries which exceed this time limit are discarded.

Aging granularity Displays how often entries are scanned to look for expiration according to the aging timer.

Port Displays the number of a port associated with conversion bridging.

Segment Displays the source routing segment number assigned to the port associated with conversion bridging.

Interface Identifies the device connected to a conversion bridge network segment and the VPI/VCI if an ATM port.

State Indicates the current state of the conversion bridge port.

MTU Specifies the maximum frame size (from the end of the RIF to the beginning of the FCS) that the conversion bridge can transmit and receive.

Example: **`list adaptive-bridge counters`**

```
Hash collision count: 28
Adaptive. database overflow  count: 0
```

Hash Collision Count Displays number of addresses that were stored (hashed) to the same location in the hash table. This number is accumulative and reflects the total number of hash collision incidents that occurred. Increases in this number may indicate a potential table size problem.

Adaptive Database Overflow Displays the number of times that an address was overwritten as the conversion database table ran out of table space.

The *database* option of the **list adaptive-bridge** command lets you list select certain portions of the adaptive bridge RIF database to display. This is due to the potential size of the database. The display options include the following:

- Address - Displays the conversion bridge database related to that specific MAC address
- All - Displays the entire database.
- Port - Displays all conversion bridge entries a specific port.
- Segment - Displays all conversion bridge entries associated with the port having the specified segment number.

The following examples illustrate each of the **list adaptive-bridge database** command options.

Note: These are only displayed if adaptive bridging is enabled.

Example: `list adaptive-bridge database address` ***mac-address***

Example: `list adaptive-bridge database all`

Example: `list adaptive-bridge database port` ***segment#***

Example: `list adaptive-bridge database segment` ***segment#***

Each entry is displayed on two lines followed by a blank line. The following information is displayed for each entry:

Canonical address Lists the MAC address of the node corresponding to this entry. This is displayed in IEEE 802 canonical (hexadecimal) format.

Interface Displays the name of the network interface that learned this entry.

Port Displays the port number of the port that learned this address entry.

Seg Displays the number of the segment that learned this address.

Age Displays the entry age in seconds.

RIF Type Displays the RIF type as SRF, STE, or ARE.

RIF Direction Displays the RIF direction as Forward or Reverse.

RIF Length Displays the RIF length in bytes.

RIF LF Displays the largest frame value encoded in the RIF.

IBM MAC Address Shows the MAC address of the node corresponding to this entry. This is displayed in the “IBM” noncanonical bit order as typically labeled on 802.5 interfaces and used by the IP/ARP, IPX, and NetBIOS protocols.

RIF Displays the Routing Information Field learned from this node.

bridge

Lists all general information regarding the bridge router configuration.

Example: `list bridge`

```
Bridge ID (prio/add):  32768/10-00-5A-63-01-00
Bridge state:          Enabled
UB-Encapsulation:      Disabled
Bridge type:           STB
Bridge capability:     ASRT
Number of ports:       2
STP Participation:     IEEE802.1d

                                                      Maximum
Port  Interface      State  MAC Address         Modes   MSDU   Segment

   1  Eth/1            Up   10-00-5A-63-01-00       T    1514
   2  FR/0:16        Down   00-00-00-00-00-00               0


SR bridge number:      7
SR virtual segment:    001
Adaptive segment:      000
```

Bridge ID
Unique ID used by the spanning tree algorithm in determining the spanning tree. Each bridge in the network is assigned a unique bridge identifier. The bridge priority is displayed in decimal followed by the hex address.

Bridge State
Indicates whether bridging is enabled or disabled.

Bridge Type
Displays the configured bridge type. This is displayed as NONE, SRB, TB, SRT, ADAPT, A/SRB, A/TB, or ASRT.

Number of Ports
Displays the number of ports configured for that bridge.

Port
Specifies a user defined number assigned to an interface by the Add Port command.

Interface
Identifies devices connected to a network segment through the bridge.

State
Indicates the current state of the port. This is displayed as UP or DOWN.

MAC address
Displays the MAC address associated with that port in canonical bit order.

Modes
Displays the bridging mode for that port. T indicates transparent bridging. SR indicates source routing. A indicates adaptive bridging.

MSDU
Specifies the maximum frame (data unit) size (including the MAC header but not the FCS field) the source routing bridge can transmit and receive on this interface.

Segment
Displays the source routing bridge segment number assigned to that port (if any).

SR bridge number
Displays the user assigned source routing bridge number.

SR virtual segment
Displays the source routing bridge virtual segment number show (if any).

Adaptive segment
Displays the number of the segment which is used in the source routing domain to route to the transparent domain (via conversion).

conversion *datagroup-option*

Displays general information about the bridge's rules for converting frame formats based on the frame type. There are a number of general datagroups which may be displayed under the **list conversion** command. These include the following:

- All - Displays all rules.
- Ethertype - Displays rules for all Ethernet types or for a specific Ethernet type.
- SAP - Displays rules for all SAP protocol identifiers or a specific 802.2 SAP type.
- SNAP - Displays rules for all SNAP protocol identifiers or a specific 802.2 SNAP type.

The following examples break down each of the list conversion display options.

Example: **`list conversion all`**

Example: **`list conversion ethertype`**

```
Ethernet type (in hexadecimal), 0 for all [0]?
```

Example: **`list conversion SAP`**

```
SAP (in hexadecimal), 100 for all [100]?
```

Example: **`list conversion SNAP`**

```
SNAP Protocol ID, return for all [00-00-00-00-00]?
```

database *datagroup-option*

Lists the contents of transparent filtering databases. There are a number of datagroups which can be chosen to be displayed under the list database command. These include the following:

- All - Displays the entire transparent bridging database.
- Dynamic - Displays all dynamic (learned) address database entries.
- Local - Displays all local (reserved) address database entries.
- Permanent - Displays all permanent address database entries.
- Port - Displays address entries for a specific port.
- Range - Displays a range of database entries from the total transparent bridging filtering address database. A starting and ending MAC address is given to define the range. All entries falling within this range will be displayed.
- Static - Displays static entries from the address database.

The following examples break down the list database command options. The first example also shows the related output.

Example: **`list database all`**

```
MAC Address    MC*  Entry Type       Age  Port(s)
00-00-00-00-AA-AA   Dynamic          295  4 (Eth/2)
00-00-00-12-34-56   Perm/Source filter    2 (TKR/1)  ->  3-4
                                          1-2
00-00-00-22-33-44   Permanent             1-2
                                          1-2
00-00-00-33-44-55   Perm Dest filter      All
00-00-00-55-66-77   Perm/Source filter    1-2,4

00-00-93-10-04-15   Registered            1 (Eth/1)
00-00-93-10-E4-F9   Dynamic          300  1 (Eth/1)
00-00-93-90-04-A6   Dynamic          300  1 (Eth/1)
00-00-A7-10-68-28   Dynamic          270  1 (Eth/1)
01-80-C2-00-00-00*  Registered            1,3
01-80-C2-00-00-01*  Reserved              All
01-80-C2-00-00-02*  Reserved              All
01-80-C2-00-00-03*  Reserved              All
01-80-C2-00-00-0D*  Reserved              All
01-80-C2-00-00-0E*  Reserved              All
01-80-C2-00-00-0F*  Reserved              All
03-00-00-00-80-00*  Reserved              All
08-00-17-00-35-F9   Dynamic          300  1 (Eth/1)
08-00-17-00-4D-DA   Dynamic          300  1 (Eth/1)
```

Note: The following fields are displayed for all of the **list database** command options.

MAC Address Specifies the address entry in 12-digit hex format (canonical bit order).

*MC** An asterisk following an address entry indicates that the entry has been flagged as a multicast address.

Entry Type Specifies one of the following types:

Reserved
Entries reserved by the IEEE802.1D standard.

Registered
Entries consist of unicast addresses belonging to interfaces participating in the bridge or multicast addresses enabled by protocol forwarders

Permanent
Entries entered by the user in the configuration process which survive power on/offs or system resets

Static
Entries entered by the user in the console process which do not survive power on/offs or system resets and are ageless.

Dynamic
Entries “learned” by the bridge “dynamically” which do not survive power on/offs or system resets and which have an “age” associated with the entry

Free
This type is not used and should not be normally be seen except in occasional “race” conditions between the console and the bridge.

Unknown
Unknown entry type. May indicate a software bug. Report the hex entry type to Customer Service.

Age Refers to the age (in seconds) of each dynamic entry. Age is decremented at each resolution interval.

Port(s) Specifies the outgoing port number(s) for that entry. Device type is also listed for single port entries. If dynamic entry on IP tunnel, the port will be “5” for IP tunnel.

Example: **`list database dynamic`**

Example: **`list database local`**

```
MAC Address    MC*  Entry Type       Age  Port(s)

00-00-93-B8-00-48   Registered            1 (TKR/1)
01-80-C2-00-00-00*  Registered            1
03-00-02-00-00-00*  Registered            1
ASRT>
```

Example: **`list database permanent`**

Example: **`list database port`** ***port#***

Example: **`list database static`**

Example: **`list database range`**

```
First MAC address [00-00-00-00-00-00]? 00-00-93-00-C0-D0
Last MAC address [FF-FF-FF-FF-FF-FF]? 01-80-C2-00-00-00

MAC Address    MC*  Entry Type       Age  Port(s)
00-00-93-10-04-15   Registered            1 (Eth/2)
01-80-C2-00-00-00   Registered            1,3
```

filtering *datagroup-option*

displays general information about the bridge’s protocol filtering databases. There are a number of general datagroups which may be displayed under the **list filtering** command. These include the following:

- All - Displays all filtering database entries.
- Ethertype - Displays Ethernet protocol type filter database entries.
- SAP - Displays SAP protocol filter database entries.
- SNAP - Displays SNAP protocol identifier filter database entries.

The following examples break down each of the list filtering display options.

Example: **`list filtering all`**

```
Ethernet type 0800 is routed on ports 1
IEEE 802.2 destination SAP 42 is routed on ports 1
IEEE 802 SNAP PID 00-00-00-08-00 is routed on ports 2-3
```

Descriptors used in explaining how packets are communicated include the following:

- Routed - Describes packets which are passed to routing forwarder to be forwarded

- Filtered- Describes packets which are administratively filtered by the user setting protocol filters

- Bridged and routed - This describes a protocol identifier for which there is a protocol entity within the system which is not a forwarder. An example of this would be a link level echo protocol. Unicast packets from this protocol are bridged or locally processed if being sent to a registered address. Multicast packets are forwarded and locally processed for a registered multicast address.

All of the descriptors just explained also apply to ARP packets with this Ethertype.

Example: **`list filtering ethertype`**

```
Ethernet type (in hexadecimal), 0 for all [0]? 0800
Ethernet type 0800 is routed on ports 1
```

Example: **`list filtering SAP`**

```
SAP (in hexadecimal), 100 for all [100]? 42
IEEE 802.2 destination SAP 42 is routed on ports 1
```

Example: **`list filtering SNAP`**

```
SNAP Protocol ID, return for all [00-00-00-00-00]?
IEEE 802 SNAP PID 00-00-00-08-00 is routed on ports 2-3
```

source-routing

Displays source-routing bridge configuration information. There are a number of general datagroup options which may be displayed under the list source-routing command. These include the following:

- Configuration - Displays general information regarding the SRB bridge.
- Counters - Displays all SRB bridge counters.
- State - Displays contents of all related SR-TB bridge databases.

The following examples illustrate each of the list source-routing display options.

Example: **`list source-routing configuration`**

```
Bridge number:              1
Bridge state:               Enabled
Maximum STE hop count       14
Maximum ARE hop count       14
Virtual segment:            003
Port  Segment  Interface   State     MTU    STE Forwarding  LNM
2    001      TKR/1       Enabled   4399   Yes             ENA
3    002      TKR/2       Enabled   4399   Yes
```

Bridge number The bridge number (in hexadecimal) assigned to this bridge.

Bridge State Indicates whether bridging is enabled or disabled.

Maximum STE hop count The maximum hop count for spanning tree explorer frames transmitting from the bridge for a given interface associated with source routing bridging.

Maximum ARE hop count The maximum hop count for all route explorer frames transmitting from the bridge for a given interface associated with source routing bridging.

Virtual segment The virtual segment number assigned for 1:N bridging.

Port The numbers of ports associated with source routing bridging

Segment The assigned segment numbers for networks associated with source routing bridging.

Interface The associated interface names. “Adaptive” is listed for interfaces participating in the SR-TB feature and DLCI for FR.

State The current port state (Enabled or Disabled).

MTU The MTU size set for that port.

STE Forwarding Indicates whether Spanning Tree Explorers received on this port are forwarded (Yes) and whether STEs from other ports go out this port.

LNM Indicates whether LAN Network Manager (LNM) agents are enabled (ENA) or disabled (DIS) on that specific port.

The counters option has further subgroups of information which may be displayed using the list source-routing command. These include the following:

- All-ports - Displays counters for all ports.
- Port - Displays counters for a specific port.
- Segment - Displays counters for the port corresponding to a specific segment.

The following examples illustrate each of the list source-routing display options.

Example: **`list source-routing counters all-ports`**

```
ASRT>list source counters all-ports
Counters for port 2, segment 001, interface TKR/1
SRF frames received:          0    sent:          0
STE frames received:          0    sent:          0
ARE frames received:        648    sent:          0
SR frames sent as TB:                             0
TB frames sent as SR:                          2057
Dropped, input queue overflow:                    0
Dropped, source address filtering:                0
Dropped, dest address filtering:
Dropped, invalid RIF length:                      0
Dropped, duplicate segment:                    2594
Dropped, segment mismatch:                        0
Dropped, Duplicate LAN ID or tree error:          0
Dropped, STE hop count exceeded:                  0
Dropped, ARE hop count exceeded:                  0
Dropped, no buffer available to copy:             0
Dropped, MTU exceeded:                            0


Counters for port 3, segment 002, interface TKR/2:
SRF frames received:          0    sent:          0
STE frames received:          0    sent:          0
ARE frames received:        825    sent:          0
SR frames sent as TB:                             0
TB frames sent as SR:                          2041
Dropped, input queue overflow:                    0
Dropped, source address filtering:                0
Dropped, dest address filtering:                  0
Dropped, invalid RIF length:                      0
Dropped, duplicate segment:                    3300
Dropped, segment mismatch:                        0
Dropped, Duplicate LAN ID or tree error:          0
Dropped, STE hop count exceeded:                  0
Dropped, ARE hop count exceeded:                  0
Dropped, no buffer available to copy:             0
Dropped, MTU exceeded:                            0
```

Port Lists the numbers of ports associated with source routing bridging

Segment Lists the source-routing segment numbers in hex.

Interface Lists the name of the network interface. DLCI for FR.

SRF Frames Received/Sent Lists the number of Specifically Routed Frames received or sent on this bridge.

STE Frames Received/Sent
Lists the number of Spanning Tree Explorer Frames received or sent on this bridge.

ARE Frames Received/Sent
Lists the number of All Routes Explorer Frames received or sent on this bridge.

SR Frames Sent as TB
Lists the number of source routing frames received on this interface that were sent as Transparent Bridge Frames.

TB Frames Sent as SR
Lists the number of transparent bridge frames received on this interface that were sent as source routing frames.

Dropped, input queue
Lists the number of frames arriving on this interface that were not bridged for flow control reasons. The input queue to the forwarder overflowed

Dropped, source address filtering
Lists the number of frames arriving on this interface that were not bridged because this source address matched a source address filter in the filtering database

Dropped, destination address filtering
Lists the number of frames arriving on this interface that were not bridged because this destination address matched a destination address filter in the filtering database

Dropped, protocol filtering
Lists the number of frames arriving on this interface that were not bridged because their protocol identifier was one that is being administratively filtered.

Dropped, invalid RIF length
Lists the number of frames arriving on this interface that were dropped because the RIF length as less than 2 or over 30.

Dropped, duplicate segment
Lists the number of frames arriving on this interface that were dropped because of a duplicate segment in the RIF. This is normal for ARE frames.

Dropped, segment mismatch
Lists the number of frames arriving on this interface that were dropped because the outgoing segment number does not match any in this bridge.

Dropped, Duplicate LAN ID or tree error:
The number of duplicate LAN IDs or Tree errors. This helps in the detection of problems in networks containing older IBM Source Routing Bridges.

Dropped, STE hop count exceeded:
The number of explorer frames that have been discarded by this port because the Routing Information Field has exceeded the maximum route descriptor length.

Dropped, ARE hop count exceeded:
The number of explorer frames that have been discarded by this port because the Routing Information Field has exceeded the maximum route descriptor length.

Dropped, no buffer available to copy:
Number of times a frame was not forwarded out of an interface, because there were no buffer resources available to copy the frame. (Frame to multicast destinations and to unknown destinations, need to be copied for transmission out on all active ports.)

Dropped, MTU exceeded:
The number of frames that were discarded by this port due to an excessive size. It is incremented by both transparent and source route bridges.

Example: **`list source-routing counters port 3`**

```
Counters for port 3, segment 002, interface TKR/1:
SRF frames received:          0    sent:           0
STE frames received:          0    sent:           0
ARE frames received:       1140    sent:           0
SR frames sent as TB:                              0
TB frames sent as SR:                           2931
Dropped, input queue overflow:                     0
Dropped, source address filtering:                 0
Dropped, dest address filtering:                   0

Dropped, invalid RIF length:                       0
Dropped, duplicate segment:                     4560
Dropped, segment mismatch:                         0
Dropped, Duplicate LAN ID or tree error:           0
Dropped, STE hop count exceeded:                   0
Dropped, ARE hop count exceeded:                   0
Dropped, no buffer available to copy:              0
Dropped, MTU exceeded:                             0
Dropped, dest address filtering:                   0
Dropped, protocol filtering:                       0
```

Example: **`list source-routing counters segment 2`**

```
Counters for port 3, segment 002, interface TKR/2:
SRF frames received:          0    sent:          0
STE frames received:          0    sent:          0
ARE frames received:       1249    sent:          0
SR frames sent as TB:                             0
TB frames sent as SR:                          3200
Dropped, input queue overflow:                    0
Dropped, source address filtering:                0
Dropped, dest address filtering:                  0
Dropped, protocol filtering:                      0
Dropped, invalid RI length:                       0
Dropped, duplicate segment:                    4996
Dropped, segment mismatch:                        0
Dropped, Duplicate LAN ID or tree error:          0
Dropped, STE hop count exceeded:                  0
Dropped, ARE hop count exceeded:                  0
Dropped, no buffer available to copy:             0
Dropped, MTU exceeded:                            0
```

spanning-tree protocol

Displays spanning tree protocol information. The spanning tree protocol is used by the transparent bridge to form a loop-free topology. There are a number of general datagroup options which may be displayed under the **list spanning-tree-protocol** command. These include the following:

- Configuration - Displays information concerning the spanning tree protocol.
- Counters - Displays the spanning tree protocol counters.
- State - Displays the current spanning tree protocol state information.
- Tree - Displays the current spanning tree information including port, interface, and cost information.

The following examples illustrate each of the list spanning-tree-protocol display options.

Example: **list spanning-tree-protocol configuration**

```
            Bridge ID (prio/add):  32768/0000-93-00-84-EA
            Bridge state:          Enabled
            Maximum age:           20 seconds
            Hello time:            2 seconds
            Forward delay:         15 seconds
            Hold time:             1 seconds
            Filtering age:         320 seconds
            Filtering resolution:  5 seconds

            Port  Interface  Priority  Cost   State
              4   Eth/1      128    100     Enabled
            128   Tunnel     128    65535   Enabled
```

Example: **list spanning-tree-protocol counters**

```
              Time since topology change (seconds)         0
              Topology changes:                             1
              BPDUs received:                              0
              BPDUs sent:                                  14170

Port  Interface  BPDUs received  BDPU input overflow  Forward transitions

   1  TKR/1            0               0               1
```

Example: **list spanning-tree-protocol state**

```
        Designated root (prio/add):    32768/00-00-93-00-84-EA
        Root cost:                     0
        Root port:                     Self
        Current (root) maximum age:    20 seconds
        Current (root) hello time:     2 seconds
        Current (root) Forward delay:  15 seconds
        Topology change detected:      FALSE
        Topology change:               FALSE

        Port  Interface   State
          4  Eth/1       Forwarding
        128  Tunnel      Forwarding
```

Example: **list spanning-tree-protocol tree**

```
        Port                  Designated   Desig.         Designated   Des.
        No.  Interface             Root     Cost              Bridge   Port

        1  TKR/1  32768/12-34-56-78-90-12    0 32768/12-34-56-78-90-12 90-01
```

transparent

Displays transparent bridge configuration information. There are a number of general datagroup options which may be displayed under the list transparent command. These include the following:

- Configuration - Displays information concerning the transparent bridge.
- Counters - Displays the transparent bridge counters. You may use all-ports after the command to display the counters for all ports or enter the specific port number after the command to display counters for one port.
- State - Displays the transparent state information.

  The following examples illustrate each of the list transparent display options.

Example: **`list transparent configuration`**

```
Filtering database size:    5141
Aging time:                 300 seconds
Aging granularity           5 seconds
Port  Interface   State     MTU
  4   Eth/1       Enabled   0
128   Tunnel      Enabled   0
```

Example: **`list transparent counters all-ports`**

```
Counters for port 4, interface Eth/1:
Total frames received by interface:          25885
Frames submitted to bridging:                13732
Frames submitted to routing:                  6101
Dropped, source address filtering:               0
Dropped, dest address filtering:             12677
Dropped, protocol filtering:                     0
Counters for port 128, interface Tunnel:
Total frames received by interface:              0
Frames submitted to bridging:                    0
Frames submitted to routing:                     0
Dropped, source address filtering:               0
Dropped, dest address filtering:                 0
Dropped, protocol filtering:                     0
Dropped, no buffer available to copy:            0
Dropped, input queue overflow:                   0
Dropped, source port blocked:                    0
Frames sent by bridging:                      5327
Dropped, dest port blocked:                      0
Dropped, transmit error:                         0
Dropped, too big to send on port:                0
```

Example: **`list transparent counters port 4`**

```
Counters for port 4, interface Eth/1:
Total frames received by interface:          25885
Frames submitted to bridging:                13732
Frames submitted to routing:                  6101
Dropped, source address filtering:               0
Dropped, dest address filtering:             12677
Dropped, protocol filtering:                     0
Dropped, no buffer available to copy:         6073
Dropped, input queue overflow:                 122
Dropped, source port blocked:                   31
Frames sent by bridging:                       388
Dropped, dest port blocked:                      0
Dropped, transmit error:                         0
Dropped, too big to send on port:                0
```

Example: **`list transparent state`**

```
Filtering database size:                      5141
Number of static entries:                        0
Number of dynamic entries:                      10
Hash collision count:                            1
Filtering database overflow count:               0
```

tunnel *bridges* or *config*

Displays tunnel configuration information. There are general datagroup options which may be displayed under the list tunnel command. These include:

- Bridges - Displays tunnel bridge information.
- Config - Displays information concerning the tunnel configuration.

The following examples illustrate each of the list tunnel display options.

Example: **`list tunnel bridges`**

Example: **`list tunnel config`**

NetBIOS

Use the **netbios** command to access the `NetBIOS>` prompt. NetBIOS console commands may be entered at the `NetBIOS>` prompt.

See “NetBIOS Commands” on page 9-24 for the NetBIOS console commands.

Syntax: netbios

Example: **netbios**

```
NetBIOS>
```

Note: If the NetBIOS filtering feature has not been purchased for your bridging router software load, you will receive the following message if you try to use this command:

```
NetBIOS Filtering is not available in this load.
```

Exit

Use the **exit** command to exit the ASRT console process and return to the GWCON environment.

Syntax: exit

Example: **exit**

Accessing the BAN Console Prompt

Use the **ban** command from the ASRT> or DLSw> console prompt to access BAN commands.

To access the BAN console prompt, enter the **ban** command from the ASRT console prompt of the DLSw console prompt. For example:

```
ASRT> ban
BAN>

   or

DLSw> ban
BAN>
```

Once you access the BAN console prompt, you can begin entering specific console commands. To return to the console prompt you came from, enter the **exit** command.

BAN Console Commands

This section summarizes and then explains the BAN console commands. Enter the commands at the BAN> prompt.

Table 8-2. BAN Console Commands Summary

Command	Function
? (Help)	Lists all of the BAN console commands, or lists the options associated with specific commands.
List	Displays all information concerning BAN ports.
Exit	Exits the BAN console process and returns to the previous console environment.

? (Help)

Use the **? (help)** command to list the commands that are available the current prompt level. You can also enter this command after specific command names to obtain a listing of the command options available for that command.

Syntax: ?

Example: **?**

List

Use the **list** to list information about all BAN ports. The information that is displayed includes the BAN port number, the MAC address for the BAN DLCI, whether the frames handled by the port are bridged or the LLC is terminated by DLSw, and the status of the port.

The status of the port will have one of three values:

- Init Fail - Indicates that a configuration problem exists.
- Up - Indicates that the Frame Relay DLCI is up and running.
- Down - Indicates that the DLCI is not active.

Syntax: list

Example: `list`

```
bridge  BAN                Boundary             bridged or
port    DLCI MAC Address   Node Identifier      DLSw terminated  Status
2       40:00:12:34:56:78  4F:FF:00:00:00:00    bridged          Up
```

Exit

Use the **exit** command to return to the previous prompt level (`ASRT>` or `DLSw>`).

Syntax: exit

Example: `exit`

Chapter 9. Using, Configuring, and Monitoring NetBIOS

This chapter describes IBM's implementation of NetBIOS over bridged networks and over DLSw networks. It includes the following topics:

- "About NetBIOS"
- "Reducing NetBIOS Traffic" on page 9-3
- "Frame Type Filtering" on page 9-4
- "NetBIOS Host Name and Byte Filtering Configuration Procedures" on page 9-15
- "About NetBIOS Configuration and Monitoring Commands" on page 9-20
- "NetBIOS Commands" on page 9-24

About NetBIOS

The NetBIOS protocol was designed for use on a LAN (token ring). It is not a routable protocol, but can be bridged, or switched using DLSw. Both of these methods of handling NetBIOS traffic are supported.

NetBIOS relies on broadcast frames for most of its functions other than data transfer. While this may not present a problem in LAN environments, if uncontrolled, it may easily present a problem in WAN environments.

The following sections describe NetBIOS names and the different types of NetBIOS broadcast communication.

NetBIOS Names

The key to communication between NetBIOS stations are the NetBIOS names. Each NetBIOS entity is assigned a NetBIOS name. In order to communicate with another NetBIOS entity, its NetBIOS name must be known. The names are used in broadcast NetBIOS frames to indicate the source NetBIOS entity of the frame and the desired target NetBIOS entity to receive the frame.

All names in NetBIOS frames are 16 ASCII characters. There are two types of NetBIOS names:

Individual (or unique)
Represents a single NetBIOS client or server. This name should be unique within the NetBIOS network.

This name is used to communicate with this particular NetBIOS entity.

Group Represents a group of NetBIOS stations (an OS/2 LAN Server domain, for example). This name should not be the same as any individual NetBIOS names in the network.

This name is used to allow communication between a group of NetBIOS entities.

A single NetBIOS station (single MAC address) may have multiple individual and/or group names associated with it. These names are generated by the NetBIOS application based upon a name or names configured at the NetBIOS station by a network administrator.

NetBIOS Name Conflict Resolution

When a NetBIOS entity is preparing to use an individual NetBIOS name as its own, it checks the network to make sure that no other NetBIOS station has already used this name.

It checks the NetBIOS name by repeatedly broadcasting a particular NetBIOS UI frame to all NetBIOS stations. If no stations respond, then the name is assumed to be unique and can be used. If a station does respond, the new station should not attempt to use this name.

NetBIOS Session Setup Procedure

To establish a NetBIOS session in order to do data transfer types of operations, the NetBIOS client first resolves the MAC address of the NetBIOS server and the LLC route to the NetBIOS server.

This is done by repeatedly broadcasting a particular NetBIOS UI frame to all NetBIOS stations. This frame contains the NetBIOS name of the server with which this client is establishing a session. When the server receives this frame with its NetBIOS name in it, it responds with a corresponding broadcast NetBIOS UI frame to the client. When the client receives the response frame, the frame contains the MAC address and the route to the NetBIOS server.

For some NetBIOS applications, finding the NetBIOS server is a multiple step process. For example, the first step may be to find a domain controller which tells the client which domain server to use. Then the client finds this domain server.

Once the MAC address of NetBIOS server and the route to the NetBIOS server is found, the NetBIOS client may take either of the following actions:

- Establish an LLC2 connection with the NetBIOS server to communicate with the server using I-frames.
- Begin communicating with the NetBIOS server using specifically-routed NetBIOS UI frames.

NetBIOS Broadcast Data Flows

For some NetBIOS applications, it is common to periodically broadcast data frames. This may be done if a station has a single frame’s worth of data to send to another NetBIOS station. It can do this by broadcasting a particular NetBIOS UI frame (with the target NetBIOS station’s name in the frame) to all NetBIOS stations.

Another case is when NetBIOS stations within a group (or domain) need to communicate with each other. This can be done by broadcasting a particular NetBIOS UI frame (with the target NetBIOS group name in the frame) to all NetBIOS stations. This is commonly done.

NetBIOS Status Flows

A less commonly used NetBIOS function is the ability to obtain status from any NetBIOS station. This is done by broadcasting a particular NetBIOS frame (with the target NetBIOS station’s name in the frame) to all NetBIOS stations. When the target NetBIOS station receives this frame, it responds with a corresponding broadcast NetBIOS response frame.

NetBIOS All-Stations Broadcast Frames

There are two types of NetBIOS functions that are rarely used. Both of these functions involve broadcasting a NetBIOS frame to all NetBIOS stations. There is no target NetBIOS name in the frames. The two functions are:

- NetBIOS general broadcast function – which sends a data frame to all NetBIOS stations on the network.
- NetBIOS terminate trace function – which allows a network administrator to terminate NetBIOS trace functions in all NetBIOS stations on the network from a single point. A particular NetBIOS frame is broadcast to all NetBIOS stations on the network.

Reducing NetBIOS Traffic

To stabilize a network, the goal is to reduce the amount of ;when 'ncer1' delete broadcast NetBIOS traffic that is forwarded through the bridged or DLSw switched networks. This can be done in two ways:

- Filter as many broadcast NetBIOS frames as possible before bridging or DLSw switching them.
- Forward unfiltered NetBIOS UI frames on as few bridge ports or DLSw TCP sessions as possible.

Table 9-1 lists the filters that IBM provides.

Table 9-1. NetBIOS Filters

Filter Type	Filters
MAC Address	Frames by either the source or destination MAC address.
Byte	Frames by byte offset and field length within a frame.
Name	Frames by NetBIOS source and destination names.
Duplicate Frame	Duplicate frames.
Response	Responses for which the router did not forward a NetBIOS broadcast frame.

Once the router filters frames, NetBIOS name lists and NetBIOS name caching and route caching controls how the remaining frames are forwarded. “NetBIOS Byte Filtering” on page 3-5 and “NetBIOS Host Name Filtering” on page 3-5 in this manual describe byte and name filtering. The *Software User’s Guide* describes MAC address filtering.

An introduction to host-name filtering and byte filtering can be found in “NetBIOS Name and Byte Filters” on page 3-4.

The next sections describe frame type, duplicate frame, and response frame filtering, as well as NetBIOS name lists and NetBIOS name and route caching.

Frame Type Filtering

Frame type filtering allows certain categories of NetBIOS frames to be filtered entirely for either DLSw traffic, bridge traffic, or both DLSw and bridge traffic.

The three categories of NetBIOS frames that can be filtered are:

- Name Conflict Resolution frames

  These are the broadcast NetBIOS frames used to make sure that a NetBIOS name to be used is unique in the network.

  In NetBIOS networks, it is critical that the NetBIOS names of stations to which a NetBIOS session is established (typically the NetBIOS servers) be unique. It is also usually critical that the individual NetBIOS names of stations within the same group (or domain) be unique. But it is often not critical that the NetBIOS names of stations from which a NetBIOS session is setup (typically the NetBIOS clients) be unique, especially across domains.

  For this reason, networks in which there is good control over the server names can gain advantage by filtering name conflict resolution frames. This is especially true for DLSw switched networks.

  The NetBIOS name-conflict resolution frames are Add-Name-Query, Add-Group-Name-Query, and Add-Name-Response.

- General Broadcast frames

  This is the broadcast NetBIOS frame used to send data to all NetBIOS stations in a network. This frame is rarely used and can typically be filtered.

  The NetBIOS General Broadcast frame is Datagram-Broadcast.

- Terminate Trace frames

  These are the broadcast NetBIOS frames used to terminate NetBIOS traces in all NetBIOS stations in a network. These frames are rarely used and can typically be filtered.

  The NetBIOS Terminate Trace frame is Terminate-Trace.

The default is to not filter any of the above frame types for bridged NetBIOS traffic, and to filter all of the above frame types for DLSw switched NetBIOS traffic. However, it may be advantageous to filter the above frame types if NetBIOS traffic is being bridged on WAN links.

For bridging, enter **set filters bridge** to turn frame type filtering on or off. For DLSw, enter **set filters dlsw** to turn frame type filtering on or off. For example:

```
NetBIOS config>set filters bridge

Filter Name Conflict frames? [Yes]:

Name conflict filtering is            ON

Filter General Broadcast frames? [Yes]:

General broadcast filtering is        ON

Filter Trace Control frames? [Yes]:

Trace control filtering is            ON
```

Duplicate Frame Filtering

All of the broadcast NetBIOS frames which could have a response are sent a fixed number of times (default 6), at a fixed interval (default 1/2 second apart) by the origin NetBIOS station. We will call these frames “NetBIOS command frames” and we will call the possible response frames “NetBIOS response frames.”

The NetBIOS command frames are the:

- Name conflict resolution frames – Add-Name-Query and Add-Group-Name-Query
- NetBIOS session setup frames – Name-Query
- NetBIOS status frames – Status-Query

The command frames are sent multiple times to increase the odds of successful delivery (these frames are connectionless frames). Each response frame is sent only once in response to each command frame received.

In a DLSw-switched network, the forwarding of each retry across the WAN sessions can be very costly. So, when the first command frame is received, it is forwarded to the appropriate neighbor DLSw and bridge ports and a copy is saved. All retries of the same NetBIOS command frame received during a configurable time period are discarded.

There is one configurable time period for the bridge network and one configurable time period for the DLSw network. The configurable time period for the bridge network is controlled by two commands:

- **enable duplicate-filtering / disable duplicate-filtering** – which controls whether duplicate NetBIOS command frames are filtered on the bridge network at all.
- **set general** (“Duplicate frame filter timeout value in seconds” parameter)

  If duplicate frame filtering is enabled for the bridge network, this value specifies for how long a period to discard duplicate NetBIOS command frames after a NetBIOS command frame has been bridged.

  If a duplicate NetBIOS command frame is received after the timeout expires, the frame is forwarded to the bridge network.

The configurable time period for the DLSw network is controlled by a single parameter:

- **set cache-parms** (“Reduced search timeout value in seconds” parameter)

  This value specifies for how long a period to discard duplicate NetBIOS command frames after a NetBIOS command frame has been forwarded to the DLSw network.

  If a duplicate NetBIOS command frame is received after the timeout expires, the frame is forwarded to the DLSw network.

  Note: Filtering of duplicate NetBIOS command frames to a DLSw network is always enabled.

When a NetBIOS command frame is received by a DLSw neighbor, the frame is forwarded to the bridge network and a copy is saved. At a configurable interval (1/2 second) for a configurable number of times (default 6), the neighbor DLSw function forwards a retry of the command frame to the bridge function. The bridge function handles the command frame based upon the configured bridge duplicate frame parameters.

The configurable number of retries and interval are controlled by the following command and parameters:

- **set general** ("Command frame retry count" and "Command frame retry timeout value in seconds" parameters)

There is one last parameter that controls how long the command frame is saved in order to perform the above bridge and DLSw network forwarding:

- **set general** ("Duplicate frame detect timeout value in seconds" parameter)

  This parameter indicates how long a received NetBIOS command frame is saved for duplicate frame and response frame processing. After the timeout expires, the command frame is deleted and the duplicate frame filter timer and reduced search timer associated with it are cancelled. The first duplicate command frame received after the timeout period is treated as the first command frame received. All response frames received after the timeout period are discarded.

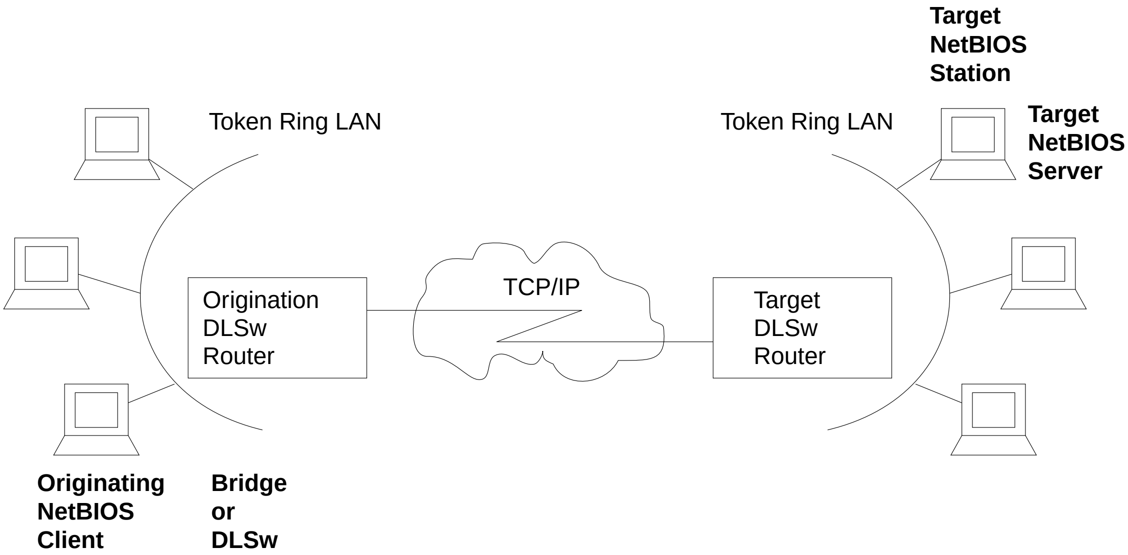
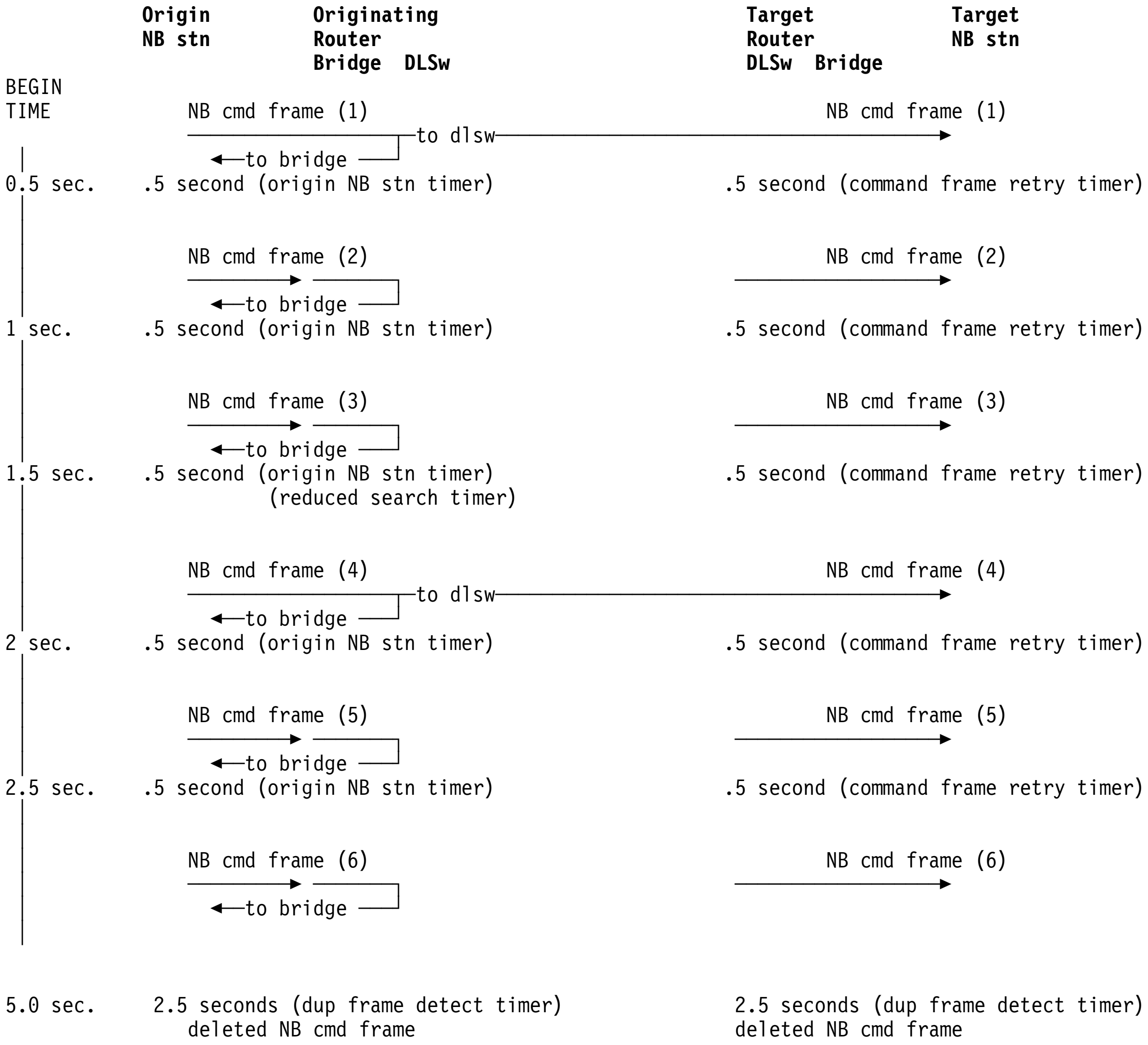


Figure 9-1. Setting Up a NetBIOS Session Over DLSw. Duplicate filtering reduces the number of broadcast frames forwarded over the DLSw WAN.

Figure 9-1, together with the following sequence, is an example using all of the defaults describe the process. To simplify the example, it is assumed that no response frame is received.


Origin NB stn
Originating Router Bridge DLSw
Target Router DLSw Bridge
Target NB stn
BEGIN TIME
NB cmd frame (1)
to dlsw
NB cmd frame (1)
to bridge
0.5 sec.
.5 second (origin NB stn timer)
.5 second (command frame retry timer)
NB cmd frame (2)
NB cmd frame (2)
to bridge
1 sec.
.5 second (origin NB stn timer)
.5 second (command frame retry timer)
NB cmd frame (3)
NB cmd frame (3)
to bridge
1.5 sec.
.5 second (origin NB stn timer)
(reduced search timer)
.5 second (command frame retry timer)
NB cmd frame (4)
to dlsw
NB cmd frame (4)
to bridge
2 sec.
.5 second (origin NB stn timer)
.5 second (command frame retry timer)
NB cmd frame (5)
NB cmd frame (5)
to bridge
2.5 sec.
.5 second (origin NB stn timer)
.5 second (command frame retry timer)
NB cmd frame (6)
NB cmd frame (6)
to bridge
5.0 sec.
2.5 seconds (dup frame detect timer)
deleted NB cmd frame
2.5 seconds (dup frame detect timer)
deleted NB cmd frame

The sequence of events is as follows:

1. The first NetBIOS command frame is received on a bridge port at the origin DLSw router. A copy of the NetBIOS command frame is saved. Because bridging is enabled, the frame is forwarded onto the bridge network. Because duplicate-filtering on the bridge network is disabled as the default, the duplicate frame filter timer is not started. Because DLSw NetBIOS is enabled, the frame is forwarded onto the DLSw network and the reduced search timer is started (default 1-1/2 seconds). The duplicate frame detect timer (default 5 seconds) is also started.

2. The target router DLSw function receives the first NetBIOS command frame. A copy of the NetBIOS command frame is saved. Because bridging is enabled, the frame is forwarded onto the bridge network. Because duplicate-filtering on the bridge network is disabled as the default, the duplicate frame filter timer is not started. The retry command timer (default 1/2 second) and the duplicate frame detect timer (default 5 seconds) are started.

3. At the origin router, the second NetBIOS command frame (first retry) is received. Because duplicate-filtering on the bridge network is disabled as the default, the frame is forwarded onto the bridge network. Because the reduced

search timeout has not expired, the frame is not forwarded onto the DLSw network.

4. At the target router, the DLSw function forwards a first retry of the NetBIOS command frame (generated locally) to the bridge function. Because duplicate-filtering on the bridge network is disabled as the default, the frame is forwarded onto the bridge network. The retry command timer (default 1/2 second) is started.
5. At the origin router, the third NetBIOS command frame (second retry) is handled in the same manner as the second NetBIOS command frame.
6. At the target router, the second retry of the NetBIOS command frame is handled in the same manner as the first retry.
7. At the origin router, the fourth NetBIOS command frame (third retry) is received. Because duplicate-filtering on the bridge network is disabled as the default, the frame is forwarded onto the bridge network. Because the reduced search timeout has now expired, the frame is forwarded onto the DLSw network. The reduced search timer is restarted.
8. At the target router, the DLSw function forwards a third retry of the NetBIOS command frame (generated locally) to the bridge function. Because duplicate-filtering on the bridge network is disabled as the default, the frame is forwarded onto the bridge network. The retry command timer (default 1/2 second is started). The target router also receives the forwarded NetBIOS command frame from the origin router, but discards it as a duplicate.
9. At the origin router, the fifth NetBIOS command frame (fourth retry) is handled in the same manner as the second NetBIOS command frame.
10. At the target router, the fourth retry of the NetBIOS command frame is handled in the same manner as the first retry.
11. At the origin router, the sixth NetBIOS command frame (fifth retry) is received. Because duplicate-filtering on the bridge network is disabled as the default, the frame is forwarded onto the bridge network. Because the reduced search timeout has not expired, the frame is not forwarded onto the DLSw network.
12. At the target router, the DLSw function forwards a fifth retry of the NetBIOS command frame (generated locally) to the bridge function. Because duplicate-filtering on the bridge network is disabled as the default, the frame is forwarded onto the bridge network. Because the retry count is now exhausted, the command retry timer is not restarted.
13. After 2-1/2 more seconds at the origin router, the duplicate frame detect timer expires and the saved NetBIOS command frame is deleted.
14. After 2-1/2 more seconds at the target router, the duplicate frame detect timer expires and the saved NetBIOS command frame is deleted.

Response Frame Filtering

The NetBIOS session setup command frame and the NetBIOS status command frame each expect a corresponding NetBIOS response frame. If no response frame is received, the command frame is retried as in the example above.

When the first NetBIOS response frame is received on the bridge network at the target router, it is forwarded back to the origin router and the saved NetBIOS command frame is deleted. Any subsequent response frame received at the target router is discarded because no corresponding NetBIOS command frame is found.

At the origin router, the received response frame is forwarded on the bridge network and the saved NetBIOS command frame is deleted. Any subsequent response frames received at the origin router (from the DLSw or bridge network) are discarded.

The NetBIOS name conflict command frames may cause, but do not require, a corresponding NetBIOS response frame. In addition, all received response frames are used (to determine whether there is more than one conflict).

Therefore, all NetBIOS name conflict frames received are forwarded, but the NetBIOS command frame is not deleted until the Duplicate Frame Detect timer expires.

NetBIOS Name Lists

NetBIOS name lists is a DLSw-only vehicle for limiting the number of DLSw partners to which a NetBIOS UI frame is forwarded.

A local NetBIOS name list can be configured at each router. This name list represents all of the NetBIOS names attached to the router’s locally bridged network that can be accessed by DLSw partners. The router sends the local NetBIOS name list to all DLSw partners. These partners use the list to limit the NetBIOS traffic the partner sends to this router.

The NetBIOS name lists are useful in environments in which there is good control over the NetBIOS names; particularly those environments that should be accessed remotely through DLSw.

Configuring Local NetBIOS Name Lists

A NetBIOS name list is a set of NetBIOS name list entries. Configuring of the local NetBIOS name list involves:

- Adding up to 30 entries into a name list
- Configuring whether this list represents all of the NetBIOS names reachable by the router’s DLSw partners.

You configure the name list entries at the `NetBIOS config>` prompt with the *add name-list* command. Each entry consists of the following information:

name qualifier

A name qualifier represents one or more NetBIOS names. Each name qualifier may be up to 16 characters. You can represent multiple NetBIOS names by using wildcards (either an imbedded ? or a trailing *) within the name.

The ? (question mark) signifies that the character in that position in the NetBIOS name may have any value.

The * (asterisk) as the last character of a name to signifies that all of the remaining characters in the NetBIOS name may be any value.

Note: In the majority of client/server NetBIOS applications, the only names required in the name lists are those of servers or domains. Individual client names do not need to be configured in name lists.

name qualifier type

NetBIOS names can be individual names or group names. Each name qualifier represents either a set of individual NetBIOS names or a set of group NetBIOS names. The name qualifier type specifies which type of NetBIOS names (individual or group) the corresponding name qualifier represents.

As a general rule, domain names are group names and client or server names are individual names.

The name list itself has an attribute that is configured at the `NetBIOS config>` prompt using the SET NAME-LIST command. That attribute is *name list exclusivity.*

The attribute indicates whether the set of name list entries represents all NetBIOS names that this router’s DLSw partners can reach (exclusive) or represents some but not necessarily all NetBIOS names that this router’s DLSw partners can reach (non-exclusive).

An exclusive name list does the best job of limiting NetBIOS DLSw traffic on the network. Only frames destined to a NetBIOS name represented by a router’s exclusive name list are forwarded to that router.

A non-exclusive name list helps limit NetBIOS DLSw traffic on the network though not as well as an exclusive name list. Frames destined to a NetBIOS name represented by a router’s non-exclusive name list will be forwarded to that router first.

If the router receives a frame destined to a NetBIOS name not represented by any router’s name lists, the router forwards the frame to all routers with non-exclusive name lists.

It is possible to control how a particular router uses its local NetBIOS name list and the name lists received from its DLSw partners using the following parameters:

use local NetBIOS name list

This function is configured with the **enable name-list local** or **disable name-list local** command at the `NetBIOS config>` prompt.

If you enable use local NetBIOS name list, the router sends the local NetBIOS name list configured at the router to all DLSw partners.

If you disable use local NetBIOS name list, the router does not send the local NetBIOS name list configured at the router to all DLSw partners.

use remote NetBIOS name lists

This function is configured with the **enable name-list remote** or **disable name-list remote** command at the `NetBIOS config>` prompt.

If you enable use remote NetBIOS name lists, the router uses all NetBIOS names lists received from the router’s DLSw partners to determine how to forward certain NetBIOS frames.

If you disable use remote NetBIOS name lists, the router ignores all NetBIOS name lists received from the router’s DLSw partners.

Committing NetBIOS Name List Changes

You can change all the NetBIOS name list parameters either permanently at the `NetBIOS config>` prompt or temporarily at the `NetBIOS>` prompt.

Because each change made requires the router to send information to each DLSw partner, you must indicate that the name list changes are ready for use by entering **set name-list** at the `NetBIOS>` command prompt.

Using NetBIOS Name Lists

The router uses NetBIOS name lists to determine how to forward the following NetBIOS frames:

- NetBIOS session setup command frame (Name-Query)
- NetBIOS status command frame (Status-Query)
- NetBIOS connectionless data transfer frame (Datagram)

Using Exclusive NetBIOS Name Lists Effectively: Configure exclusive NetBIOS name lists whenever possible. If you configure and send an exclusive name list to all DLSw partners, then the only NetBIOS frames received from the DLSw partners will be the frames whose destination name matches one of the name list entries.

A useful exclusive NetBIOS name list is the empty NetBIOS name list. If a particular router has no NetBIOS servers that are to be accessed by any of its DLSw partners, you should use an empty exclusive name list.

Using Non-Exclusive NetBIOS Name Lists: If a router has many DLSw partners all on different bridged networks, you can use non-exclusive name lists. Name list entries could be configured for the most frequently used servers so that traffic destined for these servers would go to this router first. Specifying the name list as non-exclusive allows traffic to go to less frequently used servers without having to configure the servers in the name list. Use this configuration in a network without tight control of the NetBIOS names; particularly the servers to be accessed remotely through DLSw.

Another use of non-exclusive NetBIOS name lists is in configurations that contain parallel DLSw paths between bridged networks. If two routers are on the same bridged network, one router could configure a NetBIOS name list representing one set of servers to be accessed remotely through DLSw on the bridged network and the other router could configure a NetBIOS name list representing a different set of servers. When both routers are active, the NetBIOS traffic is distributed between the two routers. If one router is inactive, all NetBIOS traffic will go through the other router because it has a non-exclusive list.

The default name list is an empty non-exclusive NetBIOS name list. This indicates that a router wants its DLSw partners to send all unforwardable NetBIOS traffic to the router.

NetBIOS Name Caching and Route Caching

NetBIOS Name Caching is the function in the router that classifies the type of NetBIOS name and the information necessary to reach the NetBIOS name. This information is used to best determine how to forward unfiltered NetBIOS frames to as few DLSw neighbors and as few bridge ports as possible. The possible types of NetBIOS names and the information saved for each are:

Individual remote
This is a NetBIOS name known to be reachable remotely via a particular DLSw TCP session. The best TCP sessions are saved. the bridge network. The MAC address associated with the name is saved. If route caching is enabled, the best LLC route between the router and the NetBIOS station is also saved.

Individual local
This is a NetBIOS name known to be reachable locally via the bridge network. The MAC address associated with the name is saved. If route caching is enabled, the best LLC route between the router and the NetBIOS station is also saved.

Group This is a NetBIOS name known to be a group name. It may be reachable remotely and/or locally and may represent multiple NetBIOS stations. No other information is saved.

Unknown Information about the NetBIOS name is not yet known, indicating that a search for the name is not complete. No other information is saved.

Whenever NetBIOS session setup frames or connectionless data transfer frames are received, the name cache is used to determine how to forward the frame. If one of these frames is received on the bridge network at a router, one of the following actions is taken:

- If the destination name in the NetBIOS frame is not in the router’s NetBIOS name cache, then all DLSw partner names lists are searched for a match.

  If matches with group name qualifiers are found, a NetBIOS name cache entry is created with the name type *group.* The frame is forwarded on all bridge ports and to all DLSw partners with non-exclusive name lists or exclusive name lists with a matching name list entry.

  If matches with individual name qualifiers are found, a NetBIOS name cache entry is created with the name type *individual remote.* The frame is forwarded to each DLSw partner with a matching name list entry.

  If no matches are found, a NetBIOS name cache entry is created with the name type *unknown.* The frame is forwarded on all bridge ports and to all DLSw partners with non-exclusive name lists.

- If the destination name in the NetBIOS frame is in the router’s NetBIOS name cache and is classified as individual remote, then the frame is forwarded on the learned best DLSw TCP session.

  If multiple equally best TCP sessions are learned, they will be used alternately on different NetBIOS session setup frames.

- If the destination name in the NetBIOS frame is in the router’s NetBIOS name cache and is classified as individual local, then the saved MAC address will replace the NetBIOS frame’s destination MAC address.

If route caching is disabled, the NetBIOS frame's routing information is left alone, and the frame is forwarded to all bridge ports.

If route caching is enabled, the NetBIOS frame's routing information is updated with the saved routing information and the frame is forwarded to the proper bridge port (determined by the MAC address and route).

- If the destination name in the NetBIOS frame is in the router's NetBIOS name cache and is classified as group or unknown, the frame is forwarded on all bridge ports and to all DLSw neighbors.

Learning NetBIOS Names

NetBIOS names are learned and classified from information in the NetBIOS session setup frames (Name-Query and Name-Recognized).

Configuring NetBIOS Name Cache Entries

It is possible to configure individual remote NetBIOS names and associate them with a particular DLSw TCP session. This can greatly reduce the search overhead. To improve performance, it is recommended to configure the remote NetBIOS servers that are accessed commonly by NetBIOS clients in the router's local bridge network.

It is not possible to configure individual local NetBIOS names and associate them with a particular MAC address and route.

There are three types of NetBIOS name cache entries:

- Permanent entries are those that are added at the NetBIOS configuration prompt (`NetBIOS config>`). The router saves permanent entries in its configuration when the router is restarted.

  Enter **add cache-entry** at the `NetBIOS config>` prompt to add a permanent entry. You are prompted to enter the NetBIOS name and the associated IP address.

- Static entries are those that are added at the NetBIOS monitoring prompt (`NetBIOS>` console). The router does not save static entries when the router is restarted.

  Enter **add cache-entry** at the `NetBIOS>` console prompt to add a static entry. You are prompted to enter the NetBIOS name and the associated IP address.

- Dynamic entries are those that are ***not*** added at the NetBIOS configuration or monitoring prompts, but are learned dynamically from the NetBIOS session setup frames. The router does not save dynamic entries when the router is restarted.

Configuring Name Cache Parameters

To prevent one type of NetBIOS name from filling up the entire name cache, there are two configurable NetBIOS name cache limits:

- Maximum number of local name cache entries specifies the maximum number of individual local NetBIOS name cache entries that can be cached at one time. Least recently used entries are overridden by new entries.

- Maximum number of remote name cache entries specifies the maximum combined number of individual remote, group, and unknown NetBIOS name cache entries that can be cached at one time. Least recently used entries are overridden by new entries.

If an entry is not referenced for a configurable timeout period, then it is automatically deleted. This timeout out period is the unreferenced entry timeout value.

The association of a NetBIOS name with either a TCP session or a MAC address and route is made at one instance in time. Because networks are changing and the best path to a NetBIOS name may change, the association between a NetBIOS name and a TCP session or a MAC address and route is saved for only a configurable period of time. After this period of time, a new best path association is learned. The parameter that controls this configurable period of time is the best path aging timeout value.

Another useful configuration parameter is the reduced search timeout value. In addition to controlling for what period of time duplicate command frames are filtered to the DLSw network, it also controls how long to wait before expanding the search for a NetBIOS name. If a NetBIOS session setup frame is received and the destination NetBIOS name is found in the router's NetBIOS name cache as an individual remote frame, then the frame is forwarded to the corresponding TCP session. If no response to this frame is received, it could be due to the name no longer being accessible via this path. The first duplicate NetBIOS session setup frame received after the reduced search timer expires is forwarded to all DLSw TCP sessions, thus expanding the search to look for a better path.

The last parameter, significant characters in name, controls how many of the 16 characters in a NetBIOS name are needed to consider it a unique NetBIOS name. Some NetBIOS applications use the 16th character of the NetBIOS name to distinguish between certain entities associated with a single NetBIOS name (for example, print server and file server). In these cases, it is best to specify significant characters in name as 15. This causes any frame in which the first 15 characters of the destination NetBIOS name matches the first 15 characters of the router's NetBIOS name cache entry to be forwarded according to the name cache entry information. Thus multiple NetBIOS names can be represented with a single NetBIOS name cache entry.

All of the above NetBIOS name cache related parameters can be configured using the **set cache-parms** command as follows.

```
NetBIOS config>set cache-parms

Significant characters in name [15]?
Best path aging timeout value in seconds [60.0]?
Reduced search timeout value in seconds [1.5]?
Unreferenced entry timeout value in minutes [5000]?
Max nbr local name cache entries [500]?
Max nbr remote name cache entries [100]?

Cache parameters set
```

See "NetBIOS Commands" on page 9-24 for more information on the **set cache-parms** command.

Displaying Cache Entries

The router provides the following commands that let you view cache entries. From the NetBIOS configuration prompt, you can use the **list cache** commands in Table 9-2.

Table 9-2. NetBIOS List Cache Configuration Commands

Command	Displays . . .
list cache all	All permanent entries. Does not show static and dynamic entries.
list cache entry-number	A permanent cache entry according to its entry number.
list cache netbios-name	A permanent cache entry for a specific NetBIOS name.
list cache ip-address	A permanent cache entry for a specific IP address.

From the NetBIOS monitoring prompt, you can use the list cache commands in Table 9-3.

Table 9-3. NetBIOS List Cache Monitoring Commands

Command	Displays . . .
list cache active	All active entries in the router's name cache, including permanent, static, and dynamic entries.
list cache config	Static and permanent entries. Does not show dynamic entries.
list cache group	Entries that exist for NetBIOS group names.
list cache local	Local cache entries. Local cache entries are those that the router learns over the bridged network.
list cache name	A cache entry for a specific NetBIOS name.
list cache remote	Remote cache entries. These are entries that the router learns over the DLSw WAN.
list cache unknown	Entries where the type of NetBIOS entry is unknown. The router considers all entries unknown until it learns the type of entry.

NetBIOS Host Name and Byte Filtering Configuration Procedures

The following sections provide examples of how to set up NetBIOS filtering. The first explains how to create a host name filter. The second demonstrates how to configure a byte filter. For more information on the commands used in these examples, see "NetBIOS Commands" on page 9-24.

To create a host name filter, enter commands at the `NetBIOS Filter config>` prompt.

```
Config>protocol asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>netbios

NetBIOS Support User Configuration

NetBIOS config>set filter name
NETBIOS Filtering configuration
NETBIOS Filter config>
```

Creating a Host Name Filter

Use the following procedure as a guideline to creating a host name filter.

1. Create an empty name filter list.

   ```
   NetBIOS Filter config>create name-filter-list
   Handle for Name Filter List []? boston
   ```

2. Add the filter items to the name filter list.

   Enter **update** to get to the prompt for that specific filter list. From this prompt, you can add filter items to the filter list.

   ```
   NetBIOS Filter config>update
   Handle for Filter List []? boston
   Name Filter List Configuration
   NetBIOS Name boston config>
   ```

3. Add filter items to the filter list with the **add** command. The way filter items are configured determines which NetBIOS packets are bridged or dropped. Host name filter items are configured with the following parameters entered in this order:

   - *Inclusive* (bridged) or *Exclusive* (dropped).
   - *ASCII* or *HEX* - how the hostname is represented.
   - *host name* - the actual host name represented in either an ASCII or hex string (see the command section that follows for syntax). This entry is case sensitive.
   - <*LAST-hex-number*> - an optional parameter for use with ASCII strings containing fewer than 16 characters.

   The following example adds a filter item to the Host Name Filter list **boston**, which allows packets containing the hostname **westboro** (an ASCII string) to be bridged (configured as *inclusive*). No <*LAST-hex-number*> parameter has been configured for this entry.

   ```
     NetBIOS Name boston config>add inclusive ascii
     Hostname []? westboro
     Special 16th character in ASCII hex (<CR> for no special char) []?
   ```

   You can enter all parameters as one string on the command line if you do not want to be prompted. Be sure to use a space between each parameter.

4. Verify the filter item entry.

   Type **list** to verify your entry:

   ```
   NetBIOS Name boston config>list

   NAME Filter List Name: boston
   NAME Filter List Default: Inclusive

    Item #   Type    Inc/Ex    Hostname         Last Char

      1      ASCII    Inc      westboro
   ```

5. Add additional filter items to the filter list.

   Repeat the first four steps to add additional filter items to the filter list. The order in which you enter filter items is important as this determines how the router applies the filter items to a packet. The first match stops the application of filter items and the router either forwards or drops the packet, depending on whether the filter item is Inclusive or Exclusive.

Entering the most common filter items first makes the filtering process more efficient because the software is more likely to make a match at the beginning of the list.

If the packet does not match any of the filter items, the router uses the default condition (Inclusive or Exclusive) of the filter list. You can change the default condition of the list by entering **default inclusive** or **default exclusive** at the filter list configuration prompt. For example:

```
NETBIOS Name boston config> default exclusive
```

6. When you have finished adding filter items to the filter list, enter **exit** to return to the `NetBIOS Filter config>` prompt.

```
NetBIOS Name boston config>exit
NetBIOS Filter config>
```

7. Add the filter to your configuration.

   The filter list containing the filter items can now be added as a filter to your bridging router configuration. Use the **filter-on** command to do this. Host name filters are configured with the following parameters (entered in this order):

   - *Input* (to filter all NetBIOS packets received on that port) or output (to filter all NetBIOS packets transmitted on that port).
   - *Port#*, which is the desired configured bridge port number on the router.
   - *Filter-list*, which is the name of the filter list (containing filter items) that you want to be included in this filter.
   - An optional operator entered as either AND or OR in all capital letters. If an operator is present, it must be followed by a filter-list name. Filters with more than one filter list are called complex filters.

   The following example adds a host name filter to affect packets input on port #3. It is comprised of the host name filter list **boston**. All packets input on port #3 are evaluated according to the rules provided by the filter items contained in the filter list **boston**. This means that all packets input on port #3 containing the hostname **westboro** are bridged.

```
NetBIOS Filter config>filter-on input
Port Number [1]? 3
Filter List []? boston
```

8. Verify the newly created filter.

   Enter **list** to verify your entry:

```
NetBIOS Filter config>list

NetBIOS Filtering: Disabled

NetBIOS Filter Lists
--------------------

    Handle           Type
    nlist            Name
    newyork          Name
    HELLO            Byte
    boston           Name

NetBIOS Filters
---------------

    Port #      Direction      Filter List Handle(s)
       3          Output       nlist
       1          Input        newyork OR HELLO
       3          Input        boston
```

9. Globally enable NetBIOS filtering.

   Use the **enable** command to globally enable NetBIOS filtering on the router.

   ```
   NetBIOS Filter config>enable Netbios-filtering
   ```

10. Restart the router to activate all NetBIOS filtering configuration changes.

    Enter **exit** followed by Ctrl P to return to the * prompt. From this prompt, enter **restart** to activate all software changes made during the NetBIOS filtering configuration process.

    ```
    NetBIOS Filter config>exit
    ASRT config>exit
    Config> Ctrl P
    * restart
    ```

Creating a Byte Filter

Use the following procedure as a guideline for creating a byte filter. Enter all commands at the `NetBIOS filtering config>` prompt.

```
Config>protocol asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>NetBIOS

NetBIOS Support User Configuration

NetBIOS config> set filter byte
NetBIOS Filtering configuration
NetBIOS Filter config>
```

1. Create an empty filter list using the **create byte-filter-list** command.

   ```
   NetBIOS Filter config>create byte-filter-list
   Handle for Byte Filter List []? westport
   ```

2. Add the filter items to the byte filter list.

   Enter **update** to get to the prompt for that specific filter list. From this prompt you can add filter items to the filter list.

   ```
   NetBIOS Filter config>update
   Handle for Filter List []? westport
   Byte Filter List Configuration
   NetBIOS Byte westport config>
   ```

   Begin adding filter items to the filter list with the **add** command. The way filter items are configured determines which NetBIOS packets are bridged or dropped. Byte filter items are configured with the following parameters (entered in this order):

   - Inclusive (bridged) or Exclusive (dropped).
   - Byte Offset - the number of bytes (in decimal) to offset into the packet being filtered. This starts at the NetBIOS header of the packet. Zero specifies that the router will examine all bytes in the packet.
   - Hex pattern - a hexadecimal number used to compare with the bytes starting at the byte offset of the NetBIOS header. See “NetBIOS Commands” on page 9-24 for syntax rules.
   - Hex mask - (if present) must be the same length as hex pattern and is logically ANDed with the bytes in the packet starting at byte-offset before the result is compared for equality with hex pattern. If the *hex-mask* argument is omitted, it is considered to be all binary ones.

The following example adds a filter item to the Byte filter list **westboro** that allows packets with a hex pattern 0x12345678 at byte offset of 0 to be bridged (configured as inclusive). No hex mask is present.

```
NetBIOS Byte westport config>add inclusive
Byte Offset [0]? 0
Hex Pattern []? 12345678
Hex Mask (<CR> for no mask) [[?
```

3. Verify the filter item entry with the **list** command.

```
NetBIOS Byte westport config>list

BYTE Filter List Name: westport
BYTE Filter List Default: Inclusive

 Item #   Inc/Ex   Offset  Pattern        Mask

   1       Inc        0    0x12345678     0xFFFFFFFF
```

4. Add additional filter items to the filter list.

   Repeat the first three steps to add additional filter items to the filter list.

5. When you have finished adding filter items to the filter list, type **exit** to return to the `NetBIOS Filter config>` prompt.

```
NetBIOS Byte westport config>exit
NetBIOS Filter config>
```

   The order in which you enter filter items is important, as this determines how the router applies the filter to a packet. The first match stops the application of filter items and the router either forwards or drops the packet, depending on whether the filter item is Inclusive or Exclusive.

   Entering the most common filter items first makes the filtering process more efficient because the software is more likely to make a match at the beginning of the list rather than having to check the whole list before making a match.

   If the packet does not match any of the filter items, the router uses the default condition (Inclusive or Exclusive) of the filter list. You can change the default condition of the list by entering **default inclusive** or **default exclusive** at the filter list configuration prompt. For example:

```
NETBIOS Byte westport config> default exclusive
```

6. Add the filter to your configuration.

   The filter list containing the filter items can now be added as a filter to your bridging router configuration. Use the **filter-on** command to do this. Host name filters are configured with the following parameters (entered in this order):

   - *Input* (to filter all packets received on that port) or output (to filter all packets transmitted on that port).
   - *Port#* - the configured bridge port number.
   - *Filter-list* - the name of the filter list (containing filter items) that you want included in this filter,
   - An optional operator entered as either AND or OR. The operator is entered in all capital letters. If an operator is present, it must be followed by a filter-list name. Filters with more than one filter list are called complex filters. These are explained in more detail in “About NetBIOS Configuration and Monitoring Commands” on page 9-20.

   The following example adds a host name filter to affect packets output on port #3. It is comprised of the byte filter list **westboro**. All packets output on port

#3 will be evaluated according to the rules provided by the filter items contained in the filter list **westboro**.

```
NetBIOS Filter config>filter-on output
Port Number [1]? 3
Filter List []? westboro
```

7. Verify the newly created filter.

   Enter **list** to verify your entry:

```
NetBIOS Filter config>list

NetBIOS Filtering: Disabled

NetBIOS Filter Lists
-------------------

    Handle           Type
    nlist            Name
    newyork          Name
    HELLO            Byte
    westboro         Byte

NetBIOS Filters
---------------

    Port #      Direction      Filter List Handle(s)
       3          Output       nlist
       1          Input        newyork OR HELLO
       3          Output       westboro
```

8. Globally enable NetBIOS filtering.

   Enter **enable** to globally enable NetBIOS filtering on the bridging router.

```
NetBIOS Filter config>enable netbios-filtering
```

9. Restart the router to activate all NetBIOS filtering configuration changes.

   Enter **exit** followed by Ctrl P to return to the * prompt. Enter **restart**.

```
NetBIOS Filter config>exit
ASRT config>exit
Config> Ctrl P
* restart
```

About NetBIOS Configuration and Monitoring Commands

NetBIOS configuration commands are available at the `ASRT/DLSW NetBIOS config>` prompt. Changes you make to the router's configuration do not take effect immediately. They become part of the router's configuration memory when you restart it. This chapter refers to configuration changes as permanent.

NetBIOS monitoring commands are available at the `ASRT/DLSW NetBIOS>` prompt. Monitoring commands take effect immediately, but are not saved in the router's non-volatile configuration memory. Thus, while monitoring commands allow you to make real-time changes to the router's configuration, these changes are temporary. The router's configuration memory overwrites them when the router restarts. This chapter refers to changes you make at the monitoring prompt as static.

Accessing the NetBIOS Configuration Environment

You can display the `NetBIOS config>` prompt from either the ASRT configuration environment or the DLSw configuration environment. Changes you make at the `NetBIOS config>` prompt affect both bridging and DLSw.

To display the `NetBIOS config>` prompt from the ASRT configuration environment:

```
Config>protocol asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>netbios

NetBIOS Support User Configuration

NetBIOS config>
```

To display the `NetBIOS config>` prompt from the DLSw configuration environment:

```
Config>protocol dls
DLSw protocol user configuration
DLSw config>netbios

NetBIOS Support User Configuration

NetBIOS config>
```

Accessing the NetBIOS Console Environment

You can display the `NetBIOS>` prompt from either the ASRT monitoring environment or the DLSw monitoring environment.

Changes you make at the `NetBIOS>` console prompt affect both bridging and DLSw.

To display the `NetBIOS>` console prompt from the ASRT monitoring environment:

```
+ protocol asrt
ASRT>netbios

NetBIOS Support User Console

NetBIOS>
```

To display the `NetBIOS>` prompt from the DLSw monitoring environment:

```
+ protocol dls
DLSw>netbios

NetBIOS Support User Console

NetBIOS>
```

Configuring NetBIOS for DLSw

If you are sending NetBIOS traffic over DLSw, use this procedure at the `DLSw config>` prompt:

- Open NetBIOS SAPs.
- Set a priority for SNA and NetBIOS sessions.
- Set the maximum NetBIOS frame size.
- Set the number of bytes to allocate for NetBIOS UI frames.

Open NetBIOS SAPs

Open NetBIOS SAPs on both sides of the link to enable DLSw to transmit NetBIOS frames.

```
DLSw config> open-sap
Interface # [0]?
Enter SAP in hex(range 0-F0),'SNA', or 'NB'[4]? nb
SAP F0 opened on interface 0
```

Set a Priority for SNA and NetBIOS Sessions

You can prioritize SNA and NetBIOS traffic to prevent one type of session from using up too much of the available bandwidth during network congestion. To do so, enter **priority** to set a priority for SNA sessions and NetBIOS sessions. You also set a message allocation that corresponds to a session's priority.

Use the **set priority** command as shown in the following example:

```
DLSw config> set priority
Priority for SNA DLSw sessions (C/H/M/L) [M]?  H
Priority for NetBIOS DLSw sessions (C/H/M/L) [M]? M
Message allocation by C/H/M/L priority (4 digits) [4/3/2/1]?
Maximum NetBIOS frame size (516, 1470, 2052, or 4399) [2052]?
```

The default message allocation of 4/3/2/1, provides the following allocation to sessions:

4 - Critical
3 - High
2 - Medium
1 - Low

The router uses the priority and message allocation to selectively limit the burst length of specific types of traffic. For example:

- If you assign SNA traffic a priority of Critical, and Critical sessions have a message allocation of 4

  and

- You assign NetBIOS traffic a priority of Medium, and Medium sessions have a message allocation of 2,

the router processes four SNA frames before it processes two NetBIOS frames. Once the router processes two NetBIOS frames, it processes four SNA frames, and so on.

In this scenario, the router dedicates two thirds of available bandwidth to SNA traffic (a ratio of 4 to 2). Note that the router counts frames, rather than bytes, when allocating bandwidth according to the priorities you assign.

You can change the message allocation for sessions from the default of 4/3/2/1. You must always enter four digits, from 9 to 1, in descending order. For example, if the SNA priority is Critical and the NetBIOS traffic is Medium, and you change the message allocation to 8/7/6/5, the router processes eight SNA frames before it processes six NetBIOS frames.

Set the Maximum NetBIOS Frame Size

You can also use the DLSw **set priority** command to change the maximum NetBIOS frame size. The default is 2052. Set this parameter to the largest frame size you expect to need, and no larger. Setting the frame size larger than needed reduces the number of available buffers.

Set the Memory Allocation for NetBIOS UI Frames

Use the DLSw **set memory** command to set the number of bytes the router allocates as a buffer for NetBIOS UI frames. If the TCP transmit buffer becomes full, the router uses this buffer for NetBIOS UI frames.

Note that the number of bytes allocated for NetBIOS is global, and not per session.

```
DLSw config> set memory
Number of bytes to allocate for DLSw (at least 26368) [141056]?
Number of bytes to allocate per LLC session [8192]?
Number of bytes to allocate per SDLC session [4096]?
Number of bytes to allocate for NetBIOS UI-frames [40960]?
```

NetBIOS Commands

Table 9-4 lists the NetBIOS configuration and monitoring commands.

Table 9-4. NetBIOS Configuration and Monitoring Commands

Command	Function
?(Help)	Lists available commands or options.
Add	Adds cache entries to the router’s name cache, and adds name list entries to the router’s local name list.
Delete	Deletes cache entries or name list entries that you added using the **add** command.
Disable	Disables duplicate frame filtering, route caching and the use of local and remote NetBIOS name lists.
Enable	Enables duplicate frame filtering, route caching and the use of local and remote NetBIOS name lists.
List	Displays various NetBIOS name cache and name list configuration information depending on whether you are at the configuration prompt or the monitoring prompt.
Set	Configures parameters for name caching, duplicate frame filtering, frame-type filtering, and name lists. Also displays the `NETBIOS Filter config>` prompt.
Test	This command is available only at the monitoring prompt and tests a particular NetBIOS name against the current NetBIOS name cache and name lists.
Exit	Returns to the previous prompt.

? (Help)

Lists available commands or options.

Syntax: ?

Example: **`set ?`**

```
CACHE-PARMS
FILTERS
GENERAL
NAME-LIST
```

Add

Adds a new name cache entry to the router’s permanent or static configuration, or adds a NetBIOS name list entry used to limit remote station access to local DLSWs. You can add name cache entries for DLSw neighbors only. The router ignores entries that you add for ASRT traffic.

Syntax: **add** cache-entry
name-list

cache-entry
Adds a new entry to the router’s name cache.

- From the configuration prompt, adds a permanent entry.
- From the monitoring prompt, adds a temporary entry.

The router prompts you for the 16th character in hex only if you have indicated via **set cache-parms** that 16 characters are relevant in a NetBIOS name.

Multiple entries with different IP addresses may be added for a single NetBIOS name. This allows the name to be accessed through multiple DLSw neighbors.

Note: The NetBIOS name is case sensitive and must match the case of the network NetBIOS name.

Example: **add cache-entry**

```
Enter up to 15 characters of NetBIOS name (no wild cards)[]Accounting
Enter last character of NetBIOS name in hex [0]? 01
Enter IP Address [0.0.0.0]? 20.2.1.3
Name cache entry has been created
```

name-list

Adds a new entry to the router’s local name list.

From the configuration prompt, adds a permanent name list entry. The change does not take effect until the router is restarted or the change is committed from the `NetBIOS>` prompt using **set name-list** command.

From the monitoring prompt, adds a temporary name list entry. The change does not take effect until the change is committed from the `NetBIOS>` prompt using **set name-list** command. The change is lost when you restart the router.

The NetBIOS name qualifier represents one or more NetBIOS names reachable on this router’s locally bridged network that are to be made reachable to other routers through DLSw.

The NetBIOS name qualifier may contain the following two types of wildcard characters:

? (question mark) Indicates that a single character in a real NetBIOS name can be any value.

*** (asterisk)** At the end of a name qualifier indicates that the remaining characters in a real NetBIOS name can be any value.

Notes:

1. If an asterisk does not appear at the end of a name qualifier, the remainder of the name qualifier up to the maximum of 16 characters is padded with nulls (hex zeroes).
2. The NetBIOS name qualifier is case sensitive and must match the case of the network NetBIOS names.

Example: **add name-list**

```
Enter up to 16 characters of NetBIOS name qualifier (wild cards OK).
  Enter name qualifier []? NY_SERV*
NetBIOS name qualifier type (I=individual, G=group) [I]?

Name list entry has been created

For the new entry to take effect, restart or commit the change using
  't 5' : 'SET NAME-LIST'.
```

Delete

Deletes name cache entries or NetBIOS name list entries.

Syntax: delete cache-entry
name-list

cache-entry

From the configuration prompt, deletes name cache entries from the router's permanent configuration. The router prompts for a record number, which is the number of the entry you want to delete. To see a list of entry numbers, enter **list cache all**.

From the monitoring prompt, deletes name cache entries from the router's static configuration or active cache. The router prompts for a cache entry name. To see a list of entries, enter **list cache conf** or **list cache active**.

Note: The NetBIOS name is case sensitive.

Example for Configuration: **`delete cache-entry`**

```
Enter name cache record number [1]? 2

Name cache entry has been deleted
```

Example for Monitoring: **`delete cache-entry`**

```
Enter up to 15 characters of NetBIOS name (no wild cards) []? ADMIN

Name cache entry NOT found in Active list for name entered
Name cache entry has NOT been deleted from Active list

Static name cache entry deleted from Config list
```

name-list

Deletes an entry from the router's local name list.

From the configuration prompt, deletes a permanent name list entry. The router prompts for a record number that is the number of the entry you want to delete. To see a list of entry numbers, enter the **list name-list all** command. The change does not take effect until the router is restarted or the change is committed from the monitoring prompt using the **set name-list** command.

From the monitoring prompt, temporarily deletes a name list entry. The router prompts for a record number that is the number of the entry you want to delete. To see a list of entry numbers, enter the **list name-list config** command. The change does not take effect until the change is committed from the monitoring prompt using the **set name-list** command. The change is lost if the router is restarted.

Example: **`delete name-list`**

```
Enter name list record number [1]? 1

Name list entry NY_SERV*         / INDIVIDUAL has been deleted.

For the deletion to take effect, restart or commit the change using
  't 5' : 'SET NAME-LIST'.
```

Disable

Disables duplicate frame filtering, use of NetBIOS name lists, or route caching.

Syntax: `disable` duplicate-filtering
name-list local
name-list remote
route-caching

duplicate-filtering

Disables duplicate frame filtering for bridging. You cannot disable duplicate frame filtering for DLSw traffic.

Example: `disable duplicate-filtering`

```
Duplicate frame filtering is          OFF
```

name-list local

Disables the use of the local name list. The local name list entries will not be sent to any DLSw partners.

From the configuration prompt, permanently disables the use of the local name list. The change does not take effect until the router is restarted or the change is committed from the monitoring prompt using the **set name-list** command.

From the monitoring prompt, temporarily disables the use of the local name list. The change does not take effect until the change is committed from the monitoring prompt using the **set name-list** command. The change is lost if the router is restarted.

Example: `disable name-list local`

```
Use of local NetBIOS name list is  DISABLED

For the change to take effect, restart or commit the change using
  't 5' : 'SET NAME-LIST'.
```

name-list remote

Disables the use of remote name lists. All NetBIOS name lists received from DLSw partners are not used.

From the configuration prompt, permanently disables the use of remote name lists. The change does not take effect until the router is restarted or the change is committed from the monitoring prompt using the **set name-list** command.

From the monitoring prompt, temporarily disables the use of remote name lists. The change does not take effect until the change is committed from the monitoring prompt using the **set name-list** command. The change is lost if the router is restarted.

Example: `disable name-list remote`

```
Use of remote NetBIOS name list is  DISABLED

For the change to take effect, restart or commit the change using
  't 5' : 'SET NAME-LIST'.
```

route-caching

Disables route caching for bridging and DLSw. Route caching is the process of converting broadcast frames to specifically routed frames (SRFs) using the entries in the NetBIOS name cache.

Example: disable route-caching

```
Route caching is                       OFF
```

Enable

Enables duplicate frame filtering, use of NetBIOS name lists, or route caching.

Syntax: enable duplicate-filtering
name-list local
name-list remote
route-caching

duplicate-filtering

Enables duplicate frame filtering for bridging. Duplicate frame filtering is always enabled for DLSw. You cannot enable and disable it.

Example: enable duplicate-filtering

```
Duplicate frame filtering is           ON
```

name-list local

Enables the use of the local name list. The local name list entries will be sent to all DLSw partners.

From the configuration prompt, permanently enables the use of the local name list. The change does not take effect until either to router is restarted or the change is committed from the monitoring prompt using the **set name-list** command.

From the monitoring prompt, temporarily enables the use of the local name list. The change does not take effect until the change is committed from the monitoring prompt using the **set name-list** command. The change is lost if the router is restarted.

Example: enable name_list local

```
Use of local NetBIOS name list is   ENABLED

For the change to take effect, restart or commit the change using
  't 5' : 'SET NAME-LIST'.
```

name-list remote

Enables the use of remote name lists. All NetBIOS names lists received from DLSw partners are used.

From the configuration prompt, permanently enables the use of remote name lists. The change does not take effect until either to router is restarted or the change is committed from the monitoring prompt using the **set name-list** command.

From the monitoring prompt, temporarily enables the use of remote name lists. The change does not take effect until the change is committed from the monitoring prompt using the **set name-list** command. The change is lost if the router is restarted.

Example: enable name_list remote

```
Use of remote NetBIOS name list is   ENABLED

For the change to take effect, restart or commit the change using
  't 5' : 'SET NAME-LIST'.
```

route-caching

Enables route caching for bridging and DLSw. Route caching is the process of converting broadcast to specifically routed frames (SRFs) using the NetBIOS name cache.

Example: **`enable route-caching`**

```
Route caching is                        ON
```

List (Configuration)

Displays all cache entries or displays cache entries by type of entry. Displays filter configuration information or general configuration information. Displays local NetBIOS name list entries.

Syntax: list
cache all
cache entry-number
cache name
cache ip-address
filters all
filters bridge
filters dlsw
general
name-list all
name-list *entry-number*

cache all

Displays all permanent entries in the router's name cache. It does not display static or dynamic entries.

Example: **`list cache all`**

```
Entry  Name                IP Address
-----  ------------------  ---------------

    1  ACCOUNTING     <00>  20.2.1.3
    2  NOTES          <00>  20.2.3.4
```

cache entry-number *record#*

Displays a cache entry according to its entry number. Enter **list cache all** to see a list of entry numbers.

Example: **`list cache entry-number`**

```
Enter name cache record number [1]? 1

Entry  Name                IP Address
-----  ------------------  ---------------
    1  ACCOUNTING     <00>  20.2.1.3
```

cache name *name*

Displays a cache entry for a specific NetBIOS name. You can use the following wildcards to simplify your search:

***** (asterisk) stands for zero or more occurrences of any characters. For example, San* could produce:

- San Francisco
- Santa Fe
- San Juan

? (question mark) stands for any one character.

$ (dollar sign) has an effect only when the number of significant NetBIOS name characters is not 16, and when the search argument does not begin with an asterisk (*).

You can use as many wildcards as you like, up to the maximum number of characters in a NetBIOS name (15 or 16, depending on the configuration).

Note: The NetBIOS name is case sensitive.

Example: **`list cache netbios-name`**

```
Enter up to 15 characters of NetBIOS name (wild cards ok) []? Acc*

Entry  Name                IP Address
-----  ------------------- ---------------
    1  Accounting     <00> 20.2.1.3
```

cache ip-address

Lets you display all entries with a specific IP address.

Example: **`list cache ip-address`**

```
Enter IP Address [0.0.0.0]? 20.2.1.3

Entry  Name                IP Address
-----  ------------------- ---------------
    1  Accounting     <00> 20.2.1.3
```

filters all

Displays whether or not frame type filtering is on or off for both bridging and DLSw. Use the **set filters bridge**s commands to turn these filters on or off.

Example: **`list filters all`**

```
Bridge name conflict filtering is      OFF
Bridge general bcast filtering is      OFF
Bridge trace control filtering is      OFF

DLS name conflict filtering is         ON
DLS general bcast filtering is         ON
DLS trace control filtering is         ON
```

filters bridge

Displays whether or not frame type filtering is on or off for bridging. Use the **set filters bridge** to turn these filters on or off.

Example: **`list filters bridge`**

```
Bridge name conflict filtering is      OFF
Bridge general bcast filtering is      OFF
Bridge trace control filtering is      OFF
```

filters dlsw

Displays whether or not frame type filtering is on or off for DLSw. Use the **set filters dlsw** to turn these filters on or off.

Example: **`list filters dlsw`**

```
DLS name conflict filtering is         ON
DLS general bcast filtering is         ON
DLS trace control filtering is         ON
```

general

Displays the current NetBIOS caching and filtering configuration.

Example: `list general`

```
Bridge-only Information:

Bridge duplicate filtering is          OFF
Bridge duplicate frame filter t/o      1.5 seconds

DLS-only Information:

DLS command frame retry count            5
DLS max remote name cache entries      100
DLS command frame retry timeout        0.5 seconds
DLS type of local name list  NON-EXCLUSIVE
DLS use of local name list is     DISABLED
DLS use of remote name list is     ENABLED

DLS-Bridge Common Information:

Route caching is                       OFF
Significant characters in name          15
Max local name cache entries           500
Duplicate frame detect timeout         5.0 seconds
Best path aging timeout               60.0 seconds
Reduced search timeout                 1.5 seconds
Unreferenced entry timeout            5000 minutes
```

name-list all

Displays all permanently configured local NetBIOS name list entries. It does not display static entries.

Example: `list name-list all`

```
Entry  Name Qualifier    Type
-----  ----------------  ----------
  1    NY_SERV*          INDIVIDUAL
  2    NY_DOMAIN*             GROUP
```

name-list *entry-number*

Displays a particular permanently configured local NetBIOS name list entry.

Example: `list name-list entry-number`

```
Enter name list record number [1]? 1

Entry  Name Qualifier    Type
-----  ----------------  ----------
  1    NY_SERV*          INDIVIDUAL
```

List (Monitoring)

Displays various types of cache entries, filter configuration, general configuration information, NetBIOS name lists, or statistics on other things.

```
Syntax:  list    cache active
                 cache config
                 cache group
                 cache local
                 cache name
                 cache remote
                 cache unknown
                 filters all
                 filters bridge
                 filters dlsw
                 general
                 name-list all
                 name-list config
                 name-list local
```

name-list remote
statistics cache
statistics frames bridge
statistics frames dlsw
statistics general bridge
statistics general dlsw

cache active

Displays all active entries in the router's name cache.

The number in angle brackets is the 16th character of the NetBIOS name. This character, which you can enter in hexadecimal if you create the cache entry, is used by some NetBIOS applications for special purposes.

If the Name Type field does not specify LOCAL, it is a remote entry.

Example: **`list cache active`**

```
Cnt  NetBIOS Name         Name Type          Entry Type
---  -------------------  -----------------  ----------
  1  HYPERION       <01>  INDIVIDUAL LOCAL   DYNAMIC
  2  LANGROUP       <00>  UNKNOWN            STATIC
  3  ACCOUNTING     <00>  GROUP              PERMANENT
```

cache config

Displays all static and permanent name cache entries. Does not show dynamic entries.

The number in angle brackets is the 16th character of the NetBIOS name. This character, which you can enter in hexadecimal if you create the cache entry, is used by some NetBIOS applications for special purposes.

Example: **`list cache config`**

```
Name                IP Address      Source     Last Mod
------------------  --------------  ---------  ---------
Admin         <00>  20.3.120.8      STATIC     ADDED
Finance       <01>  20.4.96.8       PERMANENT  MODIFIED
Notes         <00>  20.8.210.3      PERMANENT  UNCHANGED
```

cache group

Displays cache entries that exist for NetBIOS group names.

Example: **`list cache group`**

```
Cnt  NetBIOS Name         Entry Type  Loc Path State  Rem Path State
---  -------------------  ----------  --------------  --------------
  2  HYPERION       <01>  DYNAMIC     UNKNOWN         GROUP
  3  EXCEL          <00>  DYNAMIC     GROUP           GROUP
```

cache local

Displays local cache entries. Local cache entries are those that the router learns via the local bridge network.

For NetBIOS clients the Local Path State is always Unknown and the MAC address and Routing information fields are always empty.

Example: **`list cache local`**

```
Cnt  NetBIOS Name     Loc Path State  MAC Address  Routing Information
---  ---------------  --------------  ----------- --------------------
  2  HYPERION   <01>  UNKNOWN
```

Cnt	Number of the cache entry.
NetBIOS Name	The entry's NetBIOS name.
Loc Path State	Local Path State.

MAC Address If the entry is a server, displays the MAC address of the server.

Routing Information Displays standard RIF information.

cache name *name*

Displays a cache entry for a specific NetBIOS name. You can use the following wildcards to simplify your search:

***** (asterisk) stands for zero or more occurrences of any characters. For example, San* could produce:

- San Francisco
- Santa Fe
- San Juan

? (question mark) stands for any one character.

$ (dollar mark) has an effect only when the number of significant NetBIOS name characters is not 16, and when the search argument does not begin with an asterisk (*).

You can use as many wildcards as you like, up to the maximum number of characters in a NetBIOS name (15 or 16 depending on the configuration).

Note: NetBIOS names are case sensitive.

Example: **`list cache name`**

```
     NetBIOS Name        Name Type          Entry Type
     ------------------- -----------------  ----------
     HYPERION       <01>  INDIVIDUAL REMOTE  DYNAMIC

Count of name cache entry hits ............................... 20

Age of name cache entry ...................................... 689
Age of name cache last reference ............................. 85

Local path information:

  Loc Path State  Timestamp  MAC Address    LFS   Routing Information
  --------------  ---------  ------------  -----  -------------------
  UNKNOWN         689

Remote path information:

  Rem Path State  Timestamp   LFS   IP Address(es)
  --------------  ---------  -----  --------------
  BEST FOUND      85         2052   20.3.120.8
```

cache remote

Displays cache entries that the router learns over the DLSw WAN.

Example: **`list cache remote`**

```
Cnt  NetBIOS Name        Entry Type  Rem Path State  IP Address(es)
---  ------------------- ----------  --------------  --------------
  2  HYPERION       <01>  STATIC      BEST FOUND      20.3.120.8
  3  EXCEL          <00>  DYNAMIC     SEARCH ALL
```

Cnt Number of the cache entry.

NetBIOS Name The entry's NetBIOS name.

Rem Path State Remote Path State. Possible states are:

Best Found
The router found the best route to this station.

Unknown
The router has not yet found the best route to this station.

Group
The router does not search for a best path for group names.

Search Limited
The router is conducting a limited search for this NetBIOS name. See the **set cache-parms** command for more information on reduced search.

Search All
The router is conducting a full search. When the **set cache-parms** command's reduced search timer expires, the router conducts a full search.

IP Address(es) If best path found, displays the IP address or addresses associated with the neighbor DLSw that can reach the NetBIOS station.

cache unknown
Displays cache entries where the type NetBIOS name is unknown. The router enters all dynamic entries as Unknown until it learns the type of name. It then marks entries as local, remote, or group.

Example: `list cache unknown`

```
Cnt  NetBIOS Name     Entry Type      Loc Path State  Rem Path State
---  ---------------  --------------  --------------  --------------
  2  HYPERION   <01>  STATIC          UNKNOWN          UNKNOWN
  3  EXCEL      <00>  STATIC          UNKNOWN          UNKNOWN
```

filters all
Displays whether or not frame type filtering is on or off for both bridging and DLSw. Use the **set filters bridge** and **set filters dlsw** commands to turn these filters on or off.

Example: `list filters all`

```
Bridge name conflict filtering is      OFF
Bridge general bcast filtering is      OFF
Bridge trace control filtering is      OFF

DLS name conflict filtering is         ON
DLS general bcast filtering is         ON
DLS trace control filtering is         ON
```

filters bridge
Displays whether or not frame type filtering is on or off for bridging. Use the **set filters bridge** command to turn these filters on or off.

Example: `list filters bridge`

```
Bridge name conflict filtering is      OFF
Bridge general bcast filtering is      OFF
Bridge trace control filtering is      OFF
```

filters dlsw
Displays whether or not frame type filtering is on or off for both DLSw. Use the **set filters dlsw** command to turn these filters on or off.

Example: `list filters dlsw`

```
DLS name conflict filtering is         ON
DLS general bcast filtering is         ON
DLS trace control filtering is         ON
```

general

Displays the current NetBIOS caching and filtering configuration.

Example: **`list general`**

```
Bridge-only Information:

Bridge duplicate filtering is          OFF
Bridge duplicate frame filter t/o      1.5 seconds

DLS-only Information:

DLS command frame retry count            5
DLS max remote name cache entries      100
DLS command frame retry timeout        0.5 seconds
DLS type of local name list  NON-EXCLUSIVE
DLS use of local name list is     DISABLED
DLS use of remote name list is     ENABLED

DLS-Bridge Common Information:

Route caching is                       OFF
Significant characters in name          15
Max local name cache entries           500
Duplicate frame detect timeout         5.0 seconds
Best path aging timeout               60.0 seconds
Reduced search timeout                 1.5 seconds
Unreferenced entry timeout            5000 minutes
```

name-list all

Displays all currently active NetBIOS name list entries both local and remote. If local name list entries have not been committed or the use of local name lists is disabled, local name list entries will not appear in the list. If use of the remote name lists is disabled, remote name list entries will not appear in the list.

Example: **`list name-list all`**

```
Name Qualifier    Type        IP Address
----------------  ----------  ---------------
LA_DOMAIN*             GROUP  20.2.1.3
LA_SERV*          INDIVIDUAL  20.2.1.3
NY_DOMAIN*             GROUP  Local
NY_SERV*          INDIVIDUAL  Local
SF_DOMAIN*             GROUP  20.2.3.4
SF_SERV*          INDIVIDUAL  20.2.3.4
TEMP_DOMAIN            GROUP  Local
TEMP_SERV01       INDIVIDUAL  Local
```

name-list config

Displays all permanently and temporarily configured local NetBIOS name list entries.

The source field can have one of the following values:

PERMANENT Permanently configured entries.

STATIC Temporarily configured entries.

The LastMod field can have one of the following values:

ADDED The local name list entry has been added, but the change has not been committed.

DELETED The local name list entry has been deleted, but the change has not been committed

UNCHANGED The local name list entry has been added and the change has been committed

Example: **`list name-list config`**

```
Entry  Name Qualifier    Type        Source     LastMod
-----  ----------------  ----------  ---------  ---------
  1    NY_SERV*          INDIVIDUAL  PERMANENT  UNCHANGED
  2    NY_DOMAIN*             GROUP  PERMANENT  UNCHANGED
  3    TEMP_SERV01       INDIVIDUAL  STATIC     ADDED
  4    TEMP_DOMAIN            GROUP  STATIC     ADDED
```

name-list local

Displays all currently active local NetBIOS name list entries If local name list entries have not been committed or the use of local name lists is disabled, local name list entries will not appear in the list.

Example: **`list name-list local`**

```
LOCAL Name List
Type of Name List (active) .......     EXCLUSIVE
Type of Name List (pending) ...... NON-EXCLUSIVE

Name Qualifier    Type
----------------  ----------
NY_DOMAIN*             GROUP
NY_SERV*          INDIVIDUAL
TEMP_DOMAIN            GROUP
TEMP_SERV01       INDIVIDUAL
```

name-list remote

Displays all currently active remote NetBIOS name list entries for a particular DLSw partner. If use of the remote name lists is disabled, no entries will appear.

Example: **`list name-list remote`**

```
Enter IP Address [0.0.0.0]? 20.2.1.3

Partner IP Address ........... 20.2.1.3

Type of Name List ...........      EXCLUSIVE
    Use of remote name lists ..... ENABLED

Name Qualifier    Type
----------------  ----------
LA_DOMAIN*             GROUP
LA_SERV*          INDIVIDUAL
```

statistics cache

Lists the following name cache statistics.

Example: **`list statistics cache`**

```
Local name cache entries                 1
Remote name cache entries                1
Local individual names                   1
Remote individual names                  0
Group names                              0
Unknown names                            1
Name cache hits                       2194
Name cache misses                        2
```

statistics frames bridge

Lists the following name cache statistics for bridging.

Example: **`list statistics frames bridge`**

```
Frames in cache                        0
Name query frames                      0
Status query frames                    0
Add name frames                        0
Add group name frames                  0
Name in conflict frames                0
Frames not filtered as duplicates      0
```

statistics frames dlsw

Lists the following name cache statistics for DLSw.

Example: **`list statistics frames dlsw`**

```
Name query frames                      0
Status query frames                    0
Add name frames                        0
Add group name frames                  0
Name in conflict frames                0
Frames not filtered as duplicates      0
```

statistics general bridge

Displays frame counts for bridging.

Example: **`list statistics general bridge`**

```
Frames received                     1339
Frames discarded                       0
Frames forwarded to bridge          1339
Frames forwarded to DLS             1339
```

statistics general dlsw

Displays frame counts for DLSw.

Example: **`list statistics general dlsw`**

```
Frames received                     1339
Frames discarded                       0
Frames forwarded to bridge          1339
```

Set

Sets name caching parameters, turns frame type filtering on or off for either bridging or DLSw, adjusts duplicate frame filtering timers and frame retry timers, and sets NetBIOS name list parameters. Also displays the NetBIOS name and byte filtering prompt.

Syntax: set cache-parms
filters bridge
filters byte
filters dlsw
filters name
general
name-list

cache-parms

Sets name caching parameters that apply to bridging or switching.

Example: **`set cache-parms`**

```
Significant characters in name [15]?
Best path aging timeout value in seconds [60.0]?
Reduced search timeout value in seconds [1.5]?
Unreferenced entry timeout value in minutes [5000]?
Max nbr local name cache entries [500]?
Max nbr remote name cache entries [100]?

Cache parameters set
```

Significant characters in name

Determines whether the router considers 15 or 16 characters when it looks up the NetBIOS name. If you enter 15, the router ignores the 16th character. If you select 16, the router includes the 16th character when it looks up cache entries.

The default is 15.

Best path aging timeout

Amount of time the router considers the address and route for a name cache entry to be the best path to that station. When this timer expires, the router deletes the name cache entry and attempts to discover a new best path for the NetBIOS name.

To determine the best path, the router considers transmission time between nodes on all possible routes connecting those nodes, as well as largest frame size. The router does not consider a path suitable if it cannot accommodate the largest NetBIOS frame that could be transmitted over the path.

The default is 60 seconds. The range is 1.0 to 100000.0 seconds.

Reduced search timeout

When the router receives a Name-Query, Status.Query, or Datagram during the timeout period, it carries out a search based on current NetBIOS name cache information.

If the router receives a duplicate frame after this timer expires, it assumes the previous route is not longer valid and it widens its search. The router forwards the duplicate frame to both bridges and DLS. DLS broadcasts the corresponding SSP message to all possible DLS partners.

The default is 1.5 seconds. The range is 1.0 to 100.0 seconds.

Unreferenced entry timeout

The router keeps a name that is not referenced in its cache for this length of time before deleting it. If the cache fills up, the router removes entries sooner.

The default is 5000 minutes. The range is 1 to 100000 minutes.

Max nbr local name cache entries

Maximum number of locally-learned entries the router saves in the name cache.

The default is 500. The range is 100 to 30000. You can lower this value to save router memory. To optimize memory usage, processor usage, and the amount of broadcast traffic, the number of local name cache entries should be set as close as possible to the total number of NetBIOS stations (servers and clients) that are active on this router’s local bridge network.

Max nbr remote name cache entries

Maximum number of remotely-learned entries, group name entries, and unknown entries that the router saves in the name cache.

The default is 100. The range is 100 to 30000. You can lower this value to save router memory. To optimize memory usage, processor usage, and the amount of broadcast traffic, the number of remote name cache entries should be set to the number of remote NetBIOS servers that are to be accessed by NetBIOS clients on this router’s local bridge network, plus about 25%.

filters bridge

Turns frame-type filtering for bridging on or off.

Example: set filters bridge

```
Filter Name Conflict frames? [No]: y

Name conflict filtering is            ON

Filter General Broadcast frames? [No]:

General broadcast filtering is        OFF

Filter Trace Control frames? [No]:

Trace control filtering is            OFF
```

filters byte

From the `NetBIOS config>` prompt, displays the NetBIOS filtering configuration prompt (`NETBIOS Filter config>`). Configuring NetBIOS filtering is explained in Chapter 10, “Configuring NetBIOS Filtering.”

From the `NetBIOS console>` prompt, displays the NetBIOS filtering monitoring prompt (`NETBIOS Filter>`). Monitoring NetBIOS filtering is explained in Chapter 11, “Monitoring NetBIOS Filtering.”

This parameter allows you to access NetBIOS byte filtering.

Example: set filters byte

```
NETBIOS Filtering configuration
NETBIOS Filter config>
```

filters dlsw

Sets frame-type filters for DLSw traffic.

Example: set filters dlsw

```
Filter Name Conflict frames? [Yes]:

Name conflict filtering is             ON

Filter General Broadcast frames? [Yes]:

General broadcast filtering is         ON

Filter Trace Control frames? [Yes]:

Trace control filtering is             ON
```

filters name

From the `NetBIOS config>` prompt, displays the NetBIOS filtering configuration prompt (`NETBIOS Filter config>`). Configuring NetBIOS filtering is explained in Chapter 10, “Configuring NetBIOS Filtering.”

From the `NetBIOS console>` prompt, displays the NetBIOS filtering monitoring prompt (`NETBIOS Filter>`). Monitoring NetBIOS filtering is explained in Chapter 11, “Monitoring NetBIOS Filtering.”

This parameter allows you to access NetBIOS name filtering.

Example: set filters name

```
NETBIOS Filtering configuration
NETBIOS Filter config>
```

general

Sets the duplicate frame timeout, duplicate frame-detect timeout, and the command frame retry count and timeout. See “Duplicate Frame Filtering” on page 9-5 for more information on how duplicate frame filters work.

Example: **`set general`**

```
ATTENTION! Setting Duplicate Frame Filter Timeout to zero...
disables duplicate frame checking!

Duplicate frame filter timeout value in seconds [1.5]?

Duplicate frame detect timeout value in seconds [5.0]?

General parameters set
```

If DLSw is enabled, the software also prompts you for:

```
Command frame retry count [5]?
Command frame retry timeout value in seconds [0.5]?
```

Duplicate frame filter timeout

Applies only to bridged traffic if duplicate-filtering is enabled. During this timeout period, the router filters all duplicate frames it receives.

The range is 0.0 to 100.0 seconds. Zero disables duplicate frame checking. The default is 1.5 seconds.

Duplicate frame-detect timeout

Applies to both bridged and DLSw traffic. Amount of time the router saves entries in its duplicate frame filter database. When this timer expires, the router creates new entries for new frames that it receives.

The range is 0.0 to 100.0 seconds. The default is 5 seconds.

Command frame retry count

Applies only to DLSw traffic.

Number of duplicate NetBIOS UI frames the target DLSw router sends to its locally-attached LAN. These frames are sent at intervals specified by the command frame retry timeout.

The range is 0 to 10. The default is 5.

Command frame retry timeout

Applies only to DLSw traffic. This is the interval at which a neighbor DLSw router retries sending duplicate NetBIOS UI frames to its local bridge network.

The range is 0.0 to 10.0 seconds. The default is 0.5 seconds.

name-list

Sets parameters related to the local NetBIOS name list. Currently the only local NetBIOS name list related parameter is the local NetBIOS name list exclusivity.

From the configuration prompt, permanently sets the local NetBIOS name list parameters. The change does not take effect until either the router is restarted or the change is committed from the monitoring prompt using **set name-list** command.

From the monitoring prompt, this command temporarily sets the local NetBIOS name list parameters. The command also commits any NetBIOS name list changes that have been made from the configuration or monitoring prompts.

Example (configuration prompt): `set name-list`

```
Local NetBIOS name list exclusivity (E=exclusive, N=non-exclusive) [N]? e

NetBIOS name list parameter set.

For the change to take effect, restart or commit the change using
  't 5' : 'SET NAME-LIST'.
```

Example (monitoring prompt): `set name-list`

```
Local NetBIOS name list exclusivity (E=exclusive, N=non-exclusive) [N]? e

NetBIOS name list parameter set.

For the change to take effect, commit the change (next question).

The next question allows you to commit
any of the following changes (permanent and temporary):
  - changes made using ENABLE NAME-LIST LOCAL
  - changes made using ENABLE NAME-LIST REMOTE
  - changes made using DISABLE NAME-LIST LOCAL
  - changes made using DISABLE NAME-LIST REMOTE
  - changes made using ADD NAME-LIST
  - changes made using DELETE NAME-LIST
  - changes made using SET NAME-LIST

Would you like to commit the name list changes? [No]: y

Use of local name list remains    ENABLED.
Use of remote name list remains    ENABLED.
Type of local name list has changed from NON-EXCLUSIVE to     EXCLUSIVE.
Entry deleted temporarily: TEMP_SERV01      / INDIVIDUAL.
Would you still like to commit the name list changes? [No]: y


NetBIOS name list changes have been committed.
```

local NetBIOS name list exclusivity specifies whether the local NetBIOS name list represents all NetBIOS names reachable through this router (EXCLUSIVE) or represents a set (but not necessarily all) of NetBIOS names reachable through

Test (Monitoring only)

Allows testing of real NetBIOS names against the current NetBIOS cache or the NetBIOS name list.

Syntax: test   cache
                    name-list

test cache

Displays a list of current DLSw partners to which a DLSw frame with a given NetBIOS destination name would be forwarded and how the frame will be forwarded.

Example (no corresponding NetBIOS cache entry):

`test cache ABC`

```
Destination NetBIOS name being tested .... ABC            <20>

Name cache entry NOT found.

How frame destined for this NetBIOS name is forwarded to DLSw partners .....
  Send to all partners.
```

Example (corresponding NetBIOS cache entry):

`test cache LA_SERV01`

```
Destination NetBIOS name being tested .... LA_SERV01      <00>

Name cache entry found:
  Name type = INDIVIDUAL REMOTE;   Entry type = DYNAMIC

How frame destined for this NetBIOS name is forwarded to DLSw partners .....
  Send to all name list learned and dynamically learned partners.

List of DLSw partners to which frame destined for this name is forwarded .....

  Send via TCP           to 20.2.1.3 ( Name list, Learned )
```

test name-list

Displays a list of NetBIOS name list entries (local or remote) that match the given NetBIOS name.

Example: **`test name-list`**

```
Enter up to 15 characters of NetBIOS name (no wild cards).
  Enter NetBIOS name []? LA_SERV01
Enter last character of NetBIOS name in hex [0]?

Name Qualifier    Type        IP Address
----------------  ----------  ---------------
LA_SERV*          INDIVIDUAL  20.2.1.3
```

Exit

Returns to the previous prompt.

Syntax: exit

Example: **`exit`**

Chapter 10. Configuring NetBIOS Filtering

This chapter summarizes and then explains all of the NetBIOS filtering configuration commands. These commands let you configure NetBIOS filtering as an added feature to ASRT bridging. Configuration commands are accessed from the `NetBIOS config>` prompt.

Included are the following sections:

- “Accessing the ASRT and the DLSW Configuration Environments”
- “NetBIOS Filtering Configuration Commands”

Accessing the ASRT and the DLSW Configuration Environments

To display the NetBIOS filtering prompt from the ASRT environment, enter the commands as shown in the following example:

```
Config> protocol asrt
Adaptive Source Routing Transparent Bridge user configuration

ASRT config> netbios
NetBIOS Support User Configuration

NetBIOS config> set filters name or byte
NetBIOS filtering configuration

NetBIOS filter config>
```

To display the `NetBIOS config>` prompt from the DLSw configuration environment:

```
Config> protocol dls
DLSw protocol user configuration

DLSw config> netbios
NetBIOS Support User Configuration

NetBIOS config> set filters name or byte
NetBIOS filtering configuration

NetBIOS filter config>
```

Table 10-1 shows the NetBIOS filtering configuration commands.

NetBIOS Filtering Configuration Commands

Table 10-1 (Page 1 of 2). NetBIOS Filtering Configuration Commands

Command	Function
? (Help)	Lists all of the NetBIOS filtering configuration commands, or lists the options associated with specific commands.
Create	Creates byte filter and host-name filter lists for NetBIOS filtering.
Delete	Deletes byte filter and host-name filter lists for NetBIOS filtering.
Disable	Disables NetBIOS filtering on the bridging router.
Enable	Enables NetBIOS filtering on the bridging router.
Filter-on	Assigns a created filter to a specific port. This filter can then be applied to all NetBIOS packets input OR output on the specified port.
List	Displays all information concerning created filters.

Table 10-1 (Page 2 of 2). NetBIOS Filtering Configuration Commands

Command	Function
Update	Adds information to or deletes information from a host-name or byte filter list.
Exit	Exits the NetBIOS filtering configuration process and returns you to the previous prompt.

? (Help)

Use the **? (Help)** command to obtain a list of the commands available from that prompt level. You can also enter this command after specific command names to obtain a listing of the command options available for that command.

Syntax: ?

Example:
```
create ?
  byte-filter-list
  name-filter-list
```

Create

Use the **create** command to create a byte filter or host-name filter list.

Syntax: create   byte-filter-list *filter-list*
                name-filter-list *filter-list*

byte-filter-list *filter-list*

Creates a byte filter list name for NetBIOS filtering. You can use up to 16 characters to identify the list being built. *Filter-list* must be a unique name that has not been used previously with the **create byte-filter-list** or **create name-filter-list** command.

Example: `create byte-filter-list newyork`

name-filter-list *filter-list*

Creates a host-name filter list name for NetBIOS filtering. You can use up to 16 characters to identify the name filter list being built. *Filter-list* must be a unique name that has not been used previously with the **create byte-filter-list** or **create name-filter-list** command.

Example: `create name-filter-list atlanta`

Delete

Use the **delete** command to delete byte filter lists, host-name filter lists, and filters created using the **filter-on input** or **filter-on output** command. The command removes all information associated with byte and host-name filter lists. It also frees the user-defined string as a name for a new filter list.

Syntax: delete   byte-filter-list *filter-list*
                name-filter-list *filter-list*
                filter input *port#*
                filter output *port#*

byte-filter-list *filter-list*

Deletes a byte filter list created for NetBIOS filtering. *Filter-list* is the user-defined string being used to identify the byte filter list being deleted.

Example: `delete byte-filter-list newyork`

name-filter-list *filter-list*

Deletes a host-name filter list created for NetBIOS filtering. *Filter-list* is the user-defined string that is used to identify the name-filter-list being deleted.

Example: `delete name-filter-list atlanta`

filter input *port#*

Deletes a filter that was created using the **filter-on input** command. The command removes all information associated with the filter and fills any resulting gap in filter numbers.

Example: `delete filter input 2`

filter output *port#*

Deletes a filter that was created using the **filter-on output** command. The command removes all information associated with the filter and fills any resulting gap in filter numbers.

Example: `delete filter output 2`

Disable

Use the **disable** command to globally disable NetBIOS name and byte filtering on the router.

Syntax: disable netbios-filtering

Example: `disable netbios-filtering`

Enable

Use the **enable** command to globally enable NetBIOS name and byte filtering on the router.

Syntax: enable netbios-filtering

Example: `enable netbios-filtering`

Filter-on

This command assigns one or more previously configured filter lists to the input or output of a specific port.

Syntax: filter-on input *port# filter-list <operator filter-list ...>*
output *port# filter-list <operator filter-list ...>*

input *port# filter-list <operator filter-list . . . >*

This command assigns one or more filter lists to incoming packets on a specific port. The resulting filter is then applied to all NetBIOS packets input on the specified port.

Port# is a configured bridge port number on the router. The port number identifies this filter. Enter **list** to see a list of port numbers Filter-list is a string previously entered via the **create** command. To add additional filter lists to this port, enter AND or OR in all capital letters followed by the filter list name.

Note: Multiple operators can be used to create a complex filter. If you enter multiple operators, they must all be entered at the same time on the same command line.

The filter created by this command is applied to all incoming NetBIOS packets on the specified port. Each filter list on the command line is evaluated left to right along with any operators that are present. An Inclusive evaluation of a filter list is equivalent to a TRUE condition and an Exclusive evaluation is equivalent to a FALSE condition. If the result of the evaluation of the filter-list(s) is TRUE, the packet is bridged. Otherwise, the packet is filtered (dropped).

If the packet is not one of the types supported by NetBIOS filtering then all host-name filter lists for this filter are designated “Inclusive” (TRUE). If an input filter already exists for specified port number, an error message is displayed.

Example: `filter-on input 2 newyork AND boston`

output *port# filter-list <operator filter-list . . . >*

This command assigns one or more filters to outgoing packets on a port. This filter is then applied to all NetBIOS packets output on that port.

Port# is a configured bridge port number on the router. The port number identifies this filter. Enter **list** to see a list of port numbers. Filter-list is a string previously entered via the create command. An optional operator is entered as either “AND” or “OR.” The operator is entered in all capital letters. If an operator is present, it must be followed by a filter-list name. The port number is used to identify this filter.

Note: Multiple operators can be used. This creates a complex filter. If one or more operators are present, they must all be entered at the same time on the same command line.

The filter created by this command is applied to all NetBIOS packets output on the specified port number. Each filter list on the command line is evaluated left to right along with any operators that are present. An Inclusive evaluation of a filter list is equivalent to a TRUE condition and an Exclusive evaluation is equivalent to a FALSE condition. If the result of the evaluation of the filter-list(s) is TRUE, the packet is bridged. Otherwise, the packet is filtered (dropped).

If the packet is not one of the types supported by NetBIOS filtering then all host-name filter lists for this filter are designated “Inclusive” (TRUE). If an output filter already exists for specified port number, an error message is displayed.

Example: `filter-on output 2 newyork OR boston`

List

Use the **list** NetBIOS Filtering command to display all information concerning created filters.

Syntax: list

Example: `list`

```
NetBIOS Filtering: Disabled

NetBIOS Filter Lists
--------------------

Handle          Type

nlist           Name
newyork         Byte

NetBIOS Filters
---------------

Port #     Direction      Filter List Handle(s)

3          Output         nlist
```

NetBIOS Filtering: Displays whether NetBIOS filtering is enabled or disabled.

NetBIOS Filter Lists Displays the user-defined name (handle) of the configured filter lists. For type, “Name” indicates a host-name filter list and “Byte” indicates a byte filter list.

NetBIOS Filters Displays the assigned port number and direction (input or output) of each filter. Filter List Handles displays the names of the filter lists making up the filter.

Update

Use the **update** command to add or delete information from host-name or byte filter lists. The filter-list is a string previously entered via the create byte (or name) filter-list prompt. This command brings you to the `NetBIOS Byte (or Name) filter-list Config>` prompt, which lets you perform update tasks to the specified filter list. At this prompt you can add, delete, list, or move filter-items from byte and host-name filter lists. At this prompt you can also set the default value of each filter list to Inclusive or Exclusive.

Using the add subcommand creates a filter item within the filter list. The first filter item created is assigned number 1, the next one is assigned number 2, and so on. After you enter a successful add subcommand, the router displays the number of the filter item just added.

Note: Adding more filter items to filter lists adds to processing time (due to the time it takes to evaluate each filter item in the list) and can affect performance in heavy NetBIOS traffic.

The order in which filter items are specified for a given filter list is important as this determines the way in which the filter items are applied to a packet. The first match that occurs stops the application of filter items, and the filter list is evaluated as either Inclusive or Exclusive (depending on the Inclusive or Exclusive designation of the matched filter item). If none of the filter items of a filter list produces a match, then the default condition (Inclusive or Exclusive) of the filter list is returned.

The delete subcommand specifies the number of a filter item to be deleted from the filter list. When a delete subcommand is given, any hole created in the list is filled in. For example, if filter items 1, 2, 3, and 4 exist and filter item 3 is deleted, then filter item 4 will be renumbered to 3.

The default subcommand lets you change the default setting of the filter list to either Inclusive or Exclusive. If a filter list evaluates as Inclusive, then the packet is bridged. Otherwise, the packet is filtered.

The move subcommand is available to renumber filter items within a filter list. The first argument to the move subcommand is the number of the filter list to be moved. The second argument to the move subcommand is the number of the filter list after which the first filter list should be moved.

Syntax: update byte-filter-list . . .
name-filter-list . . .

byte-filter-list *filter-list*

Updates information belonging to a byte filter-list. The filter-list parameter is a string previously entered via the **create byte-filter-list** command. This command brings you to the next `NetBIOS BYTE filter-list Config>` command level (see example). At this level you can perform update tasks to the specified filter-list.

Example: **`update byte-filter-list newyork`**

```
NetBIOS Byte newyork Config>
```

At this prompt level you can execute several commands. Each available command is listed in the "**Update Byte-Filter** Command Options" section which follows. The correct syntax is listed followed by a description of that command and its required parameters.

name-filter-list *filter-list*

Updates information belonging to a name-filter list. This command is identical to the byte-filter-list command, except that it specifies a name-filter list rather than a byte-filter list. The filter-list parameter is a string previously entered via the create name-filter-list prompt. This command brings you to the next `NetBIOS Name filter-list Config>` command level (see example). At this level you can perform update tasks to the specified filter-list.

Example: **`update name-filter-list accounting`**

```
NetBIOS Name accounting Config>
```

At this prompt level you can execute several commands. Each available command is listed in the "**Update Name-Filter** Command Options" section which follows. The correct syntax is listed followed by a description of that command and its required parameters.

Update BYTE-Filter-List (Command Options)

This section lists the command options available for the **update byte-filter-list** command:

add inclusive *byte-offset hex-pattern <hex mask>*

Adds a filter item to the byte filter list. If the byte filter item that is added produces a match with a NetBIOS packet, the filter list it belongs to will evaluate to Inclusive (True).

- Byte-offset specifies the number of bytes (in decimal) to offset into the packet being filtered. This starts at the NetBIOS header of the packet.
- Hex-pattern is a hexadecimal number used to compare with the bytes starting at the byte-offset offset of the NetBIOS header. Syntax rules for hex-pattern include no 0x in front, a maximum of 32 numbers, and an even number of hex numbers.
- Hex-mask, if present, must be the same length as hex-pattern and is logically ANDed with the bytes in the packet starting at byte-offset before the result is compared for equality with hex-pattern. If the hex-mask argument is omitted, it is considered to be all binary 1s.

If the offset and pattern of a byte filter item represent bytes that do not exist in a NetBIOS packet (that is, if the packet is shorter than was intended when setting up a byte-filter list), then the filter item will not be applied to the packet and the packet will not be filtered. If a series of byte filter items is used to set up a single NetBIOS filter list, then a packet will not be tested for filtering if any of the byte filter items within the NetBIOS filter list represent bytes that do not exist in the NetBIOS packet.

Example: **`add inclusive`**

```
Byte Offset  [0] ?
Hex Pattern  [] ?
Hex Mask (<CR> for no mask)  [] ?
```

add exclusive *byte-offset hex-pattern <hex mask>*

Adds a filter item to the byte filter list. This command is identical to the add inclusive command, except that if the result of the comparison between the filter item and a NetBIOS packet results in a match, then the filter list evaluates to Exclusive (False). Datagram Broadcast Packets can be specified to be discarded by using this command with a byte offset of 4 and a byte pattern of 09.

- Byte-offset specifies the number of bytes (in decimal) to offset into the packet being filtered. This starts at the NetBIOS header of the packet.
- Hex-pattern is a hexadecimal number that is compared with the bytes starting at the byte-offset offset of the NetBIOS header. Syntax rules for hex-pattern include no 0x in front, a maximum of 32 numbers, and an even number of hex numbers.
- Hex-mask, if present, must be the same length as hex-pattern and is logically ANDed with the bytes in the packet starting at byte-offset before the result is compared for equality with hex-pattern. If the hex-mask argument is omitted, it is considered to be all binary 1's.

If the offset and pattern of a byte filter item represent bytes that do not exist in a NetBIOS packet (that is, if the packet is shorter than was intended when setting up a byte-filter list), then the filter item will not be applied to the packet and the packet will not be filtered. If a series of byte filter items is used to set up a single NetBIOS filter list, then a packet will not be tested for filtering if any of the byte filter items within the NetBIOS filter list represent bytes that do not exist in the NetBIOS packet.

Example: `add exclusive`

```
Byte Offset  [0] ?
Hex Pattern  [] ?
Hex Mask  (<CR> for no mask)  [] ?
```

default include

Changes the default setting of the filter list to “inclusive.” This command indicates that if no filter items of the filter list match the contents of the packet being considered for filtering, the filter list will be evaluated as Inclusive. This is the default setting.

default exclude

Changes the default setting of the filter list to “exclusive.” This command indicates that, if no filter items of the filter list match the contents of the packet being considered for filtering, the filter list will be evaluated as Exclusive.

delete *filter-item*

Deletes a filter item from the filter list.

- Filter-item is a decimal number representing a filter item that was previously created by the add command.

list

Displays information related to filter items in the specified filter list.

```
BYTE Filter List Name:     Engineering
BYTE Filter List Default:  Exclusive
Filter Item # Inc/Ex       Byte Offset    Pattern          Mask
1          Inclusive        14        0x123456         0xFFFF00
2          Exclusive         0        0x9876           0xFFFF
3          Exclusive        28        0x1000000        0xFF00FF00
```

move *filter-item1 filter-item2*

Reorders filter items within the filter list. The filter item whose number is specified by filter-item1 is moved and renumbered to be just after filter item2.

exit

Exits to the previous command prompt level.

Update NAME-Filter-List (Command Options)

The following section lists the command options available for the update name-filter-list command:

add inclusive *ASCII host-name <LAST-hex number>*

Adds a filter item to the host-name filter list. With this command, the host name fields of the NetBIOS packets are compared with the host-name given in this command. The following list shows how these comparisons are made:

- ADD_GROUP_NAME_QUERY: Source NetBIOS name field is examined
- ADD_NAME_QUERY: Source NetBIOS name field is examined
- DATAGRAM: Destination NetBIOS name field is examined
- NAME_QUERY: Destination NetBIOS name field is examined

If there is a match (taking into account wildcard designations in this command), then the filter list evaluates to Inclusive. If not, the next filter item of the filter list (if any) of the filter is applied to the packet. If the packet is not one of the four types supported by NetBIOS Name filtering, then the packet is bridged.

- Host-name is an ASCII string up to 16 characters long. A question mark (?) can be used in host-name to indicate a single character wildcard. An asterisk (*) can be used as the final character of host-name to indicate a wildcard for the remainder of the host-name. If host-name contains fewer than 15 characters, it is padded to the 15th character with ASCII spaces. Host-name can contain any character but the following:

  . / \ [ ] : | < > + = ; , <space>

- LAST-hex-number can be used if host-name contains fewer than 16 characters. It is a hexadecimal number (with no 0x in front of it) which indicates the value to be used for the last character. If the LAST argument is not specified on a hostname less than 16 characters, then a "?" wildcard is supplied for the 16th character.

add inclusive HEX *hexstring*

Adds a filter item to the host-name filter list. This command is functionally the same as add inclusive ASCII command. However, the representation of hostname is different. This command supplies the hostname as a series of hexadecimal numbers (with no 0x in front).

- Hexstring must consist of an even number of hexadecimal numbers. If you do not supply a full 32 hexadecimal numbers, ASCII blanks are padded to the 29th and 30th numbers and a wildcard is supplied as the 31st and 32nd (16th byte) numbers. A wildcard for a single byte can be specified by "??."

add exclusive ASCII *host-name <LAST-hex-number>*

Adds a filter item to the host-name filter list. This command is identical to the add inclusive ASCII command, except that packets that are matched against this filter item produce an Exclusive result for the filter list.

- Host-name is an ASCII string up to 16 characters long. A question mark (?) can be used in host-name to indicate a single character wildcard. An asterisk (*) can be used as the final character of host-name to indicate a wildcard for the remainder of the host-name. If host-name contains fewer than 15 characters, it is padded to the 15th character with ASCII spaces. Host-name can contain any character but the following:

  . / \ [ ] : | < > + = ; , <space>

- LAST-hex-number can be used if host-name contains fewer than 16 characters. It is a hexadecimal number (with no 0x in front of it) that indicates the value to be used for the last character. If the LAST argument is not specified on a host-name less than 16 characters, then a "?" wildcard is supplied for the 16th character.

add exclusive HEX *hexstring*

Adds a filter item to the name filter list. This command is functionally the same as the add inclusive hex command, except that packets that are matched against this filter item produce an Exclusive result for the filter list.

- Hexstring must consist of an even number of hexadecimal numbers. If you do not supply a full 32 hexadecimal numbers, ASCII blanks are padded to the 29th and 30th numbers and a wildcard is supplied as the 31st and 32nd (16th byte) numbers. A wildcard for a single byte can be specified by "??."

default include

Changes the default setting of the filter list to “inclusive.” This command indicates that, if no filter items of the filter list match the contents of the packet being considered for filtering, the filter list will evaluate to Inclusive. This is the default setting.

default exclude

Changes the default setting of the filter list to “exclusive.” This command indicates that, if no filter items of the filter list match the contents of the packet being considered for filtering, the filter list is evaluated as Exclusive.

delete *filter-item*

Deletes a filter item from the filter list.

- Filter-item is a decimal number representing a filter item that was previously created by the add command.

list

Displays information related to filter items in the specified filter-list.

```
NAME Filter List Name: nlist
NAME Filter List Default: Exclusive

Filter Item #   Type    Inc/Ex        Hostname       Last Char

      1         ASCII   Inclusive     EROS
      2         ASCII   Inclusive     ATHENA
      3         ASCII   Exclusive     FOOBAR
```

move *filter-item1 filter-item2*

Reorders filter items within the filter list. The filter item whose number is specified by filter-item1 is moved and renumbered to be just after filter-item2.

exit

Exits to the previous command prompt level.

Exit

Use the **exit** command to return to the previous prompt.

Syntax: exit

Example: `exit`

Chapter 11. Monitoring NetBIOS Filtering

This chapter summarizes and then explains the NetBIOS Filtering console commands. These commands let you monitor and display NetBIOS Filter information as an added feature to ASRT bridging. Console commands are entered at the `NetBIOS console>` prompt.

Changes you make at the `NetBIOS>` console prompt affect both bridging and DLSw.

Included in this chapter are the following sections:

- "Accessing the ASRT and the DLSw NetBIOS Filtering Console Environments"
- "NetBIOS Filtering Monitoring Commands"

Accessing the ASRT and the DLSw NetBIOS Filtering Console Environments

To display the `NetBIOS>` console prompt from the ASRT monitoring environment:

```
+ protocol asrt

ASRT> netbios
NetBIOS Support User Console

NetBIOS console> list filter name or byte

NetBIOS filter>
```

To display the `NetBIOS>` console prompt from the DLSw monitoring environment:

```
+ protocol dls
DLSw> netbios
NetBIOS Support User Console

NetBIOS console> list filter name or byte
NetBIOS filtering

NetBIOS filter>
```

NetBIOS Filtering Monitoring Commands

Table 11-1 lists the NetBIOS filtering commands.

Table 11-1. NetBIOS Filtering Monitoring Commands Summary

Command	Function
List	Displays all information concerning created filters.
Exit	Exits the NetBIOS Filtering console process and returns to the previous ASRT prompt level.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: `?`

or

`list ?`

List

Use the **list** NetBIOS Filtering command to display all information concerning created filters.

Syntax: list byte-filter-lists
filters
name-filter-lists

byte-filter-lists

Displays information related to filter items in the specified byte-filter-list.

Example: **`list byte-filter-lists`**

```
BYTE Filter-List Name:  Engineering
BYTE Filter-List Default:  Exclusive

Filter Item # Inc/Ex     Byte Offset  Pattern      Mask

1             Inclusive     14        0x123456    0xFFFF00
2             Exclusive      0        0x9876      0xFFFF
3             Exclusive     28        0x1000000   0xFF00FF00
```

Filter Item# Specifies the filter item number of the filter item. Filter items are evaluated in numerical order when determining the Inclusive/Exclusive status of the filter list.

Inc/Ex Specifies the default status of the filter item.

Byte-offset Specifies the number of bytes (in decimal) to offset into the packet being filtered. This starts at the NetBIOS header of the packet.

Pattern The hexadecimal number used to compare with the bytes starting at the byte-offset of the NetBIOS header. Syntax rules for hex-pattern include no 0x in front, a maximum of 32 numbers, and an even number of hex numbers.

Mask If present, must be the same length as hex-pattern and is logically ANDed with the bytes in the packet, starting at byte-offset, before the result is compared for equality with hex_pattern. If the hex-mask argument is omitted, it is considered to be all binary 1s.

filters

Displays information related to all configured filters.

Example: **`list filters`**

```
NetBIOS Filtering: Enabled

    Port #      Direction      Filter List Handle(s)   Pkts Filtered

       1          Input       valencia                        0
       2          Output      raleigh                         0
```

name-filter-lists

Displays information related to filter items in the specified name-filter-list.

Example: `list name-filter-lists`

```
NAME Filter List Name: nlist
NAME Filter List Default: Exclusive


Filter Item #   Type    Inc/Ex     Hostname   Last Char

 1              ASCII   Inclusive  EROS       <0x03>
 2              ASCII   Inclusive  ATHENA
 3              ASCII   Exclusive  FOOBAR
```

Filter Item# Specifies the filter item number of the filter item. Filter items are evaluated in numerical order when determining the Inclusive/Exclusive status of the filter list.

Inc/Ex Specifies the default status of the filter item.

Type “ASCII” indicates a host-name filter item added as ASCII characters. “Hex” indicates a host name filter item added as hexadecimal numbers

Host-name ASCII string up to 16 characters long. A question mark (?) can be used in hostname to indicate a single-character wildcard. An asterisk (*) can be used as the final character of hostname to indicate a wildcard for the remainder of the hostname. If hostname contains fewer than 15 characters, it is padded to the 15th character with ASCII spaces. Hostname can contain any character but the following:

. / \ [ ] : | < > + = ; , <space>

Last char Used if host-name contains fewer than 16 characters. It is a hexadecimal number (with no 0x in front of it) which indicates the value to be used for the last character. If the LAST argument is not specified on a hostname less than 16 characters, then a “?” wildcard is supplied for the 16th character.

Exit

Use the **exit** command to return to the previous prompt.

Syntax: exit

Example: `exit`

Chapter 12. Using, Configuring, and Monitoring LAN Network Manager (LNM)

This chapter describes IBM's ASRT implementation of the LAN Network Manager (LNM). It includes the following sections:

- "About LNM"
- "Configuring LNM" on page 12-4
- "LNM Commands" on page 12-5

About LNM

LNM lets you manage token-ring networks interconnected by source route bridges. It lets you monitor the operation of rings, bridges, and individual ring stations.

Information collected by software agents on the bridge is available to LNM management stations. More specifically, LNM agents forward collected information via another agent called the LAN Reporting Mechanism (LRM), a proprietary IBM protocol. Information forwarding is done via an LLC2 connection to a LAN Network Manager station.

LNM Agents and Functions

The LNM agents and their functions include:

- Configuration Report Server (CRS) - reports ring topology changes and ring station status to LNM.
- Ring Parameter Server (RPS) - services requests from ring stations for ring parameter information including ring number, the soft error report timer value, and the physical location.
- Ring Error Monitor (REM) - collects error reports from ring stations and analyzes them. When thresholds are exceeded, REM may forward error information to LNM.
- LAN Reporting Mechanism (LRM) - controls the establishment of reporting links from LNM stations to the bridge agents. Also manages the transfer of information to and from the other agents over these links.

Figure 12-1 illustrates the connection between the IBM bridge, LNM agents, and the IBM LNM station.

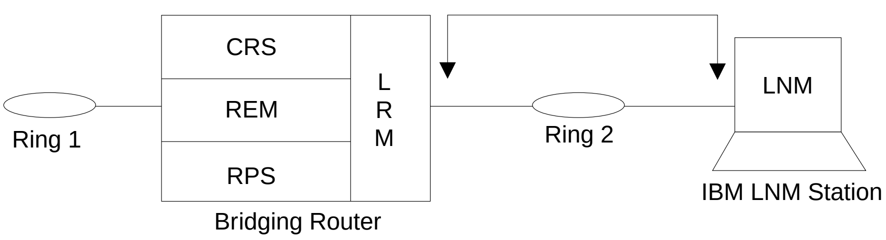


Figure 12-1. LNM Station and Agents

The following sections describe each LNM agent in more detail.

Configuration Report Server

At the request of LNM, CRS obtains and forwards ring station status to LNM. You can also use CRS to set ring station parameters and remove a station from the ring.

Configuration information generated by ring stations is forwarded to LNM. When LNM requests the status of a ring station, CRS builds and sends MAC frames to the station to obtain the information. CRS then sends the following frames to the ring station:

- Request Ring Station Address MAC frame
- Request Ring Station State MAC frame
- Request Ring Station Attachments MAC frame

When the ring station replies, CRS puts the information into a properly formatted LLC2 frame and forwards it to LNM.

CRS can also remove a ring station from the ring at the request of LNM. To remove a ring station, CRS sends a Remove Station MAC frame to the ring. CRS also returns a response to LNM indicating the success or failure of the removal.

When CRS receives a Report New Active Monitor MAC frame, it forwards the information to LNM. When a Report NAUN (Next Active Upstream Neighbor) Change MAC frame is received, this information is also reported. The CRS agent has its own functional address that ring station MAC layers can use to forward MAC frames to CRS.

Ring Parameter Server

RPS inserts ring stations onto the ring. When a ring station is newly inserted into the ring the following occurs:

- The new station sends a Request Initialization MAC frame to RPS for that ring. This MAC frame includes some information about the station.
- RPS responds with an Initialize Ring Station MAC frame containing the ring number and the interval of time to wait between sending Report Soft Error MAC frames. The information gleaned from the Request Initialization frame is passed to LNM so that it can maintain a database of all ring stations on the ring.
- RPS also responds to a request for status from LNM. The ring number, RPS version information and the soft error report timer value are returned to LNM.

The RPS function has an associated functional address for receiving the MAC frames that other ring stations send to it.

Ring Error Monitor

REM observes the operation of the attached token-ring by looking for hard errors and soft errors. It then reports these to the LRM and aids in isolating the cause of the errors. It does the following during hard error detection:

- Hard errors are detected on the ring by the receipt of Beacon MAC frames.
- Stations in the fault domain attempt to correct the problem by possibly removing themselves from the ring.
- REM determines if the hard error condition is corrected or not and then reports the results to LNM.

REM monitors soft errors as follows:

- Soft Error MAC frames are sent periodically by ring stations to REM to inform it of the number of times various intermittent faults, for example, CRC errors and frequency errors, occur.
- When the number of soft errors for a station a certain threshold, REM reports this condition to LNM.
- REM also monitors the Report Soft Error MAC frames for receiver congestion conditions. Receiver congestion indicates that a ring station discarded frames due to a shortage of receive buffers.
- If the number of times a station reports receiver congestion exceeds a certain threshold, REM reports this condition to LNM. When the receiver congestion condition returns to normal, LNM is notified that the receiver congestion condition has ended.

LAN Reporting Mechanism

LRM controls the connection of LNM to the agents. LRM establishes reporting links between itself and each connected LNM. A *reporting link* is an LLC2 connection between LNM and LRM.

All communication between LNM and the agents is done via a reporting link. LRM passes management data to and from the appropriate agents to the reporting links. Up to four reporting links are supported. One is designated the *controlling link* and the other three are designated as *observing links*.

An LNM connected via the controlling link can perform all available operations. LNMs connected by observing links can perform only a limited subset of the available operations.

LNM Configuration Restrictions

IBM 2216 supports multiport Token-Ring and two Token-Ring configurations

The LNM agent and the LNM station always assume that messages are being passed on a two-party model. LNM is enabled, however, on a per-bridge port basis to be consistent with the existing configuration.

This is what is meant by a multiport configuration. LNM can be enabled on any source-routing token-ring bridge port. In other words, an instance of LNM is created for each port upon which LNM is enabled.

With the exception of a two-token-ring configuration, the other port in a two-port model is always designated by a bogus address. This address corresponds to something that is not a real token-ring interface. It can correspond to a virtual ring or a serial line interface.

Only in the case where the 2216 bridge has two source routing token-ring ports is the other port in the two-port model bridge a token-ring with a real address.

To obtain the MAC addresses needed to configure the LNM Manager, enter **`list lnm ports`**.

The LAN Bridge Server (LBS) can report packets-forwarded and packets-discarded performance data statistics when requested by the manager station. Remote configuration updates from the manager station are not supported.

Logical Link Class 2 Support

In LANs, the data link layer comprises two sublayers: the medium access control (MAC) and the link layer control (LLC). LLC provides two types of service:

- LLC1 (Type 1) - an unacknowledged connectionless service
- LLC2 (Type 2) - a set of connection-oriented service

LAN Network Manager (LNM) requires LLC2 connection-oriented services. LLC2 provides capabilities for:

- Initiating new data link connections
- Managing data link connections
- Exchanging data in sequential order (in a guaranteed fashion)
- Executing a level of flow control on the established connections
- Terminating link connections upon request from the service user or unrecoverable link errors.

The LLC sublayer conforms to the IEEE 802.5 standard.

Configuring LNM

This section summarizes the procedure for basic configuration of the LNM feature on your bridging router.

1. Obtain the MAC address required for network manager software.

   Enter the **list lnm ports** command at the ASRT> prompt to obtain the MAC addresses required by the Network Manager software running on the Network Manager Station. For example:

   ```
   ASRT> list lnm ports
   Port Number [1]? 1
   Port 1
   LNM Agents Enabled: RPS CRS REM
   Reporting Link          State          LNM Station Address
   0                       ACTIVE         10:00:5A: F1:02:37
   1                       AVAILABLE
   2                       AVAILABLE
   3                       AVAILABLE
   MAC Addresses to use when configuring LNM Manager:
   00:00: C9:08:35:47
   40:00: D9:08:35:47
   LNM not enabled on port 4
   LNM not enabled on port 5
   ```

   The MAC addresses displayed (shown in bold in the example) are used by the Network Manager to configure it to the LNM agents present in the router.

   Note: These addresses must be entered exactly as they appear in the output, otherwise LNM will not configure correctly.

2. Enable the LNM agents on the router. Type **enable lnm** at the LNM config> prompt to enable the LNM agents on the desired port of your bridging router. For example:

   ```
   LNM config>enable lnm
   Port Number [1]? 1
   ```

   The default setting has all LNM agents enabled.

3. Check the configuration by displaying enabled LNM agents. Type **`list port`** at the `LNM config>` prompt to display which LNM agents are enabled on your configured port. For example:

```
LNM config>list port
Port Number [1]? 1
LNM Agents Enabled: RPS CRS REM
```

LNM Commands

This section describes the LNM configuration and monitoring commands. These commands allow you to configure and monitor network parameters for the LNM.

Enter configuration commands at the `LNM config>` prompt. Access this prompt as follows:

```
Config>protocol asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>lnm
LNM configuration
LNM config>
```

Enter monitoring commands at the `LNM>` prompt. Display this prompt as follows:

```
+protocol asrt
ASRT>lnm
LNM>
```

Table 12-1 lists the LNM commands.

Table 12-1. LNM Command Summary

Command	Function
?(Help)	Lists all of the LNM commands or the options associated with specific commands. This command is used for both configuration and monitoring.
Disable	Disables all LNM agents on a specified port or specified LNM agents (RPS, CRS, or REM) on a specified port. Disables the setting of certain LNM parameters from the remote LNM application linked to the bridge. Applies globally to all instances of LNM within the bridge. This command is used for configuration only.
Enable	Enables all LNM agents on a specified port or specified LNM agents (RPS, CRS, or REM) on a specified port. Enables the setting of certain LNM parameters from the remote LNM application linked to the bridge. Applies globally to all instances of LNM within the bridge. This command is used for configuration only.
List	Displays the LNM agents that have been enabled for the specified port. Displays the passwords configured for the bridge. This command is used for both configuration and monitoring.
Set	Sets the password for the specified reporting link number. This command is used for configuration only.
Exit	Returns to the previous prompt. This command is used for both configuration and monitoring.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

Disable

Use the **disable** command to disable all LNM agents (RPS, CRS, or REM) on a specified port.

This command also disables the setting of the reporting link passwords from the remote LNM application linked to the bridge.

Syntax: disable

agent *port#*
lnm ...
configuration-remote-change

agent *port#*

Disables the specified LNM agent (RPS, CRS, or REM) on the specified port. If the port is not configured then the message `LNM not configured for port XX` is displayed, and the command has no effect.

Example: **`disable REM 1`**

lnm

Disables LNM on the specified port. If the port is not configured for LNM, the message `LNM not configured for port XX` is displayed, and the command has no effect.

Example: **`disable lnm`**

```
Port number [1]? 1
LNM not configured for Port 1
```

configuration-remote-change

Disables the setting of the reporting link passwords from the remote LNM application linked to the bridge. This command applies globally to all instances of LNM within the bridge.

Example: **`disable configuration-remote-change`**

```
CONFIGURATION-REMOTE-CHANGE: disabled
```

Enable

Enables all LNM agents on a specified port or enables specified LNM agents (CRS, REM, or RPS) on a specified port.

If the interface is not a token-ring then the message `Port number XX is not token-ring` is displayed and the command has no effect.

If the port is not configured, then the message `Port number XX does not exist` is displayed and the command has no effect.

If the specified agent is already enabled for the specified port the message `Already enabled` is displayed.

This command also enables the setting of the reporting link passwords from the remote LNM application linked to the bridge.

Syntax: enable

agent *port#*
lnm . . .
configuration-remote-change

agent *port#*

Enables the specified LNM agent (RPS, CRS, or REM) on the specified port.

Example: `enable CRS 1`

lnm *port#*

Enables all LNM agents on the specified port.

Example: `enable lnm`

```
Port Number [1]? 1
```

configuration-remote-change

Enables the setting of the reporting link passwords from the remote LNM application linked to the bridge. The default setting disables the setting of LNM configuration parameters remotely.

This command applies globally to all instances of LNM within the bridge.

Example: `enable configuration-remote-change`

```
CONFIGURATION-REMOTE-CHANGE: Enabled
```

List (configuration command)

Displays the LNM agents enabled for the specified port, and also displays passwords that have been configured for the bridge.

Syntax: list

password
port . . .

password

Displays the passwords that have been configured for the reporting links of the bridge. Displays whether or not the passwords can be changed by the remote LNM application.

Example: `list password`

```
Reporting Link     Password
0             87654321
1             MADRAS
2             ABC1234
3             123ABC
CONFIGURATION-REMOTE-CHANGE: Disabled
```

port *port#*

Displays the LNM agents enabled for the specified port if the port is a token-ring port supporting Source Routing Bridging.

Example: `list port`

```
Port Number [1]? 1
LNM Agents Enabled: RPS CRS REM
```

List (monitoring command)

Syntax: list

bridge
LNM ports
source-routing configuration

bridge

Displays whether LNM is enabled on a specific port.

Example: **`list bridge`**

```
Bridge ID (prio/add):  32768/00-00-00-00-00-38
Bridge state:          Enabled
UB-Encapsulation:      Disabled
Bridge type:           SR-TB
Bridge capability:     ASRT
Number of ports:       5
STP Participation:     IEEE802.1d on TB ports and IBM-8209 on SR ports
Maximum
Port  Interface State MAC Address      Modes   MSDU  Segment  Flags
1  FDDI/0      Up  AA-00-04-00-26-14    T    4491           RD
2  TKR/0       Up  00-00-93-90-4C-F7    T    2096           RD
3  TKR/1     Down  00-00-00-00-00-00 SR        0 223        RD,LE
4  FDDI/1    Down  00-00-00-00-00-00    T       0           RD
5  Eth/0     Down  AA-00-04-00-26-14            0           RD
Flags: RE =IBMRT PC behaviour Enabled, RD = IBMRT PC behaviour Disabled
LE = LNM Enabled, LD = LNM Disabled, LF = LNM Failed
SR bridge number:      8
SR virtual segment:    812
Adaptive segment:      214
```

lnm ports

Displays information about the configuration of the LNM enabled on the bridging router.

Example: **`list LNM ports`**

```
LNM not enabled on port 1
LNM not enabled on port 2
Port 3
LNM Agents Enabled: RPS CRS REM
Reporting Link          State          LNM Station
Address
0                 AVAILABLE
1                 AVAILABLE
2                 AVAILABLE
3                 AVAILABLE
MAC Addresses to use when configuring LNM Manager:
00:00:00:00:00:00
00:00:00:00:00:00
LNM not enabled on port 4
LNM not enabled on port 5
```

source-routing configuration

Displays whether LNM is enabled on a specific port.

Example: **`list source-routing configuration`**

```
Bridge number:               8
Bridge state:                Enabled
Maximum STE hop count        14
Maximum ARE hop count        14
Virtual segment:             812
Port  Segment  Interface  State    MTU   STE Forwarding   LNM
   3  223      TKR/1      Enabled  4399  Auto             ENA
   -  214      Adaptive   Enabled  1470  Yes
```

Set

Sets the password for the specified reporting link number. The link number can be 0, 1, 2, or 3. Link 0 is used for the controlling link. Links 1, 2, and 3 are used for observing links.

The password should consist of six to eight characters, and must match the password used by LNM when it establishes a reporting link with the bridge. If the password is not set for a link, it defaults to the string 00000000.

Syntax: set password *link#* *password*

Example: **`set password`**

```
Link Number [0]? 1
Enter new password :  [ABCDEFGH]? guesswho
```

Exit

Use the exit command to return to the ASRT prompt.

Syntax: exit

Example: **`exit`**

Chapter 13. Configuring TCP/IP Host Services

This chapter describes how to configure the TCP/IP Host Services (TCP/IP Host) protocol and how to use the TCP/IP Host configuration commands. The chapter includes the following sections:

- “Basic Configuration Procedures”
- “Accessing the TCP/IP Host Configuration Environment”
- “TCP/IP Host Configuration Commands” on page 13-2

See “TCP/IP Host Services (Bridge-Only Management)” on page 3-3 if you want to know more about why you would use TCP/IP host services.

Do not use this chapter if you are configuring the router for IP routing; instead, refer to Chapter 16, “Using and Configuring IP.”

Basic Configuration Procedures

The following sections describe the basic configuration procedures for enabling TCP/IP Host Services on your 2216.

Setting the IP Address

To minimally configure TCP/IP Host services, assign the 2216 an IP address by using the **set ip-host** command. This IP address is associated with the 2216 as a whole, instead of being associated with a single interface.

Adding a Default Gateway

The 2216 uses its default gateway to communicate with hosts and gateways that are not on the bridged network to which the 2216 is directly connected. The 2216 can dynamically learn its default gateway using either ICMP Router Discovery (see the **enable router-discovery** command in this chapter) or RIP (see the **enable rip-listening** command in this chapter). You also can statically specify one or more default gateways by using the **add default gateway** command. The 2216 uses only one default gateway at a time; any additional default gateways are used for backup.

To save the assigned IP address and default gateway information, exit from the `TCP/IP-Host config>` prompt to the `Config>` and use the **restart** command. After restarting the 2216, return to the `TCP/IP-Host config>` prompt.

Enabling TCP/IP Host Services

After assigning and saving the 2216 IP address and default gateway information, use the **enable services** command to enable TCP/IP Host Services.

Accessing the TCP/IP Host Configuration Environment

To access the TCP/IP Host configuration environment, enter the following command at the `Config>` prompt:

```
Config> protocol iphost
TCP/IP-Host Services user configuration
TCP/IP-Host config>
```

© Copyright IBM Corp. 1997

Note: To configure Host services you cannot have any IP address configured on the interfaces. The router cannot be configured as a router for IP. The Host services are for bridging only.

TCP/IP Host Configuration Commands

This section summarizes and explains all the TCP/IP Host configuration commands. The TCP/IP Host configuration commands allow you to specify network parameters for the TCP/IP Host bridge. Restart the router to activate the configuration commands. Enter the TCP/IP Host configuration commands at the `TCP/IP-Host config>` prompt. Table 13-1 shows the commands.

Table 13-1. TCP/IP Host Configuration Commands Summary

Command	Function
? (Help)	Lists all of the TCP/IP Host configuration commands, or lists the options associated with specific commands.
Add	Adds a default-gateway.
Delete	Deletes a default-gateway.
Disable	Disables TCP/IP Host Services, router-discovery processes, and RIP listening.
Enable	Enables TCP/IP Host Services, router-discovery processes, and RIP listening.
List	Lists the current TCP/IP Host configuration.
Set	Sets the 2216’s IP address.
Exit	Exits the TCP/IP Host configuration process and returns to the CONFIG environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
LIST
SET
ADD
DELETE
ENABLE
DISABLE
EXIT
TCP/IP-Host config>
```

Add

Use the **add** command to add default gateways (that is, routers) to your configuration.

Default gateways are used when trying to send packets to IP destinations that are off the local connection. The routing table is then built up through redirect processing. An attempt is made to detect routers that disappear. If the 2216 has booted over the network (via TFTP/BootP), then the default gateway is configured using the information from the booting process.

Syntax: add default-gateway *def-gateway-IP-address*

Example: **`add default-gateway`**

```
Default-Gateway address [0.0.0.0]?  123.45.67.89
```

Delete

Use the **delete** command to delete default gateways from your 2216 configuration. Enter the IP address of the default gateway you want to remove after the **delete** command.

Syntax: delete default-gateway *def-gateway-IP-address*

Example: **`delete default-gateway`**

```
Enter address to be deleted [0.0.0.0]?  123.45.67.89
```

Disable

Use the **disable** command to disable the following TCP/IP functions:

- TCP/IP Host Services
- Router-discovery processes
- RIP listening

Syntax: disable rip-listening
router-discovery
services

rip-listening
Disables the building of routing table entries that have been gathered by listening to the RIP protocol. By default, RIP-listening is disabled.

Example: **`disable rip-listening`**

router-discovery
Disables the ability to learn default gateways by receiving ICMP Router Discovery messages. By default, router discovery is enabled.

Example: **`disable router-discovery`**

services
Disables the TCP/IP Host Services protocol entirely. If IP routing is not enabled, TCP/IP Host Services is enabled by default.

Example: **`disable services`**

Enable

Use the **enable** command to enable the following TCP/IP functions:

- TCP/IP Host Services
- Router discovery processes
- RIP listening

Syntax: enable rip-listening
router-discovery
services

rip-listening

Enables the building of routing table entries that have been gathered by the bridge “listening” to the RIP protocol. RIP-listening is disabled by default.

Example: `enable rip-listening`

router-discovery

Enables the learning of default gateways through reception of ICMP Router Discovery messages. By default, router discovery is enabled.

Example: `enable router-discovery`

services

Enables the TCP/IP Host Services protocol. If IP routing is not enabled, TCP/IP Host Services is enabled by default.

Example: `enable services`

List

Use the **list** command to display information about the current TCP/IP Host configuration.

Syntax: list all

Example: `list all`

```
IP-Host IP address : 128.185.142.1
Address mask : 255.255.255.0

Default Gateway IP-address(es)
128.185.142.47

TCP/IP-Host Services Enabled.

RIP-LISTENING Disabled.

Router Discovery Enabled.
```

IP-Host IP address Displays the current IP-Host IP address.

Address mask Displays the current IP-Host IP subnet address mask.

Default Gateway IP-address(es) Displays the current default gateway IP address.

TCP/IP Host Services Displays whether TCP/IP Host Services is enabled or disabled.

RIP-LISTENING Displays whether RIP-LISTENING is enabled or disabled.

Router Discovery Displays whether Router Discovery is enabled or disabled.

Set

Use the **set** command to set the 2216’s IP address. You must assign the 2216 an IP address before enabling TCP/IP Host Services.

Note: If the IP address is not already configured, it is set (by default) using boot information. This process applies only if the 2216 is a network host operating as an IP host.

Syntax: set IP-Host address *IP-host-address*

Example: `set ip 123.45.67.89`

```
Address mask [255.255.0.0]?
IP-Host Address set.
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 14. Monitoring TCP/IP Host Services

This chapter describes how to monitor the TCP/IP Host Services on the 2216. The chapter includes the following sections:

- “Accessing the TCP/IP Host Console Environment”
- “TCP/IP Host Console Commands”

Accessing the TCP/IP Host Console Environment

To access the TCP/IP Host console environment, enter the following command at the + (GWCON) prompt:

```
+ protocol iphost
TCP/IP-Host Services user configuration
TCP/IP-Host>
```

TCP/IP Host Console Commands

This section summarizes and then explains the TCP/IP Host console commands. These commands allow you to view parameters and enter information requests from the active console. Enter these commands at the `TCP/IP-Host>` prompt. Table 14-1 shows the commands.

Table 14-1. TCP/IP Host Console Commands Summary

Command	Function
? (Help)	Lists all of the TCP/IP Host console commands, or lists the options associated with specific commands.
Dump	Displays the current IP routing table. One line is printed for each destination.
Interface	Displays the IBM 2216’s IP address.
Ping	Continuously pings a given destination, printing a line for each response received.
Traceroute	Displays the hop-by-hop route to a given destination.
Routers	Displays the list of all IP routers known to the 2216.
Exit	Exits the TCP/IP Host console process and returns to the GWCON environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
DUMP routing address
INTERFACE address
PING address
TRACEROUTE address
ROUTERS
EXIT
```

Dump

Use the **dump** command to display the current IP routing table. One line is printed for each destination. Many of the entries that are displayed are the result of ICMP redirects.

Syntax: dump

Example: **dump**

```
Type   Dest net         Mask      Cost  Age  Next hop(s)

Stat   0.0.0.0          00000000   0     0   128.185.142.47
Dir*   128.185.142.0    FFFFFF00   1     0   TKR/0


Default gateway in use.
Type Cost Age  Next hop
Stat 0    0    128.185.142.47

Routing table size: 768 nets (43008 bytes), 2 nets known
```

Type (route type)
Indicates how the route was derived:

RIP - the route was learned through the RIP protocol.

Stat - a statically configured route.

Dest net
Displays the IP address of the destination network/subnet.

Mask
Displays the IP address mask.

Cost
Displays the Route Cost.

Age
Displays the time that has elapsed since the routing table entry was last refreshed for RIP and BGP routes.

Next Hop
Displays the IP address of the next router on the path toward the destination host. Also displayed is the interface type used by the sending router to forward the packet.

Default gateway
Displays the IP address of the default gateway along with the route type, cost, age, and next-hop information associated with that entry.

Routing table size
Displays the current size (in networks and bytes) of the current table. Also identifies the number of networks (nets) known to the host.

Interface

Use the **interface** command to display the IBM 2216's IP address. When TCP/IP Host Services are running over the bridge, a single address is displayed on the console as Bridge/0.

Syntax: interface

Example: **interface**

```
Interface   IP Address(es)      Mask

  TKR/0     128.185.142.16      255.255.255.0

              or

  BDG/0     128.185.142.16      255.255.255.0
```

Interface Displays a single address as BDG/0 when TCP/IP Host Services are running over the bridge. When services are disabled, interfaces with their corresponding numbers are displayed.

IP Address Displays the IP address of the TCP/IP Host Services interface.

Mask Displays the IP address subnet mask.

Ping

Use the **ping** command to make the router send ICMP Echo Requests to a given destination once a second ("pinging") and watch for a response. This command can be used to isolate trouble in an internetwork environment.

This process is done continuously, incrementing the ICMP sequence number with each additional packet. Matching received ICMP Echo responses are reported with their sequence number and the round trip time. The granularity (time resolution) of the round trip time calculation is platform specific, and usually is around 20 milliseconds.

To stop the pinging process, type any character at the console. At that time, a summary of packet loss, round trip time, and number of unreachable ICMP destinations will be displayed.

When a multicast address is given as destination, there may be multiple responses printed for each packet sent, one for each group member. Each returned response is displayed with the source address of the responder.

Note: The size of the ping (number of data bytes in the ICMP message, excluding the ICMP header) is 56 bytes, and the TTL used is 60. The size of the ping (number of data bytes in the ICMP message, excluding the ICMP header), TTL value, and frequency of pinging are all user configurable. The default values are a size of 56 bytes, a TTL of 64 seconds, and a frequency of 1 ping per second.

Syntax: ping *destination source size ttl frequency*

Example: **`ping 128.185.142.11  128.185.142.06 56  60  1`**

```
PING 128.185.142.11: 56 data bytes
56 bytes from 128.185.142.11: icmp_seq=0. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=1. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=2. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=3. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=4. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=5. time=0. ms

----128.185.142.11 PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
```

Traceroute

Use the **traceroute** command to display the entire path to a given destination, hop by hop. For each successive hop, the traceroute command sends out three probes and prints the IP address of the responder along with the round trip time associated with the response. If a particular probe receives no response, an asterisk (*) is printed. Each line in the display relates to this set of three probes, with the leftmost number indicating the distance from the router executing the command (in router hops).

The traceroute is complete when the destination is reached, an ICMP Destination Unreachable message is received, or the path length reaches 32 router hops.

Syntax: traceroute   *interface-address*

Example: **`traceroute 128.185.142.239`**

```
TRACEROUTE 128.185.142.239: 56 data bytes
   1 128.185.142.7 16 ms 0 ms 0 ms
   2 128.185.123.22 16 ms 0 ms 16 ms
   3 * * *
   4 * * *
   5 128.185.124.110 16 ms ! 0 ms ! 0 ms !
```

TRACEROUTE Displays the destination area address and the size of the packet being sent to that address.

1 The first trace showing the destination's NSAP and the round trip time it took the packet to reach the destination and return. The packet is traced three times.

Destination unreachable Indicates that no route to the destination is available.

*1 * * *
2 * * ** Indicates that the router is expecting some form of response from the destination, but the destination is not responding.

When a probe receives an unexpected result (see the previous output example), several indicators can be printed. These indicators are explained in the following table.

!N Indicates that an ICMP Destination Unreachable (net unreachable) has been received.

!H Indicates that an ICMP Destination Unreachable (host unreachable) has been received.

!P Indicates that an ICMP Destination Unreachable (protocol unreachable) has been received.

! Indicates that the destination has been reached, but the reply sent by the destination has been received with a TTL of 1. This usually indicates an error in the destination, prevalent in some versions of UNIX, whereby the destination is inserting the probe's TTL in its replies. This unfortunately leads to a number of lines consisting solely of asterisks before the destination is finally reached.

Routers

Use the **routers** command to display the list of all IP routers that are known to the 2216. Routers can be learned through:

- Static configuration (using the **add default-gateway** command explained on page 13-2).
- Received ICMP redirects
- ICMP Router Discovery messages (if configured)
- RIP updates (if configured)

Each router is listed with its origin, its priority (used when selecting the default route), and its lifetime (the number of seconds before the router will be declared invalid unless it is heard from again).

Syntax: routers

Example: `routers`

Exit

Use the **exit** command to exit the TCP/IP Host console process and return to the GWCON environment.

Syntax: exit

Example: `exit`

Part 2. Configuring and Monitoring Router Protocols

Chapter 15. Overview of Classical IP Over ATM 15-1
Benefits of Classical IP 15-1
Components of Classical IP 15-2
Timeouts and Refresh 15-2
IP Addresses and CIP components 15-3
ATM Addresses of CIP components 15-3
Virtual Channel Connections 15-3
Key Configuration Parameters for Classical IP 15-4

Chapter 16. Using and Configuring IP 16-1
Basic Configuration Procedures 16-1
IP Filtering 16-7
Configuring the BOOTP/DHCP Forwarding Process 16-10
Configuring UDP Forwarding 16-12
IP Multicast Support 16-13
Accessing the IP Configuration Environment 16-16
IP Configuration Commands 16-16

Chapter 17. Monitoring IP 17-1
Accessing the IP Console Environment 17-1
IP Console Commands 17-1

Chapter 18. Using and Configuring IPX 18-1
IPX Overview 18-1
Configuring IPX 18-2
Optional Configuration Tasks 18-3
Accessing the IPX Configuration Environment 18-15
IPX Configuration Commands 18-15
Accessing the IPX Interface Filter Configuration Environment 18-30
IPX Interface Filter Configuration Commands 18-30

Chapter 19. Monitoring IPX 19-1
Accessing the IPX Console Environment 19-1
IPX Console Commands 19-1
IPX Interface Filter Monitoring Commands 19-14

Chapter 20. Using and Configuring ARP 20-1
ARP Overview 20-1
Inverse ARP Overview 20-3
Classical IP and ARP Over ATM Overview (RFC 1577) 20-4
IPX and ARP Over ATM Overview (RFC 1483) 20-10
Accessing the ARP Configuration Environment 20-10
ARP and Inverse ARP Configuration Commands 20-11
ARP Over ATM Configuration Commands 20-16

Chapter 21. Monitoring ARP 21-1
Accessing the ARP Console Environment 21-1
ARP Console Commands 21-2
ARP Over ATM Console Commands 21-6

Chapter 22. Using, Configuring, and Monitoring XTP 22-1
The X.25 Transport Protocol 22-1
DTE Address Wildcards 22-3
XTP Backup Peer Function 22-4
Configuring XTP 22-5
Configuration Procedures 22-5
XTP Configuration Commands 22-13
XTP Monitoring Commands 22-19

Chapter 23. Using and Configuring OSPF 23-1
The OSPF Routing Protocol 23-1
Configuring OSPF 23-4
Accessing the OSPF Configuration Environment 23-17
OSPF Configuration Commands 23-17

Chapter 24. Monitoring OSPF 24-1
Accessing the OSPF Console Environment 24-1
OSPF Console Commands 24-1

Chapter 25. Configuring SNMP 25-1
Accessing the SNMP Configuration Environment 25-1
SNMP Configuration Commands 25-1

Chapter 26. Monitoring SNMP 26-1
Accessing the SNMP Console Environment 26-1
SNMP Console Commands 26-1

Chapter 27. Using and Configuring DLSw 27-1
About DLSw 27-1
Using DLSw Features 27-3
Setting up DLSw 27-18
Sample DLSw Configuration 27-23
Accessing the DLSw Configuration Environment 27-35
Pre-Configuration Requirements 27-36
DLSw Configuration Commands 27-37

Chapter 28. Monitoring DLSw 28-1
Accessing the DLSw Console Environment 28-1
DLSw Console Commands 28-1

Chapter 15. Overview of Classical IP Over ATM

Note: See the glossary for definitions of the acronyms and terms used in this chapter. Classical IP over ATM is simply an extension of the current IP paradigm. That is, IP is independent of the medium over which it travels. An IP subnet is a group of Class A, B, or C hosts that share a common network and subnetwork portion of their IP address. The host portion of the IP address is unique for each station on the subnet. These are the characteristics of the subnet:

- Traffic from a host can be sent to any other host on the subnet.
- Traffic destined for a host outside of the subnet must pass through a gateway or router that attaches to more than one subnet.
- All members of the subnet use the same Maximum Transmission Unit (MTU) size. For classical IP, this value defaults to 9180 bytes.

Benefits of Classical IP

The Internet Engineering Task Force (IETF) has standardized a solution for sending IP traffic over an ATM interface.[1] The design described in this standard strives to keep the ATM infrastructure transparent to IP. Most applications that run today in a LAN or WAN environment will see no difference in functionality; however, their performance and throughput gains can be substantial.

In addition to the high link speeds that ATM provides, Classical IP (CIP) requires fewer framing bytes than, for example, LANs, which contain source and destination MAC addresses. Therefore, less bandwidth is used for overhead bytes, and more is used for data. In addition, no broadcast traffic is required for the resolution of ARP frames. In a broadcast environment, ARP traffic can adversely affect all stations in the subnet.

In CIP, the ARP traffic affects only the ARP Server and the client requesting the information. Other stations on the subnet are unaffected by this traffic. Even non-broadcast traffic on a shared medium such as Token-Ring or Ethernet precludes other stations from using that medium for discrete amounts of time. In CIP, independent channels are established between hosts having the conversation. These channels can be established with traffic parameters that protect the conversation from being impacted by other conversations.

The same benefits from simplifying moves, adds, and deletes that was described for ELANs apply to the CIP logical IP subnet (LIS). Membership is not based on physical location. Logically related stations are grouped in the same LIS. The ease with which a client can register to the ARP Server makes additions and changes trivial. Deletion from a subnet occurs naturally as the ARP Server ages its entries.

While all members of a LIS must support the Classical IP model, the router can route between subnets that are CIP-based and subnets that are based on LAN emulation (LE). Some equipment can be more adept at CIP, while other equipment

[1] The standard is “Classical IP and ARP over ATM,” RFC 1577, Hewlett-Packard Laboratories, Jan. 1994.

can be more adept at LANE. The flexibility of the router allows equipment to be utilized in the most effective manner.

Finally, investments in a CIP solution are protected. Enhancements to the IP over ATM work in the IETF will provide continual growth in function and performance. Distributed ARP Servers, Next Hop Routing Protocol (NHRP), Multicast Address Resolution Service (MARS), resource ReSerVation Protocol (RSVP), and other work that is being defined in the IETF will provide continual growth in functionality and performance.

Components of Classical IP

The LIS contains all of the properties of a normal IP subnet whether it is Ethernet, Token-Ring, or Frame Relay. However, because ATM is a Non-Broadcast Multiple Access (NBMA) network, the existing broadcast method for resolving addresses cannot be performed. ARP Servers and ARP Clients were developed to solve this problem.

Within the CIP model, there are two forms of requests and replies: ATMARP request and replies, which are also referred to simply as ARPs, and InATMARP request and replies. InATMARP is used determine the IP and ATM addresses of the entity at the other end of a VCC. ATMARP is used to request the ATM address associated with a particular IP address.

One ARP Server is defined per LIS. The server maintains the translation of IP addresses to ATM addresses. The server allows clients to register by accepting incoming VCCs and querying the client (with an InATMARP request) for the appropriate mapping information, which consists of the IP and ATM addresses of the client. The ARP Server also responds to ATMARP requests for ATM addresses corresponding to IP addresses specified by the client. Finally, the ARP server updates its tables by aging its ARP entries and managing incoming VCCs.

The ARP Client is the entity that always places calls. As a client is initialized, it places a call to the ARP Server, and, through the exchange of InATMARP requests and replies, registers with the ARP Server. When a client has traffic to transmit to another client on the LIS, it sends an ARP request, which contains the target IP address, to the ARP Server. If the server finds the target ATM address in its table, it sends that address back in its reply; if the server cannot find the ATM address, it sends back a negative acknowledgement (NAK) reply. When the client receives the ATM address, it then uses this address to place a call to the target client. IP datagrams then traverse this VCC.

Timeouts and Refresh

Both clients and servers age their ARP table entries. Once the timer expires, these ARP entries are deleted. If traffic is flowing when an ARP entry gets aged out, that traffic will cease until a new ARP entry is created.

To avoid an interruption in service, the router provides an automatic refresh option. This option allows the router client to transmit either an ARP to the ARP Server or an InATMARP to the target client some time before the ARP entry expires. If the target replies, then the timer of the ARP entry is reset. If the target does not reply, then the entry is deleted. The ARP Server automatically sends out an InATMARP message before aging an entry out of its table. The router CIP clients and ARP Servers have default aging periods of 5 minutes and 20 minutes respectively. These times are configurable for each LIS.

IP Addresses and CIP components

IP addresses are key to IP routing.

Helpful Tip: When you configure the router, the act of adding an IP address to an ATM interface automatically creates a CIP client.

You must then specify whether the router is also to act as the ARP Server for the LIS. An MSS ARP Server never exists without a paired client and each LIS has one client/server pair. The router supports up to 32 LISs per ATM interface.

Creation of an IP address on the router implies packet forwarding behavior; the router forwards packets between subnets even when no routing protocol, such as Open Shortest Path First (OSPF), is configured. Furthermore, if a packet is sent to the router and the destination of the packet is not the router, but the destination is on the same subnet as the source, the router sends an Internet Control Message Protocol (ICMP) redirect message to the originator, and forwards the packet to the correct host.

ATM Addresses of CIP components

In general, ATM addresses must be unique among CIP components; however, on the router, client/server pairs share an ATM address, so that a single connection can used for both control and data traffic. The ESI and selector portions of the ATM address of a CIP component can be explicitly configured or generated automatically at run-time. The ESI defaults to the MAC address burned into the ATM interface hardware. As in LAN Emulation, you can override the default by explicitly selecting one of the locally-administered ESIs defined for the ATM interface.

Important: If only a client is being created, then explicitly configuring the ESI or selector is not recommended; however, if a client/server pair is being created, then at least the selector should be specified in order to provide the server with a fixed address that can be configured at all the clients on the LIS.

The client/server pair is created when you designate the ATM client as an ATM ARP server during configuration. When there is a client/server pair, the ARP server in that pair is local, that is, it is located in the router rather than remotely on the ATM network.

Virtual Channel Connections

The router implementation of Classical IP supports both switched virtual circuits (SVCs) and permanent virtual circuits (PVCs). SVCs require a signaling protocol to establish connections. PVCs do not require a signaling protocol, but do require configuration in both the ATM network and end systems.

SVCs can be generated automatically through the address resolution and call setup procedures of Classical IP or an SVC can be explicitly configured. Automatic SVCs are brought up and torn down by the ARP subsystem as required for sending IP traffic. A configured SVC is brought up during initialization and kept up indefinitely.

PVCs and configured SVCs do not require an ARP Server. That is, a LIS could consist of hosts that were interconnected only by configured information. While these techniques can prove useful in small networks, the amount of manual configuration can quickly become prohibitive in larger networks.

Control channels are connections from a client to a server; data channels are connections from one client to another. The attributes of both control channels and data channels can be tailored to meet specific user needs. For example, *Quality of Service* characteristics can be specified for each LIS by configuring VCC traffic parameters such as Peak and Sustained Rates.

Key Configuration Parameters for Classical IP

Due to the simplicity of CIP, very few configuration parameters are required. The information required for a client-only configuration is:

1. IP address and Subnet mask
2. ATM address of the ARP Server

Configuration of a client/server pair requires:

1. IP address and Subnet mask
2. Answering *yes* to the question asking whether this client is also a server
3. Specifying an explicit selector for the ATM address of the server

The Maximum AAL-5 Service Data Unit (SDU) Size for CIP components does not generally need to be configured because the default of 9188 bytes is usually appropriate. However, if you need to change the SDU size, you will need to understand the relationships between the Max SDU Size for CIP components, the Max AAL-5 SDU Size for the ATM interface, and the CIP Maximum Transmission Unit (MTU) Size.

The Max AAL-5 SDU Size for CIP components can be configured for a client, but the value set for the client cannot be greater than the Max AAL-5 SDU Size for the ATM interface, which defaults to 9234 bytes. Although the Max CIP SDU Size can be configured for a client, the value set for one client can affect the MTU size for all the clients on the ATM interface. All CIP clients on the same ATM interface share a common MTU Size that is dependent on the Max SDU Sizes: the CIP MTU Size is set to “the smallest CIP Max SDU Size – 8” (CIP frames have an 8 byte header). Consequently, all LISs associated with a given ATM interface must have the same MTU. Therefore, care should be excercised when altering the Max CIP SDU Size.

Chapter 16. Using and Configuring IP

This chapter describes how to configure the Internetwork Protocol (IP) and how to use the IP configuration commands. Included are the following sections:

- "Basic Configuration Procedures"
- "Configuring the BOOTP/DHCP Forwarding Process" on page 16-10
- "Configuring UDP Forwarding" on page 16-12
- "IP Multicast Support" on page 16-13
- "Accessing the IP Configuration Environment" on page 16-16
- "IP Configuration Commands" on page 16-16

Basic Configuration Procedures

This section outlines the initial steps required to get the IP protocol up and running. Details about making further configuration changes are covered in other sections of this chapter. Details on individual configuration commands are covered in the command section of this chapter. The following list outlines the initial configuration tasks to bring up IP on the router. After completing these tasks, you must restart the router for the new configuration to take effect.

1. Access the IP configuration environment. (See "Accessing the IP Configuration Environment" on page 16-16.)
2. Assign IP addresses to network interfaces. (See "Assigning IP Addresses to Network Interfaces.")
3. Enable dynamic routing. (See "Enabling Dynamic Routing" on page 16-2.)
4. Add static routing information (if necessary). (See "Adding Static Routing Information" on page 16-4.)
5. Enable ARP subnet routing (if necessary). (See "Enabling ARP Subnet Routing" on page 16-6.)
6. Set up ARP parameters (if necessary). (See "Setting Up ARP Configuration" on page 16-6.)

   If RFC 1577 (Classical IP and ARP over ATM) is being used, additional ARP Server and ARP Client configuration may be required for each IP address added to this interface. This configuration is described in "ARP Over ATM Configuration Commands" on page 20-16.
7. Exit the IP configuration process.
8. Restart the router to activate the configuration changes.

The following sections discuss each configuration task in more detail.

Assigning IP Addresses to Network Interfaces

Use the IP configuration **add address** command to assign IP addresses to the network interfaces. The arguments for this command include the interface number (obtained from the `Config>` **list devices** command) and the IP address and its associated address mask.

In the following example, network interface 2 has been assigned the address 128.185.123.22 with the associated address mask 255.255.255.0 (using the third byte for subnetting).

```
IP Config> add address 2 128.185.123.22 255.255.255.0
```

The IBM 2216 allows multiple IP addresses to be configured for an interface as long as each address is for a different network/subnet. It is not valid to configure an interface with multiple IP addresses where only the host portion of the address is different.

IP allows you to use a serial line interface for IP traffic without assigning a real IP address to the line. However, you must still assign each serial line a pseudo IP address; this address is used by the router to refer to the interface but is never used externally. Use the **add address** command to assign the serial line an address of the form 0.0.0.n, where n is the interface number (again obtained from the `Config>` **list devices** command). This address format tells the router that the interface in question is an *unnumbered serial line*.

To enable IP on serial-line interface number 2 without assigning the interface an IP address, use the following command:

```
IP Config> add address 2 0.0.0.2
```

Enabling Dynamic Routing

Use the following procedures to enable dynamic routing on the router. The router software supports OSPF and RIP for interior gateway protocols (IGPs) as well as BGP, which is an external gateway protocol.

OSPF, RIP, and BGP may be used over ATM if LAN emulation is selected. If RFC 1577, Classical IP and ARP over ATM (sometimes referred to native IP over ATM) is selected, only OSPF and BGP may be used. In the latter case, the ATM network is treated as a Non-Broadcast Multiple Access (NBMA) network for configuration purposes.

All routing protocols can run simultaneously. However, most routers will probably run only a single routing protocol (one of the IGPs). The OSPF protocol is recommended because of its robustness and the additional IP features (such as equal-cost multipath and variable-length subnets) that it supports.

Setting the Routing Table Size

The routing table size determines the number of entries in the routing table from all sources, including dynamic routing protocols and static routes. The default size is 768 entries.

To change the size of the routing table, use the **set routing table-size** configuration command. Setting the routing table size too small results in routes being discarded. Setting it too large results in inefficient use of memory resources. After operation, use the console **dump** command to view the contents of the table and then adjust the size as necessary, allowing some room for expansion.

Enabling the OSPF Protocol

OSPF configuration is done via its own configuration console (entered via the `Config>` **protocol ospf** command). To enable OSPF, use the following command:

```
OSPF Config> enable OSPF
```

After enabling the OSPF protocol, you are prompted for size estimates for the OSPF link state database. This gives the router some idea how much memory must be reserved for OSPF. You must supply the following two values that will be used to estimate the size of the OSPF link state database:

- Total number of external routes imported into the OSPF routing domain.
- Total number of OSPF routers in the routing domain.

Enter these values at the following prompts (sample values have been provided):

```
OSPF Config> enable ospf
Estimated # external routes[0]? 200
Estimated # OSPF routers [50]? 60
```

Next, configure each IP interface that is to participate in OSPF routing. To configure an IP interface for OSPF, use the following command:

```
OSPF Config> set interface
```

You are prompted to enter a series of operating parameters. Each interface is assigned a cost as well as other OSPF operating parameters.

When running other IP routing protocols besides OSPF, you may want to enable the exchange of routes between OSPF and the other protocols. To do this, use the following command:

```
OSPF Config> enable AS-boundary-routing
```

For more information on the OSPF configuration process, see Chapter 23, “Using and Configuring OSPF” on page 23-1.

Enabling the RIP Protocol

This section describes how to initially configure the RIP protocol. When configuring the RIP protocol, you can specify which set of routes the router will advertise and/or accept on each IP interface.

With an ATM network, RIP will work properly only if LAN Emulation is configured. For 1577 clients, use OSPF instead of RIP for an IGP.

First, enable the RIP protocol with the following command:

```
IP Config> enable RIP
```

When RIP is enabled, the following default behavior is established:

- The router includes all network and subnet routes in RIP updates sent out on each of its configured IP interfaces. It does not include default and static routes.
- The router processes all RIP updates received on each of its configured IP interfaces.
- RIP will not override default and static routes.

To change any of the default sending/receiving behaviors, use the following IP configuration commands, which are defined on a per-IP-interface basis.

```
IP Config> enable/disable sending net-routes
IP Config> enable/disable sending subnet-routes
IP Config> enable/disable sending static-routes
IP Config> enable/disable sending host-routes
IP Config> enable/disable sending default-routes
IP Config> enable/disable receiving rip
IP Config> enable/disable receiving dynamic nets
IP Config> enable/disable receiving dynamic subnets
IP Config> enable/disable receiving host-routes
IP Config> enable/disable override default
IP Config> enable/disable override static-routes
```

Enabling the BGP Protocol

The BGP protocol is enabled from its own configuration prompt, `BGP Config>` For more information about configuring BGP, refer to the discussion on using and configuring BGP4 in *Protocol Configuration and Monitoring Reference Volume 2*.

Adding Static Routing Information

This procedure is necessary only if you cannot gain routing information from any of the above dynamic routing protocols. Static routing persists over power failures and is used for routes that never change or cannot be learned dynamically.

Static routing information consists of any of the following items:

- **Default Gateway.** Packets are routed to default (authoritative) gateways when the packet destination cannot be found in the routing table.
- **Default Subnet Gateways.** If you are using subnetted networks, you can define a separate default gateway for each subnetted network.
- **Static Network/Subnet/Host Routes.** For each destination that is to have a fixed route, configure the next hop and distance to the destination.

Default Gateway

Routers send packets having unknown destinations (that is, destinations not present in the routing table) toward the default gateway. A default gateway is configured in the router by specifying the next hop to use to get to the default gateway and the cost of sending packets to the default gateway.

In the following example, the next hop toward the default gateway is 192.9.1.4 and the cost of sending a packet to the default gateway is 5.

```
IP Config> set default network-gateway
Default gateway [0.0.0.0]? 192.9.1.4
gateway’s cost [0]? 5
```

Default gateways can be learned and advertised by both the OSPF and RIP protocol. For the OSPF protocol, a router can be configured to advertise itself as the default gateway with the following OSPF command:

```
OSPF Config> enable/disable AS-boundary-routing
```

The RIP protocol can be configured so that it will advertise knowledge of the default gateway (if it has any) to its neighbors. RIP can also be configured so that a learned default gateway will (or will not) override a statically configured default gateway. These configuration tasks are accomplished with the following two commands:

```
IP Config> enable/disable sending default-routes
IP Config> enable/disable override default
```

Finally, a router that runs BGP can be configured to advertise itself (via the OSPF and RIP protocol) as the default gateway whenever it has BGP-learned routes in its routing tables. For OSPF, this is accomplished through the OSPF **enable/disable AS-boundary-routing** command. For RIP, the following commands are used:

```
IP Config> set originate-RIP-default
```

Default Subnet Gateways

There can be a default subnet gateway configured for each subnetted network that the router knows about. When the router attempts to forward a packet to a destination belonging to the subnetted network, but that destination cannot be found in the routing table, the packet is forwarded instead to the default subnet gateway.

Configuring default subnet gateways is the same as configuring the above default network gateway. The only difference is that you must specify the subnetted network on the command line. For example, to create a default subnet gateway for the subnetted network 18.0.0.0, you could use the following command:

```
IP Config> set default subnet-gateway
For which subnetted network [0.0.0.0]? 18.0.0.0
Default gateway [0.0.0.0]? 128.185.123.22
gateway's cost [0]? 2
```

This example specifies that the next hop to the subnet default gateway is 128.185.123.22, and that the cost of routing a packet to the default subnet gateway is 2.

Static Network/Subnet/Host Routes

Configure static routes for those destinations that cannot be discovered by the dynamic routing protocols, or to establish permanent or temporary main or backup routes. The destination is described by an IP network/subnet/host number (**dest-addr**) and the destination's address mask (**mask**). For host routes, the mask is always 255.255.255.255. The route to the destination is described by the IP address of the first hop router to use (**1st-hop**) and the cost of routing a packet to the destination (**cost**). To create, modify, or delete a static route, use the commands:

```
IP Config> add route dest-addr mask 1st-hop cost
IP Config> change route dest-addr mask new-mask 1st-hop cost
IP Config> delete route dest-addr mask
```

These commands take effect immediately, without the need to reboot the router.

Routes dynamically learned through the OSPF and RIP protocols can override static routes. For the RIP protocol, you can disable this override behavior. See the RIP section of this chapter concerning the **enable/disable override static-routes** commands.

You can configure both OSPF and RIP to advertise configured static routes over interfaces where these dynamic protocols are enabled.

To configure RIP to advertise static routes, enter the following command at the IP Config> prompt:

```
IP Config> enable sending static-routes ip-interface-address
```

To configure OSPF to advertise static routes, enter the following command at the OSPF Config> prompt:

```
OSPF Config> enable as boundary
Import static routes? yes
```

Setting Up ARP Configuration

The Address Resolution Protocol (ARP) is used to map protocol addresses to hardware addresses before a packet is forwarded by the router. ARP is always active on the router, so you do not need to do any additional configuration to enable it with its default characteristics. However, if you need to alter any ARP configuration parameters (such as **enable auto-refresh** or **set refresh-timer** which changes the default refresh timer), or if you need to add, change, or delete permanent address mappings, see Chapter 20, “Using and Configuring ARP.”

If LAN Emulation is configured on an interface, the defaults apply. You can effectively use the ARP protocol without any changes. If RFC 1577 (Classical IP and ARP over ATM) is used, additional configuration for ARP Clients and ARP Servers is required for each IP address configured on that ATM interface (as described in “ARP Over ATM Configuration Commands” on page 20-16).

Enabling ARP Subnet Routing

If there are hosts on attached subnetted networks that do not support IP subnetting, use Address Resolution Protocol (ARP) subnet routing (described in RFC 1027). When the router is configured for ARP subnet routing, it will reply by proxy to ARP requests for destination (that is, off the LAN if the router is itself the best route to the destination, and the destination is in the same natural network as the source). For proper operation, all routers attached to a LAN containing subnetting-ignorant hosts should be configured for ARP subnet routing.

To enable ARP subnet routing, use the following command:

```
IP Config> enable ARP-subnet-routing
```

Enabling ARP Network Routing

Some IP hosts ARP for all destinations, whether or not the destination is in the same natural network as the source. For these hosts, ARP subnet routing is not enough, and the router can be configured to reply by proxy to any ARP request as long as the destination is reachable through the router and the destination is not on the same local network segment as the source.

To enable ARP network routing, use the following command:

```
IP Config> enable arp-network-routing
```

IP Filtering

Filtering is a process by which the user specifies certain criteria that the router uses to control packet forwarding. The following two main types of filtering are provided to help users achieve their security and administrative goals:

- Access control
- Route filtering

Access Control

Access control allows the IP router to control the processing of individual packets based on source and destination IP addresses, IP protocol number, and by destination port number for the TCP and UDP protocols. This can control access to particular sets of IP hosts and services.

You can define access controls by configuring access control lists. One global list and two lists per interface can be specified. The global list applies to the router as a whole. Interface lists, also known as packet-filters, are assigned names and only apply to the designated interface. For each interface, one list applies to incoming packets, and the other applies to outgoing packets. The lists are applied independently of each other. A packet might *pass* an incoming interface list, and be *dropped* by the global list.

Figure 16-1 illustrates the series of searches a packet must pass before being forwarded (routed):

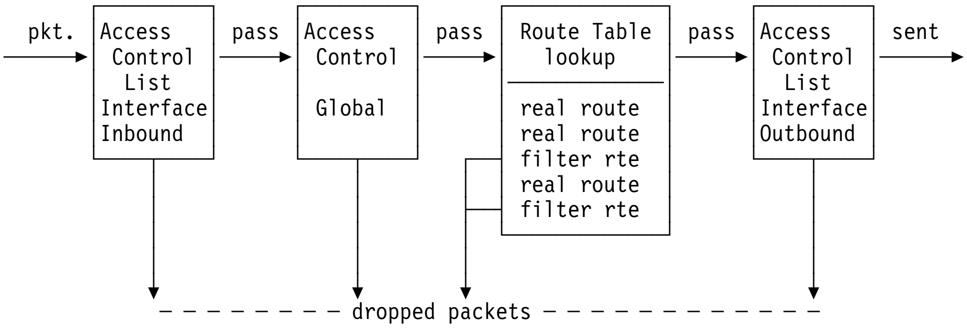


Figure 16-1. Access Control - Searching a Packet for Forwarding

Each access control list consists of one or more access control records that set the filtering criteria.

Access Control Records

Each record in a list may be inclusive or exclusive. Source and destination IP addresses and masks are required for every record and IP protocol number ranges and destination port ranges can also be specified. As IP packets flow through the router, IP headers are compared to access control list records. A packet matches a record if every specified field in the record matches a corresponding field in the packet's IP header. If a packet matches a record, and the record is inclusive, the packet *passes*. If the record is exclusive, the packet is *dropped* and is not processed any further by the router. If no records match after going through the entire list, the packet is also dropped. When defining records in access control lists, it is important to remember the following:

The order of records in a list is important. Configuration commands are provided to change the order of records in a list.

- For every list that includes at least one access control record, an inclusion record must exist for any packets to pass the list. One method of allowing all packets that do not match any of the specified records in a list to pass is to include the following wildcard record as the last record in the list:

```
add access-control inclusive 0.0.0.0  0.0.0.0
```

Source and Destination IP Addresses

Each record has an IP address and mask pair for both the source and destination IP addresses. When an IP packet is compared against an access control record, the IP address in the packet is “and-ed” with the mask in the record, and the result compared with the address in the record. For example, a source address of 26.0.0.0 with a mask of 255.0.0.0 in an access control record will match any IP packet source address with 26 in the first byte. A destination address of 192.67.67.20 and a mask of 255.255.255.255 will only match IP packet destination host address 192.67.67.20. An address of 0.0.0.0 with mask 0.0.0.0 is a wildcard, and matches any IP address.

Protocol Number

Each record can also have an IP protocol number range. This range is compared to the protocol byte in the IP header; a protocol value within the specified range (inclusive) will match. If you specify a range of 0 to 255, any protocol will match. Commonly used protocol numbers are 1 (ICMP), 6 (TCP), 17 (UDP), and 89 (OSPF).

Port Number

TCP/UDP port number ranges can also be specified in an access control record. This range is compared to the port number field in the TCP or UDP header of the IP packet; a port number value within the specified range (inclusive) will match. This field is ignored for IP packets that are not TCP or UDP packets. If you specify a range of 0 to 65535, any port number will match. Commonly used port numbers are 21 (FTP), 23 (Telnet), 25 (SMTP), 513 (rlogin) and 520 (RIP). See RFC 1700 (Assigned Numbers) for a list of IP protocol and port numbers.

Examples

The following example allows any host to send packets to the SMTP TCP socket on 192.67.67.20.

```
add access-control  inclusive 0.0.0.0  192.67.67.20  255.255.255.255 6 6 25 25
```

The next example prevents any host on subnet 1 of Class B network 150.150.0.0 from sending packets to hosts on subnet 2 of Class B network 150.150.0.0 (assuming a 1-byte subnet mask).

```
add access-control exclusive 150.150.1.0 255.255.255.0 150.150.2.0 255.255.255.0 0 255 0 65535
```

This command allows the router to send and receive all RIP packets.

```
add access-control inclusive 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 17 17 520 520
```

Enabling Access Control

IP Access Control (including global and interface access control) is enabled with the **set access-control on** command, and disabled with the **set access-control off** command.

If IP access control is enabled, you must be careful with packets that the router originates and receives. Be sure not to filter out the RIP or OSPF packets being sent or received by the router. The easiest way to do this is to add a wildcard inclusive entry as the last in the access control list. Alternately, you can add specific entries for RIP and OSPF, perhaps with restrictive addresses and masks. Note that some OSPF packets are sent to the Class D multicast addresses 224.0.0.5 and 224.0.0.6, which is important if address checking is being done for routing protocols. See the **add** command for more information on access control.

Defining the Global Access Control List

The global access control list is defined when records are added at the `IP Config>` prompt:

```
IP Config>  add access-control ...
```

Global access control list records can be listed, moved, or deleted using the **list**, **move**, or **delete** commands. See these commands for further information.

Defining Interface Specific Access Control Lists (Packet Filters)

To define interface specific access control lists, use the **add packet-filter** command at the `IP Config>` prompt. The router prompts you for the filter name, direction (input or output), and the interface number to which it applies.

```
Packet-filter name [ ]? test
Filter incoming or outgoing traffic? [IN]? in
Which interface is this filter for [0]? 1
```

You can use the **list packet-filter** command to list all interface specific access control lists configured in the router.

Setting Up Access Control Records for Interface Specific Access Control Lists

You must define access control records for each defined list. Otherwise, defined lists will have no effect on incoming or outgoing traffic. Use the **update packet-filter** command at the `IP Config>` prompt to define access control records. The router first prompts you for the name of the list (packet-filter) you want to update. The `IP Config>` prompt then changes to `Packet-filter ‘name’ Config>` where ‘name’ is the list name you provide.

```
IP Config>  update packet-filter
Packet-filter name [ ]?  test
Packet-filter ‘test’ Config>
```

From this prompt, you can issue **add**, **list**, **move**, and **delete** commands. These commands are similar to those used to modify the global access control list.

Route Filtering

Route filtering impacts packet forwarding by influencing the content of the routing table. In general, route filtering is more efficient but less flexible than access control. Filtering based on source IP address, IP protocol, and TCP/UDP port number can only be done using access control, described above. Route filtering is *not* recommended when OSPF is used in your network; OSPF-learned internal routes will override filtered routes in the routing table.

The following methods are used in this router to influence the content of the routing table.

- Filter routes
- RIP input filters

Defining a Filter Route

You can designate an IP destination to be inserted in the routing table as a *filter route*. IP packets will not be forwarded to these destinations, and routing information concerning them will not be advertised.

To configure a filter route, enter the following command at the `IP Config>` prompt:

```
IP Config>  add filter  dest-IP-address  address-mask
```

Filter routes will be listed as an entry with the type *fltr* when the **dump** command is used to view the IP routing table.

Note: If a more specific route is available, packets will be forwarded. For example, if a filter route is defined for network 9.0.0.0 (mask 255.0.0.0), but a route is learned for a subnet of the network (for example 9.1.0.0, mask 255.255.0.0), then packets will be forwarded to subnet 9.1.0.0 but not to other subnets of that network.

Defining RIP Input Filters

When RIP is used as the dynamic routing protocol, you can configure certain interfaces to ignore routes in RIP updates.

The following command results in ignoring all RIP updates received on an interface:

```
IP Config>  disable receiving rip  ip-interface-address
```

The following commands result in ignoring certain types of routes received on an interface:

```
IP Config> disable receiving dynamic nets ip-interface-address
IP Config> disable receiving dynamic subnets ip-interface-address
IP Config> disable receiving dynamic host ip-interface-address
```

When the latter group of commands are used, you can allow specific routes to be accepted using the following command:

```
IP Config> add accept-rip-route ip-network/subnet/host
```

Configuring the BOOTP/DHCP Forwarding Process

BOOTP (documented in RFC 951 and RFC 1542) is a bootstrap protocol used by a diskless workstation to learn its IP address, the location of its boot file, and the boot server name. Dynamic Host Configuration Protocol (DHCP), documented in RFC 1541, is used to allocate reusable network addresses and host-specific configuration parameters from a server.

The following terms are useful when discussing the BOOTP/DHCP forwarding process:

- *Client* - the workstation requiring BOOTP/DHCP services.
- *Servers* - the boot host (with UNIX daemon bootpd, DOS version available from FTP software, or OS/2) or other BOOTP/DHCP server that is providing these services. This router does not provide server support.
- *BOOTP relay agent* or *BOOTP forwarder* - a device which forwards requests/replies exchanged by the Client and Server. This router supports the relay agent function.

The following steps outline an example of the BOOTP forwarding process. (DHCP exchanges proceed in a similar way):

1. The Client copies its Ethernet address (or appropriate MAC address) into a BOOTP packet and broadcasts it onto the local LAN. BOOTP is running on top of UDP.
2. The local BOOTP relay agent receives the packet and checks to see if the packet is well formatted and that the maximum number of application hops has not expired. It also checks to see if the client has been trying long enough.

   Note: If multiple hops are required before reaching the BOOTP agent, the packet is routed normally via IP. All other routers would not examine the packet to determine whether it is a BOOTP packet.
3. The Local BOOTP agent forwards a separate BOOTP request to each of its configured servers. The BOOTP request is the same as the one that was initially sent by the client except that it has a new IP header with the relay agent's IP address copied into the body of the BOOTP request.
4. The server receives the request and looks up the client's hardware (for example, Ethernet) address in its database. If found, it formats a BOOTP reply containing the client's IP address, the location of its boot file, and the boot server name. The reply is then sent to the BOOTP relay agent.
5. The BOOTP relay agent receives the reply and makes an entry in its ARP table for the client and then forwards the reply to the client.
6. The client then continues to boot using TFTP, using the information in the BOOTP reply packet.

Enabling/Disabling BOOTP Forwarding

To enable or disable BOOTP forwarding on the router, enter the following command at the IP configuration prompt. (Enable BOOTP Forwarding to allow the router to forward BOOTP and/or DHCP requests and replies between Clients and Servers on different segments of your network.)

```
IP Config> enable/disable bootp
```

When enabling BOOTP, you are prompted for the following values:

- Maximum number of application hops you want the BOOTP request to go. This is the maximum number of BOOTP relay agents that can forward the packet. This is NOT the maximum number of IP hops to the Server. A typical value for this parameter is 1.

- Number of seconds you want the Client to retry before the BOOTP request is forwarded. *This parameter is not commonly used.* A typical value for this parameter is 0.

After accepting a BOOTP request, the router forwards the BOOTP request to each BOOTP server. If there are multiple servers configured for BOOTP, the router replicates the packet.

Configuring a BOOTP/DHCP Server

To add a BOOTP or DHCP server to the router’s configuration, enter the following command at the IP configuration prompt:

```
IP Config> add BOOTP-SERVER [IP address of server]
```

Multiple servers can be configured. In addition, if only the network number of the server is known or if multiple servers reside on the same network segment, a broadcast address can be configured for the server.

Configuring UDP Forwarding

User datagram protocol (UDP), documented in RFC 768, is a transport layer protocol providing connectionless service using the Internet Protocol. With UDP Forwarding, locally delivered UDP packets (such as UDP Broadcast on an IBM 2216-attached LAN) can be forwarded to a specific IP destination or to a destination network as a directed broadcast.

For example, NetBIOS uses UDP broadcasts in some client-server applications to broadcast Name-Query packets. Unless you set up UDP Forwarding, the router drops those packets; thus, the router will not forward the broadcast packets beyond the local network.

Follow these steps to configure UDP Forwarding:

1. Add a UDP destination port number and IP address. The router maps this IP address to the UDP port.

   ```
   IP Config> add udp-destination
   UDP port number [-1] 36
   Destination IP address [0.0.0.0] 20.1.2.2
   ```

2. Enable UDP Forwarding.

   ```
   IP Config>enable udp-forwarding
   For which UDP port number [-1] 36
   ```

In the above example, the router forwards packets it receives for UDP port 36 to IP address 20.1.2.2.

Enter **list udp-forwarding** to see the UDP Forwarding configuration.

Enabling/Disabling UDP Forwarding

To enable or disable UDP Forwarding on the router, enter the following command at the IP configuration prompt. (Enable UDP Forwarding to allow the router to forward UDP Broadcast packets to a given address on a per-UDP port basis.)

```
IP Config> enable/disable udp-forwarding port-number
```

Adding a UDP Destination

Add UDP Forwarding destinations by specifying the IP address to which the packets are to be forwarded followed by the port number. To add a UDP destination, enter the following command at the IP configuration prompt:

```
IP Config> add udp-destination port-number  destination-ip-address
```

IP Multicast Support

IP multicast is an extension of LAN multicasting to a TCP/IP Internet. It is the ability of an IP host to send a single datagram (called IP multicast datagram) that will be delivered to multiple destinations. IP multicast datagrams are identified as those packets whose destinations are class D IP addresses (that is, whose first byte lies in the range 224-239 inclusive). Each class D address defines a multicast group.

The extensions required of an IP host to participate in IP multicasting are specified in RFC 1112 (Host Extensions for IP Multicasting.) That document defines a protocol, the Internet Group Management Protocol (IGMP), that enables hosts to dynamically join and leave multicast groups. This router implements the IGMP protocol functions that enable it to keep track of IP group membership on its local physical and on its emulated LANs by sending IGMP Host Membership Queries and receiving IGMP Host Membership Reports.

A router must also be able to route IP multicast datagrams between the source and (multiple) destination hosts. This router supports the Multicast Open Shortest Path First (MOSPF) protocol as defined by RFC 1584 (Multicast Extensions to OSPF), and the Distance Vector Multicast Routing Protocol (DVMRP).

A MOSPF router distributes group location information throughout the routing domain by flooding a new type of link state advertisement, the group-membership-LSA (type 6). This in turn enables the MOSPF routers to most efficiently forward a multicast datagram to its multiple destinations: each router calculates the path of the multicast datagram as a tree whose root is the datagram source, and whose terminal branches are LANs containing group members. For more information, see “Multicast OSPF” on page 23-3.

DVMRP is a multicast routing protocol derived from the Routing Informatin Protocol (RIP). This router provides support for DVMRP so that you can exchange multicast routing information with other routing entities that do not support MOSPF. This router's DVMRP implementation also allows tunneling of DVMRP information over an MOSPF-capable network and over a non-multicast-capable IP network.

This router also allows you to “enroll” the router itself as a member of one or more multicast groups. As a member of a multicast group, the router will respond to “pings” and SNMP queries addressed to the group address (one command could be used to query multiple routers).

Additionally, the router’s IP multicasting support is used to establish and manage DLSw groups, which reduces the amount of configuration needed for DLSw. For additional information, refer to Chapter 27, “Using and Configuring DLSw” on page 27-1.

Configuring the router for IP Multicast

To enable the router to track IP multicast group memberships and forward multicast datagrams, you must enable MOSPF, DVMRP, or both MOSPF and DVMRP.

To enable MOSPF, you must first enable OSPF (see "Enabling the OSPF Protocol" on page 16-3) and then do the following:

1. Enable multicast forwarding by entering the following command at the `OSPF Config>` prompt:

   ```
   OSPF Config> enable multicast-routing
   ```

2. Set the following parameters when configuring each OSPF interface through the `OSPF Config>` **set interface** command:

   ```
     Forward multicast datagrams (Yes or No)?
     Forward as datalink unicasts (Yes or No)?
     IGMP polling interval (in seconds) [60]?
     IGMP timeout (in seconds) [180]?
   ```

For more details on these configuration commands, see "Multicast OSPF" on page 23-3.

The following OSPF monitoring commands are used to obtain information about IP multicast groups and MOSPF routing:

```
OSPF> advertisement ls-type link-state-id
      (by entering type 6 in the ls-type field, this
       command displays the contents of the group-membership
       link-state-advertisement sent out by the router)

OSPF> interface interface-ip-address
       (a number of output fields are specific to multicasting
        and provide statistics on IGMP and multicast packets)

OSPF> Mcache
       (displays list of currently active multicast cache
        entries

OSPF> Mgroups
      (displays group membership of the router's attached
       interfaces as reported via IGMP)

OSPF> Mstat
      (displays various multicast packet statistics)
```

For more information on these commands, refer to "Multicast OSPF" on page 23-3.

Enabling DVMRP: To enable DVMRP:

1. Enable DVMRP on the router

   ```
   DVMRP config> dvmrp on
   ```

2. Establish which LAN interfaces DVMRP would run on

   ```
   DVMRP config> phyint interface-address metric threshold
   ```

When DVMRP is the only multicast routing protocol on an interface, IGMP polling interval and timeout are set and cannot be changed. These values are 125 and 270 seconds respectively.

Refer to "Configuring DVMRP" in *Protocol Configuration and Monitoring Reference Volume 2* for details on these commands and other configuration commands used to set the interaction between DVMRP and MOSPF when both are active on the router.

Enrolling the router in IP multicast groups

If the router itself is to join one or more multicast groups, the following join/leave commands are used:

- **join multicast-group-address**
- **leave multicast-group-address**

These **join** and **leave** commands are accessible from the `OSPF Config` prompt and the `OSPF` monitoring prompt. They are also available on the DVMRP monitoring console.

Note that these commands are not necessary for the router to perform its IP multicast forwarding or IGMP group tracking functions; they are used to add the router to groups so that it can respond to “pings” and SNMP queries addressed to these groups.

Accessing the IP Configuration Environment

To access the IP configuration environment, enter the following command at the `Config>` prompt:

```
Config> Protocol IP
Internet protocol user configuration
IP Config>
```

IP Configuration Commands

This section summarizes and then explains all IP configuration commands. These commands allow you to modify the IP protocol behavior to meet your specific requirements. Some amount of configuration is necessary to produce a fully functional IP router. Enter IP configuration commands at the `IP config>` prompt.

Table 16-1. IP Configuration Commands Summary

Command	Function
? (Help)	Lists the configuration commands or lists the actions associated with specific commands.
Add	Adds to the IP configuration information. Interface addresses can be added, along with access controls, filters, and packet-filters.
Change	Modifies information that was originally entered with the **add** command.
Delete	Deletes IP configuration information that had been entered with the **add** command.
Disable	Disables certain IP features that have been turned on by the **enable** command.
Enable	Enables IP features such as ARP subnet routing, UDP Forwarding, originate default, directed broadcasts, BOOTP, and the various RIP flags controlling the sending and receiving of RIP information.
List	Displays IP configuration items.
Move	Changes the order of access control records.
Set	Establishes IP configuration modes such as the use of access control and the format of broadcast addresses. Also sets IP parameters such as default routers, TTL (time-to-live) of packets originated by the router, and the size of the IP routing table.
Update	Used to assign access control entries.
Exit	Exits the IP configuration process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **`?`**

```
ADD
CHANGE
DELETE
DISABLE
ENABLE
LIST
MOVE
SET
UPDATE
EXIT
```

Example: **`add ?`**

```
accept-rip-route
access-control
address
bootp-server
filter
packet-filter
REDUNDANT Default Gateway
route
udp-destination
```

Add

Use the **add** command to add IP information to your configuration.

Syntax: add accept-rip-route . . .
access-control . . .
address . . .
bootp-server
filter . . .
packet-filter
route . . .
udp-destination . . .

accept-rip-route *IP-network/subnet*

Allows an interface to accept a RIP route when input RIP filtering is enabled for an interface. You can print the list of networks/subnets that have already been entered using the **list rip-routes-accept** command. You can enable the input filtering of RIP routes on a per-IP-interface basis. This is done separately for network-level routes (for example, a route to 10.0.0.0) for subnet-level routes (for example, a route to 128.185.0.0), and for host-level routes (for example 128.185.123.28). To enable input filtering of routes on an IP interface, use the **disable dynamic nets/subnets/host** commands.

IP network/subnet

Valid Values: any valid IP address

Default Value: none

Example: **`add accept-rip-route 10.0.0.1`**

or

Example: **`add accept-rip-route`**

```
Network number [0.0.0.0]? 10.0.0.0
```

access-control *type IP-source source-mask IP-dest dest-mask*

[first-protocol last-protocol] [first-port last-port]

Adds an access control record to the end of the global access control list. This allows you to describe a class of packets to forward or drop, depending on the type of the record. The length and order of the IP access control list can affect the performance of the IP forwarder. Each record must be assigned the following: type, IP source, source-mask, IP destination, and destination-mask fields. The type must either be inclusive or exclusive. The *IP-source* and *IP-dest* fields are in the form of IP addresses in dotted decimal notation. Optionally, you may specify an IP protocol number range with the *first-protocol* and *last-protocol* fields, which are an inclusive range of IP protocols that match this entry. You also may specify a TCP or UDP port number or port number range that matches an entry, where "port number range" is an inclusive range of TCP and UDP ports that matches this entry. Specify TCP or UDP in the protocol fields, then specify the port number range in the first-port and last-port fields.

type Indicates whether packets are sent or dropped for a specific address or set of addresses.

Specify *Include* to cause the router to receive a packet and to forward it if it matches criteria in the remaining arguments.

Specify *Exclude* to cause the router to discard the packets.

IP-source

Valid Values: any valid IP address

Default Value: none

source-mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: none

IP-dest

Valid Values: any valid IP address

Default Value: none

dest-mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: none

first-protocol

The lower boundary of a range of IP protocol numbers.

Some commonly used protocol numbers are:

- 1 for ICMP
- 6 for TCP
- 17 for UDP
- 89 for OSPF

Valid Values: 0 to 255

Default Value: 0

last-protocol

The upper boundary of a range of IP protocol numbers.

Some commonly used protocol numbers are:

- 1 for ICMP
- 6 for TCP
- 17 for UDP
- 89 for OSPF

Valid Values: 0 to 255

Default Value: 255

first-port

The lower boundary of an IP TCP/UDP port number range.

Some commonly used port numbers are:

- 21 for FTP
- 23 for Telnet
- 25 for SMTP
- 513 for rlogin
- 520 for RIP

Valid Values: a port number in the range of 0 - 65535

Default Value: 0

last-port

The upper boundary of an IP TCP/UDP port number range.

Some commonly used port numbers are:

- 21 for FTP
- 23 for Telnet
- 25 for SMTP
- 513 for rlogin
- 520 for RIP

Valid Values: a port number in the range of 0 - 65535

Default Value: 65535

Example: **`add access-control inclusive`**

```
Internet source [0.0.0.0]?
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]?
Destination mask [255.255.255.255]?
Enter starting protocol number ([CR] for all) [-1]?
IP config>
```

address *interface-number* *IP-address* *address-mask*

Assigns an IP address to one of the router's hardware network interfaces. A hardware network interface will not receive or transmit IP packets until it has at least one IP address. You must specify an IP address together with its subnet mask. For example, if the address is on a class B network, using the third byte for subnetting, the mask would be 255.255.255.0. Use the **list devices** command to obtain the appropriate command interface-number. Serial lines do not need addresses. Such lines are called unnumbered. However, you must still enable them for IP traffic using the **add address** command. The address then used is 0.0.0.n, where n is the *interface-number*.

Valid Values: For non-serial line interfaces:

- The class A range is 1.0.0.1 through 126.255.255.254
- The class B range is 128.0.0.1 through 191.255.255.254
- The class C range is 192.0.0.1 through 223.255.255.254

For serial line interfaces:

- 0.0.0.n, where n is the hardware interface number.

You must specify an IP address together with its subnet mask. For example, if the address is on a class B network, using the third byte for subnetting, the mask would be 255.255.255.0. Use the **List Devices** option to obtain the appropriate option interface-number.

interface-number

Valid Values: any defined interface number

Default Value: none

IP-address

Valid Values: any valid IP address

Default Value: none

address mask

Valid Values: 0.0.0.0 - 255.255.255.255

Default Value: none

Example: `add address 0 128.185.123.22 255.255.255.0`

bootp-server *server-IP-address*

Adds a BOOTP/DHCP server to a network configuration. Acting as a bootp relay agent, your router accepts and forwards BOOTP/DHCP requests to the BOOTP/DHCP server. BOOTP is a bootstrap protocol used by a router or a diskless workstation to learn its IP address, the location of its boot file, and the boot server name. DHCP is Dynamic Host Configuration Protocol, used to configure a host over a network connection.

server-IP-address

Valid Values: any valid Bootp server IP address

Default Value: none

Example: `add bootp-server 128.185.123.22`

filter *dest-IP-address address-mask*

Designates an IP destination to be filtered. IP packets will not be forwarded to filtered destinations, nor will routing information be disseminated concerning such destinations. Packets to filtered destinations are simply discarded. You must specify a filtered destination as an IP address with its subnet mask. For example, to filter a subnet of a class B network, using the third byte for subnetting, the mask would be 255.255.255.0. Using the filter mechanism is more efficient than IP access controls, although not as flexible. Filters also affect the operation of the IP routing protocols, unlike access controls. Filtered networks/subnets are overridden if learned using the OSPF routing protocol.

The effect of this command is immediate; you do not have to reboot the router for it to take effect.

dest-IP-address

Valid Values: any valid IP address

Default Value: none

address mask.

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: 0.0.0.0

Example: **`add filter 127.0.0.0 255.0.0.0`**

packet-filter *filter-name type intf#*

Defines a packet filter record within the router configuration.

filter-name

Valid Values: any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: none

type *IN* filters incoming traffic.

OUT filters outgoing traffic.

intf# **Valid Values**: any defined interface

Default Value: none

Example: **`add packet-filter`**

```
Packet-filter name [ ]? filt-1-0
Filter incoming or outgoing traffic? [ IN]?
Which interface is this filter for [0]? 1
```

route *IP-network/subnet/host IP-mask next-hop cost*

Adds a static network/subnet/host route to the router's IP configuration. When dynamic routing information is not available for a particular destination, static routes are used.

The destination is specified by an IP address (*IP-network/subnet/host*) together with an address mask (*IP-mask*). If the destination IP address is a network address, then the IP mask must be a network mask. If the destination IP address is a subnet address, then the IP mask must be a subnet mask. Finally, if the destination IP address is a host address, then the IP mask must be a host mask (which means that the only valid value is 255.255.255.255.) The IP-mask must be accurate; if it is not, the static route will not be accepted.

The route to the destination is specified by the IP address of the next hop (*next-hop*), and the cost (*cost*) of routing the packet to the destination. The next hop must be on the same (sub)net as one of the router's directly connected interfaces. Static routes are always overridden by routes learned through OSPF. By default, static routes are also overridden by routes learned through RIP; however, you can change that with the **enable/disable override static-routes** command.

The effect of this command is immediate; you do not have to reboot the router for it to take effect.

IP-network/subnet/host

Valid Values: any valid IP address

Default Value: none

IP-mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: none

next-hop

Valid Values: any valid IP address

Default Value: none

cost **Valid Values**: an integer in the range of 0 - 255

Default Value: 1

Examples: `add route  17.0.0.0  255.0.0.0  128.185.123.22  6`

Class A (network example):

```
add route 9.0.0.0  255.0.0.0  17.102.23.1  4
```

Class A (subnet example):

```
add route 9.67.0.0  255.255.0.0  17.102.23.1  4
```

Class B (subnet example):

```
add route 192.3.2.32  255.255.255.224  17.102.23.1  4
```

Class B (host example):

```
add route 167.59.34.67  255.255.255.255  17.102.23.1  4
```

udp-destination *port-number address*

Adds a UDP destination port number and IP address.

You can enter a broadcast or unicast IP address.

Repeat this command to add more than one IP address for the same UDP port. This causes the router to forward the packet to each of the IP addresses.

port-number

Valid Values: 0 to 65535

Default Value: none

address

Valid Values: any valid IP address

Default Value: none

Example:

```
add udp-destination 36 20.1.2.2
```

Change

Use the **change** command to change an IP configuration item previously installed by the **add** command. In general, you must specify the item you want to change, just as you specified the item with the **add** command.

Syntax: change access-control . . .
address . . .
route . . .

access-control *record-number type IP-source source-mask IP-dest dest-mask*

[first-protocol last-protocol] [first-port last-port]

Modifies an existing global access-control record. Use the **list access-control** command to view all existing records and obtain the record number.

Example: **`change access-control 2`**

```
Enter type [E]? i
Internet source [1.1.1.1]?
Source mask [255.255.255.255]?
Internet destination [2.2.2.2]?
Destination mask [255.255.255.255]?
Enter starting protocol number [6]?
Enter ending protocol number [6]?
Enter starting port number [23]?
Enter ending port number [23]?
```

address *old-address new-address new-mask*

Modifies one of the router's IP interface addresses. You must specify each new address together with the new address' subnet mask. This command can also be used to change an existing address' subnet mask.

For non-serial line interfaces:

- The class A range is 1.0.0.1 through 126.255.255.254
- The class B range is 128.0.0.1 through 191.255.255.254
- The class C range is 192.0.0.1 through 223.255.255.254

For serial line interfaces:

- 0.0.0.n, where n is the hardware interface number.

Use these address guidelines to:

1. Enter the *old-address*.
2. Enter the *new-address*.
3. Enter the new *new-mask*.

Example: **`change address 192.9.1.1 128.185.123.22 255.255.255.0`**

route *destination mask new-1st-hop new-cost*

Modifies either the next hop or the cost associated with a configured static network/subnet route. The effect of this command is immediate; you do not have to reboot the router for it to take effect.

destination

Valid Values: any valid IP address

Default Value: none

mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: none

new-1st-hop

Valid Values: any valid IP address

Default Value: none

new-cost

Valid Values: an integer in the range of 0 - 255

Default Value: 1

Example: `change route 10.0.0.0  255.0.0.0  128.185.123.18 6`

Delete

Use the **delete** command to delete an IP configuration item previously installed by the **add** command. In general, you must specify the item you want to delete, just as you specified the item with the **add** command.

Syntax: delete accept-rip-route . . .
access-control . . .
address . . .
bootp-server
default network/subnet-gateway . . .
filter . . .
packet-filter
route . . .
udp-destination . . .

accept-rip-route *net-number*

Removes a route from the list of networks that the RIP protocol always accepts.

Valid Values: Any IP address contained in the list of accepted networks.

Default Value: none

Example: `delete accept-rip-route 10.0.0.0`

access-control *record-number*

Deletes one of the access control records from the global access control list.

Example: `delete access-control  2`

address *ip-interface-address*

Deletes one of the router's IP interface addresses.

Valid Values: any valid IP address

Default Value: none

Example: `delete address 128.185.123.22`

bootp-server *server-IP-address*

Removes a BOOTP server from an IP configuration.

Valid Values: any valid Bootp-Server IP address

Default Value: 0.0.0.0

Example: `delete bootp-server 128.185.123.22`

default *network/subnet-gateway [subnetted network]*

Deletes either the default gateway or the default subnet gateway for the specified subnetted network.

Valid Values: any valid IP address

Default Value: 0.0.0.0

Example: **delete default subnet-gateway 128.185.0.0**

filter *destination address destination mask*

Deletes one of the router's filtered networks. The effect of this command is immediate; you do not have to reboot the router for it to take effect.

destination address
Valid Values: any valid IP address

Default Value: 0.0.0.0

destination mask
Valid Values: 0.0.0.0 - 255.255.255.255

Default Value: none

Example: **delete filter 127.0.0.0**

```
          Address mask [0.0.0.0]? 255.0.0.0
```

packet-filter *filter-name*

Deletes a specified packet-filter from the router's configuration.

Valid Values: any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: none

Example: **delete packet-filter pf-in-0**

```
IP config> delete packet-filter pf-in-0
All access controls defined for 'pf-in-0' will also be deleted.
Are you sure you want to delete(Yes or [No]): y
Deleted
IP config>
```

route *destination address destination mask*

Deletes one of the router's configured static routes. The effect of this command is immediate; you do not have to reboot the router for it to take effect.

destination address
Valid Values: any valid IP address

Default Value: none

destination mask
Valid Values: any valid IP mask

Default Value: none

Example: **delete route 10.0.0.0**

```
          Address mask [0.0.0.0]? 255.0.0.0
```

udp-destination *port-number address*

Deletes a UDP Forwarding destination address that was configured using the **add udp-destination** command.

The result is that locally delivered UDP datagrams received at the specified port will not be forwarded to the specified IP address.

port-number

Valid Values: any integer in the range of 0 - 65535

Default Value: none

address

Valid Values: any valid IP address

Default Value: none

Examples:

```
delete udp-destination 36 20.1.2.2
```

Disable

Use the **disable** command to disable IP features previously enabled by the **add** command.

Syntax: disable arp-net-routing
arp-subnet-routing
bootp-forwarding
directed-broadcast
echo-reply
override default/static-routes . . .
packet-filter
per-packet-multipath
receiving rip . . .
receiving dynamic nets/subnets/host . . .
rip
sending default/net/subnet/poisoned/host/static . . .
source-routing
udp-forwarding . . .

arp-net-routing

Turns off ARP network routing. When this is enabled, the router replies by proxy to all ARP requests for remote destinations that are best reached through the router. This is the default and the generally recommended setting.

Example: `disable arp-net-routing`

arp-subnet-routing

Turns off the IP feature called ARP subnet routing or proxy ARP, which, when enabled, deals with hosts that have no IP subnetting support. This is the default and the generally recommended setting.

Example: `disable arp-subnet-routing`

bootp-forwarding

Turns off the BOOTP/DHCP relay function.

Example: `disable bootp-forwarding`

directed-broadcast

Disables the forwarding of IP packets whose destination is a nonlocal (for example, remote LAN) broadcast address. The source host originates the packet as a unicast where it is then forwarded as a unicast to a destination subnet and “exploded” into a broadcast. You can use these packets to locate network servers.

Note: Forwarding and exploding cannot be disabled separately.

Example: `disable directed-broadcast`

echo-reply

Disables the router's ICMP Echo Reply function. Thus a ping sent to any of the router's interfaces will not generate a reply. The router defaults to echo-reply enabled.

Example: `disable echo-reply`

override default/static-routes *ip-interface-address*

Prevents an RIP default route received on interface *ip-interface-address* from being installed as the router's default route. The **disable override static-routes** command prevents RIP routes received on interface *ip-interface-address* from overriding any of the router's static routes.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `disable override default 128.185.123.22`

packet-filter *filter-name*

Disables specified interface-specific access control list (packet-filters).

filter-name

Valid Values: Any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: None

Example: `disable packet-filter pf-in-0`

per-packet-multipath

If per-packet-multipath is disabled, the router will choose the first available path to a destination. The default for this feature is disabled.

Example: `disable per-packet-multipath`

receiving rip *ip-interface-address*

Prevents any RIP packets from being received on interface *ip-interface-address.*

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `disable receiving rip 128.185.123.22`

receiving dynamic nets/subnets/host *ip-interface-address*

The **disable receiving dynamic nets** command ensures that for RIP updates received on the interface *ip-interface-address*, the router accept only those network level routes entered by the **add accept-rip-route** command. The **disable receiving dynamic subnets** command produces the analogous behavior for subnet routes. The **disable receiving dynamic host** produces the analogous behavior for host routes.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `disable receiving dynamic nets 128.185.123.22` sending default/net/subnet/static *ip-interface-address ip-interface-address*

Prevents the router from advertising a default route in RIP updates sent out the interface ip-interface-address. The other flags that control the RIP routes sent out an interface are **host-routes. static-routes. net-routes**, and **subnet-routes**. You can turn these off individually. A route is advertised if it is specified by any of the enabled flags.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `disable sending net-routes 128.185.123.22`

rip

Turns off the RIP protocol.

Example: `disable rip`

source-routing

Prevents the router from forwarding source-routed packets (that is, IP datagrams that include a source-route option, thus avoiding normal IP routing tables). This option defaults to source-routing enabled.

Example: `disable source-routing`

udp-forwarding *port-number*

Disables UDP forwarding for packets received by the router with the specified UDP destination port number.

Default: UDP forwarding is disabled for all port numbers.

port-number

Valid Values: an integer in the range of 0 - 65535

Default Value: 0

Example: `disable udp-forwarding 36`

Enable

Use the **enable** command to activate IP features, capabilities, and information added to your IP configuration.

Syntax: enable arp-net-routing
arp-subnet-routing
bootp-forwarding
directed-broadcast
echo-reply
override default ...
override static-routes ...
packet-filter
per-packet-multipath
receiving rip ...
receiving dynamic nets ...
receiving dynamic subnets ...
rip
sending default-routes ...
sending net-routes ...
sending poisoned-reverse-routes
sending subnet-routes ...

sending static-routes ...
sending host-routes
source-routing
udp-forwarding ...

arp-net-routing

Turns on ARP network routing. When enabled, the router replies by proxy to all ARP requests for remote destinations that are best reached through the router. Use this command when there are hosts on the LAN that ARP for all destinations, instead of (as is proper) only local destinations.

Example: **`enable arp-net-routing`**

arp-subnet-routing

Turns on the router's ARP subnet routing (sometimes also called Proxy ARP) function. This function is used when there are subnet-incapable hosts attached to directly-connected IP subnets. The directly connected subnet having subnet-incapable hosts must use ARP for this feature to be useful.

The way ARP subnet routing works is as follows. When a subnet-incapable host wants to send an IP packet to a destination on a remote subnet, it does not realize that it should send the packet to a router. The subnet-incapable host therefore simply broadcasts an ARP request. This ARP request is received by the router. The router responds as the destination (hence the name proxy) if both arp-subnet-routing is enabled and if the next hop to the destination is over a different interface than the interface receiving the ARP request.

If there are no hosts on your LAN that are “subnet-incapable,” do not enable ARP-subnet routing. If ARP subnet routing is needed on a LAN, it should be enabled on all routers on that LAN.

Example: **`enable arp-subnet-routing`**

bootp-forwarding

Turns on BOOTP/DHCP packet forwarding. In order to use BOOTP forwarding, you must also add one or more BOOTP servers with the **add bootp-server** command.

Example: **`enable bootp-forwarding`**

```
Maximum number of forwarding hops [4]?
Minimum seconds before forwarding [0]?
```

Maximum number of forwarding hops
Maximum number of allowable BOOTP agents that can forward a BOOTP request from the client to the Server (this is not the maximum number of IP hops to the server).

Default: 4

Minimum seconds before forwarding
This parameter is generally not used. Use this parameter when there is a redundant path between the client and the server, and you want to use the secondary path(s) as a standby.

Default Value: 0

directed-broadcast

Enables the forwarding of IP packets whose destination is a network-directed or subnet-directed broadcast address. The packet is originated by the source host as a unicast where it is then forwarded as a unicast to a destination subnet and "exploded" into a broadcast. These packets can be used to locate network servers. This command enables both the forwarding and exploding of directed broadcasts. The IP packet forwarder never forwards link level broadcasts/multicasts, unless they correspond to Class D IP addresses. (See the OSPF **enable multicast-routing** command.) The default setting for this feature is enabled.

Note: Forwarding and exploding cannot be implemented separately. Also, the router will not forward all-subnets IP broadcasts.

Example: `enable directed-broadcast`

echo-reply

Enables the building and sending of an ICMP Echo Reply in response to an ICMP Echo Request.

Example: `enable echo-reply`

override default *ip-interface-address*

Enables received RIP information to override the router's default gateway. This command is invoked on a per-IP-interface basis. When the **enable override default** command is invoked, default RIP routes received on interface ip-interface-address overwrite the router's current default gateway, providing the cost of the new default is cheaper.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable override default 128.185.123.22`

override static-routes *ip-interface-address*

Enables received RIP information to override some of the router's statically configured routing information. This command is invoked on a per-IP-interface basis. When the **enable override static-routes** command is invoked, RIP routing information received on interface ip-interface-address overwrite statically configured network/subnet routes providing the cost of the RIP information is cheaper.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable override static-routes 128.185.123.22`

packet-filter *filter-name*

Enables specified interface-specific access control list (packet-filters).

filter-name

Valid Values: any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: none

Example: `enable packet-filter pf-in-0`

per-packet-multipath

If per-packet-multipath is enabled, and there are multiple equal-cost paths to a destination, then the router chooses the path for forwarding each packet in a round-robin fashion. The default for this feature is disabled.

Example: `enable per-packet-multipath`

receiving rip *ip-interface-address*

Enables the processing of RIP updates that are received on a particular interface. This command has an analogous disable command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving rip** command, no RIP updates will be accepted on interface *ip-interface-address* address.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable receiving rip 128.185.123.22`

receiving dynamic nets *ip-interface-address*

Modifies the processing of RIP updates that are received on a particular interface. This command has an analogous disable command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving dynamic nets** command, for RIP updates received on interface *ip-interface-address*, the router will not accept any network-level routes unless they have been specified in an **add accept-rip-route** command.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable receiving dynamic nets 128.185.123.22`

receiving dynamic subnets *ip-interface-address*

Modifies the processing of RIP updates that are received on a particular interface. This command has an analogous disable command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving dynamic subnets** command, for RIP updates received on interface *ip-interface-address*, the router will not accept any subnet-level routes unless they have been specified in an **add accept-rip-route** command.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable receiving dynamic subnets 128.185.123.22`

receiving dynamic host *ip-interface-address*

Modifies the processing of RIP updates that are received on a particular interface. This command has an analogous disable command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving dynamic host** command, for RIP updates received on interface *ip-interface-address*, the router will not accept any host routes unless they have been specified in an **add accept-rip-route** command.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable receiving dynamic host 128.185.   123.22`

rip

Enables the router's RIP protocol processing.

When RIP is enabled, the following default behavior is established:

- The router includes all network and subnet routes in RIP updates sent out on each of its configured IP interfaces.
- The router processes all RIP updates received on each of its configured IP interfaces.

To change any of the default sending/receiving behaviors, use the IP configuration commands which are defined on a per-IP-interface basis.

Example: `enable rip`

sending default-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous disable command. (See the **disable sending** command.) The effect of the **enable sending** command is additive. Each separate enable sending command specifies that a certain set of routes should be advertised from a particular interface. A route is included in a RIP update only if it has been included by at least one of the enable sending commands. The **enable sending default-routes** command specifies that the default route (if one exists) should be included in RIP updates sent out interface ip-interface-address.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable sending default-routes 128.185.123.22`

Note: Some settings of the **enable sending ...** commands are redundant. For example, if you invoke **enable sending net-routes**, **enable sending subnet-routes**, and **enable sending host-routes** for a particular interface, there is no need to also specify **enable sending static-routes** (because each static route is a network-level, subnet, or host route). By default, when you first enable RIP, sending net-routes, sending subnet-routes, and sending host-routes are enabled for each interface, while sending static-routes and sending default are disabled.

sending net-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous disable command. (See the **disable sending** command.)

The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes should be advertised from a particular interface. A route is included in an RIP update only if it has been included by at least one of the **enable sending** commands. The **enable sending network-routes** command specifies that all network-level routes should be included in RIP updates sent out interface ip-interface-address. A network-level route is a route to a single class A, B, or C IP network.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable sending net-routes 128.185.123.22`

sending poisoned-reverse-routes *ip-interface-address*

A technique used by RIP to improve convergence time when routes change (for complete details on the technique, refer to rfc 1058). Use of this technique increases the size of RIP update messages. You may find it more acceptable to minimize routing overhead by accepting somewhat slower convergence. The **disable sending poisoned-reverse-routes** command specifies that poisoned reverse routes should not be included in RIP updates sent out on an interface specified by the **enable ip-interface-address** command.

Default: Enabled

ip-interface-address

Valid Values: any valid IP address

Default Value: none

sending subnet-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous disable command. (See the **disable sending** command.) The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes should be advertised out a particular interface. A route is included in an RIP update only if it has been included by at least one of the enable sending commands. The **enable sending subnet-routes** command specifies that all subnet routes should be included in RIP updates sent out interface ip-interface-address. However, a subnet route is included only if ip-interface-address connects directly to a subnet of the same IP subnetted network.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable sending subnet-routes 128.185.123.22`

sending static-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous disable command. (See the **disable sending** command.) The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes should be advertised out a particular interface. A route is included in an RIP update only if it has been included by at least one of the enable sending commands. The **enable sending static-routes** command specifies that all statically configured and directly connected routes should be included in RIP updates sent out interface ip-interface-address.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable sending static-routes 128.185.123.22`

sending host-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous **disable ...** command. (See the **disable sending** command.) The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes should be advertised out a particular interface. A route is included in an RIP update only if it has been included by at least one of the **enable sending** commands. The **enable sending host-routes** command specifies that all host routes should be included in RIP updates sent out interface ip-interface-address.

source-routing

Allows the router to forward IP packets containing an IP source route option.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable source-routing`

udp-forwarding *port-number*

Enables UDP forwarding for packets received by the router with the specified UDP destination port number.

Default: UDP forwarding is disabled for all port numbers.

port-number

Valid Values: an integer in the range of 0 - 65535

Default Value: 0

Example: `enable udp-forwarding 36`

List

Use the **list** command to display various pieces of the IP configuration data, depending on the particular subcommand invoked.

Syntax: list
all
access-controls
addresses
bootp
filters
packet-filter
protocols
rip-routes-accept
routes
sizes
tags
udp-forwarding

all

Displays the entire IP configuration.

Example: `list all`

access-controls

Displays the configured access control mode (enabled or disabled) and the list of configured global access control records. Each record is listed with its record number. This record number can be used to reorder the list with the IP **move access-control** command.

Example: `list access control`

addresses

Displays the IP interface addresses that have been assigned to the router, along with their configured broadcast formats.

Example: `list addresses`

bootp

Indicates whether BOOTP forwarding is enabled or disabled as well as the configured list of BOOTP servers.

Example: `list bootp`

packet-filter *[filter-name]*

Lists information on packet filters. If you specify a name, the command lists access control information configured for the filter. If you do not specify a filter name, the command lists configured packet-filters.

Example: `list packet-filter pf-in-0`

```
Name               Direction     Interface
pf-in-0            In            0

Access Control is: enabled
List of access control records:

                                                  Beg  End Beg End
    Ty Source      Mask      Destination  Mask    Pro  Pro Prt Prt
1   E  128.185.0.0 FFFF0000  0.0.0.0      00000000 0   255  0  65535
2   I  0.0.0.0     00000000  0.0.0.0      00000000 0   255  0  65535
```

protocols

Displays the configured state of the IP routing protocols (OSPF, RIP, BGP) along with other general configuration settings.

Example: `list protocols`

rip-routes-accept

Displays the set of routes that the RIP routing protocol always accepts. See the IP configuration commands **enable/disable receiving dynamic nets/subnets/hosts** for more information.

Example: `list rip-routes-accept`

routes

Displays the list of static routes that have been configured.

Example: `list routes`

sizes

Displays the routing table size, reassembly buffer size, and the route cache size.

Example: `list sizes`

tags

Displays the per-interface tags that will be associated with received RIP information. These tags can be used to group routes together for later readvertisement via BGP where a tag will be treated as if it were a route's source autonomous system (AS). Tags are also propagated by the OSPF routing protocol.

Example: `list tags`

udp-forwarding

Displays all the configured information for the UDP Forwarding function, including all ports and all IP addresses.

Example: `list udp-forwarding`

Move

Use the **move** command to change the order of records in the global access control list. This command places record number from# immediately after record number to#. After you move the records, they are immediately renumbered to reflect the new order.

The router applies the access control records in a list in the order that they were created. For each packet received on an interface, the router applies each access control record in order until it finds a match. The first record that matches the packet determines whether it will be discarded, or forwarded to its destination.

This makes the order of the access control records very important. If they are in the wrong order, certain packets may slip through, or be blocked, in a manner contrary to your intentions.

Let us say, for example, that access control record 1 enforces the rule: *all packets from network 10.0.0.0 shall be blocked on this interface.* Contrary to this, access control record 2 states: *Packets from subnet 10.5.5.0 in network 10.0.0.0, which are destined for address 1.2.3.4, shall be allowed to pass.* Assigned in this order, these records will block all traffic from 10.0.0.0, even though record 2 explicitly allows certain types of packets to pass.

In this example, record 1 makes record 2 moot. Record 1 guarantees that the router discards all packets from 10.0.0.0, despite the intent of record 2, which is that certain packets be forwarded. The key to fixing this type of problem is in the order of the access control records. This way, packets in subnet 10.5.5.0 and destined for address 1.2.3.4 will pass through the interface; the router discards all other packets from 10.0.0.0 as intended.

Syntax: move   access-control *from# to#*

Example: `move 5 2`

Set

Use the **set** command to set certain values, routes, and formats within your IP configuration.

Syntax:
```
set     access-control . . .
        broadcast-address . . .
        cache-size
        default network-gateway . . .
        default subnet-gateway . . .
        internal-ip-address
        originate-rip-default
        reassembly-size
        router-id . . .
        routing table-size . . .
        tag . . .
        ttl
```

access-control *on* or *off*

Allows you to configure the router to enable or disable IP access control. Setting access-control *on* enables the global access control list as well as the interface specific lists. Setting it *off* disables all lists; but does not delete them

Example: `set access-control on`

broadcast-address *ip-interface-address style fill-pattern*

Specifies the IP broadcast format that the router uses when broadcasting packets out a particular interface. IP broadcasts are most commonly used by the router when sending RIP update packets.

The style parameter can take either the value local wire or the value network. Local-wire broadcast addresses are either all ones (255.255.255.255) or all zeros (0.0.0.0). Network style broadcasts begin with the network and subnet portion of the ip-interface-address.

You can set the fill-pattern parameter to either 1 or 0. This indicates whether the rest of the broadcast address (that is, other than the network and subnet portions, if any) should be set to all ones or all zeros.

When receiving the router recognizes all forms of the IP broadcast address.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

style

Valid Values: *local-wire* or *network*

Default Value: local-wire

fill-pattern

Valid Values: *0* or *1*

Default Value: 1

The example below configures a broadcast address of 255.255.255.255. The second example produces a broadcast address of 192.9.1.0, assuming that the network 192.9.1.0 is not subnetted.

Example:
```
set broadcast-address 192.9.1.11 local-wire 1
set broadcast-address 192.9.1.11 network 0
```

cache-size *entries*

Configures the maximum number of entries for the IP routing cache. This cache stores information about the specific IP addresses to which the router has recently forwarded packets. The cache reduces the processing time needed to forward multiple packets to the same destination.

In contrast with this cache, the IP routing *table* stores information about all accessible networks but does not contain specific IP destination addresses. Use the **set routing table-size** command to configure the size of the IP routing table.

Valid Values: 64 to 10000

Default Value: 64

Example:
```
set cache-size 64
```

default network-gateway *next-hop cost*

Configures a route to the authoritative router (default gateway). You should assume that the router's default gateway has more complete routing information than the router itself.

The route is specified by the IP address of the next hop (next-hop) and the distance (cost) to the default gateway.

All packets having unknown destinations are forwarded to the authoritative router (default gateway).

next-hop **Valid Values**: any valid IP address

Default Value: 0.0.0.0 with a gateway cost of 1.

cost **Valid Values**: an integer in the range of 0 - 255

Default Value: 1

Example:
```
set default network-gateway 192.9.1.10 10
```

default subnet-gateway *subnetted-network next-hop cost*

Configures a route to a subnetted network's authoritative router (default subnet gateway). You can configure a separate default subnet gateway for each subnetted network.

The IP address of the next hop (next-hop) and the distance (cost) to the default subnet gateway specify the route.

All packets destined for unknown subnets of a known subnetted network are forwarded to the subnetted network's authoritative router (default subnet gateway).

subnetted network

Valid Values: any valid IP address

Default Value: 0.0.0.0

next-hop

Valid Values: any valid IP address

Default Value: 0.0.0.0

cost

Valid Values: an integer in the range of 0 - 255

Default Value: 1

Example: `set default subnet-gateway 128.185.0.0 128.185.123.22 6`

internal-IP-address *ip-address*

Configures an IP address that is independent of the state of any interface. The internal address is always considered active. The primary reason for defining an internal address is to provide an address for a TCP connection that will not become inactive when an interface becomes inactive. This address is used for data link switching (DLSw), allowing alternate paths to be used to avoid disrupting DLSw connections when an interface becomes inactive. Because the internal address remains active and because OSPF maintains active IP routes to this destination, IP routing can switch DLSw traffic onto the alternate path without bringing down the TCP connection or disrupting the SNA sessions that are running on top of DLSw.

The internal IP address also provides some value when unnumbered interfaces are used. It is the first choice as a source address for packets originated by this router and transmitted over an unnumbered interface. The stability of this address makes it easier to keep track of such packets. The chance for confusion is further reduced when the same IP address is used for both the router ID and the internal address. Therefore the router ID will default to the internal address.

When an internal address is defined it will be advertised by OSPF as a host route into all areas directly attached to the router.

Valid Values: any valid IP address.

Default Value: none

Example: `set internal-ip-address 142.82.10.1`

originate-rip-default

Causes RIP to advertise this router as the default gateway. Use this command in the following environment:

- The IP routes in this router’s routing table are determined by a number of protocols.
- RIP is one of those protocols.
- At most partial routing information is imported from the other protocols and advertised by RIP.

Traffic in the RIP network for destinations that are not known by RIP can follow the default path to this router. The more complete routing information in this node’s route table can then be used to forward the traffic along an appropriate path towards its destination. You can configure the router to only originate the default when routes are known to this router that will not be advertised in the RIP network.

When you issue this command, you will be prompted as to whether to originate a rip-default for the other routing protocols your router is running.

This default route will direct traffic bound for a non-RIP network to a boundary router. Originating a single default route means that the boundary router does not have to distribute the other network's routing information to the other nodes in its network.

from AS number

Valid Values: an integer in the range of 0 - 65535

Default Value: none

to network number

Valid Values: any valid IP address

Default Value: none

default cost

Valid Values: an integer in the range of 0 - 255

Default Value: 1

Example: `set originate-rip-default`

```
IP Config> set originate rip-default
Always originate default route? [No]:?
Originate default if BGP routes available? [No] yes
  From AS number [6]?
  To network number [0.0.0.0]?
Originate default if OSPF routes available? [No]
Originate default cost [1]?
```

means a default route is always originated

- Answering “yes” to the “BGP” question originates a default whenever there are BGP routes in the routing table.
- Answering “yes” to the “if OSPF routes available” question causes the RIP default to be advertised when OSPF routes are in the routing table.
- When the router does decide to originate a RIP default, it uses the “original default cost” number.

reassembly-size *bytes*

Configures the size of the buffers that are used for the reassembly of fragmented IP packets.

Valid Values: 2048-65535

Default: 12000

Example: `set reassembly-size 12000`

router-id *ip-address*

Sets the default IP address used by the router when sourcing various IP packets. This address is of particular importance in multicasting and OSPF.

The router ID must match one of the configured IP interface addresses of the router or the configured internal IP address. If not, it is ignored. When ignored, or just not configured, the default IP address of the router (and its OSPF router ID) is set to the internal IP address (if configured) or to the first IP address in the router's configuration.

Valid Values: any valid IP address

Default Value: none

Example: `set router-id 128.185.120.209`

routing table-size *number-of-entries*

Sets the size of the router's IP routing table. The default size is 768 entries. Setting the routing table size too small causes dynamic routing information to be discarded. Setting the routing table size too large wastes router memory resources. See “Sizes” on page 17-8 for additional information about table sizes.

Valid Values: an integer number of entries in the range of 64 - 65535

Default Value: 768 entries

Example: **`set routing table-size 1000`**

tag

Configures the per-interface tags associated with received RIP information. These tags can be used to group routes together for later readvertisement via BGP where a tag will be treated as if it were a route’s source autonomous system (AS) number. (Refer to the section titled “Originate, Send, and Receive Policies” in the chapter “Using and Configuring BGP” in *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 1 Release 1*.) Tags are propagated also by the OSPF routing protocol.

Valid Values: an integer in the range of 0 - 65535

Default Value: 0

Example: **`set tag`**

```
Interface address [0.0.0.0]? 1.1.1.1
Interface tag (AS number) [0]? 1
```

ttl

Specifies the time-to-live for packets originated by the router.

Valid Values: a numeric in the range of 1 - 255

Default Value: 64

Example: **`set ttl 255`**

Update

Use the **update packet-filter** command at the `IP config>` prompt to assign access control entries. The router prompts you for the name of the filter you want to update. The `IP config>` prompt changes to incorporate the packet filter name you provide.

Valid Values: any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: none

```
IP Config> update packet-filter
Packet-filter name [ ]? pf-1-in
Packet-filter 'pf-1-in' Config>
```

You can access a list of sub-commands by typing **?** at the `Packet-filter 'name' Config>` prompt.

```
Packet-filter 'test' Config> ?
LIST
CHANGE
DELETE
ADD
MOVE
EXIT
```

Adding and Changing Access Controls to a Packet Filter

Use the **add access-control** command to add access controls to the specified packet filter. The router prompts you for the access control type (either Exclusive or Inclusive), and the source and destination addresses and masks of packets to which the filter will apply.

type **Valid Values**:

- *Exclusive*— Specifies that any packets matching one or more of the filters in the access control list for this interface will be dropped.
- *Inclusive*— Specifies that only packets matching one or more of the filters in the access control list for this interface will be forwarded.

Default Value: Exclusive

source address

Valid Values: A valid IP address in dotted decimal notation.

Default Value: 0.0.0.0

source mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: 255.255.255.255

destination address

Valid Values: A valid IP address in dotted decimal notation.

Default Value: 0.0.0.0

destination mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: 255.255.255.255

first protocol

The lower boundary of a protocol number range.

The commonly used protocol numbers are:

- **1** for ICMP
- **6** for TCP
- **17** for UDP
- **89** for OSPF.

See RFC 1340, "Assigned Numbers" for details on IP protocol numbers.

Valid Values: 0 to 255

Default Value: 0

last protocol

The upper boundary of a protocol number range.

The commonly used protocol numbers are:

- **1** for ICMP
- **6** for TCP
- **17** for UDP
- **89** for OSPF.

See RFC 1340, "Assigned Numbers" for details on IP protocol numbers.

Valid Values: 0 to 255

Default Value: 0

first port

The lower boundary of a IP TCP/UDP port range.

Valid Values: a port number in the range of 0 - 65535

Address Default Value: 0

Some commonly used port numbers are:

- 21 for FTP
- 23 for Telnet
- 25 for SMTP
- 513 for rlogin
- 520 for RIP

last port

The upper boundary of a IP TCP/UDP port range.

Valid Values: a port number in the range of 0 - 65535

Address Default Value: 0

Some commonly used port numbers are:

- 21 for FTP
- 23 for Telnet
- 25 for SMTP
- 513 for rlogin
- 520 for RIP

This example shows how to exclude all incoming packets originating from network 128.185.0.0 and received on interface 0.

```
Packet-filter 'pf-in-0' Config> add access-control
Enter type [E]?
Internet source [0.0.0.0]? 128.185.0.0
Source mask [255.255.255.255]? 255.255.0.0
Internet destination [0.0.0.0]?
Destination mask [255.255.255.255]? 0.0.0.0
Enter starting protocol number ([CR] for all) [-1]?
```

Use the **change access-control** command to change existing access controls using the index number of the access control that you want to change.

You can use the **list access-control** command to view the access controls configured for each packet filter.

```
Packet-filter 'pf-in-0' Config> list access-control
Access Control is: enabled
List of access control records:

                                                  Beg  End Beg End
    Ty Source      Mask      Destination  Mask    Pro  Pro Prt Prt
1   E  128.185.0.0 FFFF0000  0.0.0.0      00000000 0   255  0  65535
2   I  0.0.0.0     00000000  0.0.0.0      00000000 0   255  0  65535
```

You can change the order of a packet filter's access control records with the **move access-control** command as shown.

```
Packet-filter 'test' Config> move access-control
Enter index of control to move [1]?
Move record AFTER record number [0]? 2
About to move:

                                                Beg  End Beg End
    Ty Source      Mask      Destination  Mask    Pro  Pro Prt Prt
1   E  10.0.0.0    FFFF0000  0.0.0.0      00000000  0  255  0  65535
to be after:
2   I  10.5.5.0    FFFF0000  1.2.3.4      FF0000FF 0   255  0  65535
Are you sure this is what you want to do (Yes or [No]): y
```

Deleting Access Controls for a Filter

Use the **delete access-control** command to delete a record from a packet filter's access-control list.

```
Packet-filter 'test' Config> delete access-control
Enter index of access control to be deleted [1]? 4
```

The router responds by displaying the access-control record you have specified.

```
                                                Beg  End Beg End
    Ty Source      Mask      Destination  Mask    Pro  Pro Prt Prt
4   I  1.2.9.9     FF0000FF  0.0.0.0      00000000 0   255  0  65535
Are you sure this is the record you want to delete (Yes or [No]): y
Deleted
Packet-filter 'test' Config>
```

Exiting the Access Controls Process

Exit the access controls process by typing **exit** at the prompt. This returns you to the `IP config>` prompt.

```
Packet-filter 'test' Config> exit
IP config>
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Chapter 17. Monitoring IP

This chapter describes the IP console commands and includes the following sections:

- “Accessing the IP Console Environment”
- “IP Console Commands”

Accessing the IP Console Environment

For information on how to access the IP console environment, refer to “Getting Started (Introduction to the User Interface)” in the *Software User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

IP Console Commands

This section summarizes and then explains all the IP console commands. Table 17-1 lists the IP console commands. The commands allow you to monitor the router’s IP forwarding process. The monitoring capabilities include the following: configured parameters such as interface address and static routes can be viewed, the current state of the IP routing table can be displayed, and a count of IP routing errors can be listed.

Table 17-1 (Page 1 of 2). IP Console Command Summary

Command	Function
? (Help)	Lists the console commands or lists the actions associated with specific commands.
Access controls	List the current IP access control mode, together with the configured access control records.
Cache	Displays a table of all recent routed destinations.
Counters	Lists various IP statistics, including counts of routing errors and packets dropped.
Dump routing tables	Lists the contents of the IP routing table.
Interface addresses	Lists the router’s IP interface addresses.
Packet-filter	Displays the access-control information defined for the specified packet-filter, or all filters.
Ping	Sends ICMP Echo Requests to another host and watches for a response. This command can be used to isolate trouble in an internetwork environment.
Route	Lists whether a route exists for a specific IP destination, and if so, the routing table entry that corresponds to the route.
Sizes	Displays the size of specific IP parameters.
Static routes	Displays the static routes that have been configured. This includes the default gateway.
Traceroute	Displays the complete path (hop-by-hop) to a particular destination.

Table 17-1 (Page 2 of 2). IP Console Command Summary

Command	Function
UDP-Forwarding	Displays the UDP port numbers and destination IP addresses that you added using the **add** command or the **enable** command.
Exit	Exits the IP console environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ACCESS controls
CACHE
COUNTERS
DUMP routing tables
INTERFACE addresses
PACKET-FILTER summary
PING dest-addr [src-addr size ttl rate]
REDUNDANT default gateway
ROUTE given address
SIZES
STATIC routes
TRACEROUTE address
UDP-FORWARDING
EXIT
```

Access Controls

Use the **access controls** command to print the global access control mode in use together with a list of the configured access control records.

The access control mode is either disabled (meaning that no access control is being done and the access control records are being ignored) or enabled (meaning that access control is being done and the access control records are being recognized). When access control is enabled, access control records are scanned in order looking for the first match.

Exclusive (E) means that packets matching the access control record are being discarded. Inclusive (I) means that packets matching the access control record are being forwarded. When access control is enabled, packets failing to match any access control record are discarded. Pro (protocol) indicates the IP protocol number, and Port indicates the UDP or TCP destination port number.

Syntax: access

Example: **access**

```
Access Control currently enabled
Access Control run 13 times, 14 cache hits

List of access control records:
                                                    Beg End Beg  End
    Ty Source        Mask      Destination   Mask     Pro Pro Port Port   Use
1   E  10.5.22.0     FFFFFF00  0.0.0.0       00000000 0   255 0    65535  0
2   E  0.0.0.0       00000000  10.5.22.0     FFFFFF00 0   255 0    65535  0
3   I  0.0.0.0       00000000  0.0.0.0       00000000 0   255 0    65535  27
```

The IP access control system is based on a global list of *inclusive* and *exclusive* access control records. If access control is enabled, each IP packet being originated, forwarded, or received, is subject to the access control list.

The Use field (far right) specifies the number of times the access control system matched a particular record to an incoming packet, for example, the number of times that a particular record in the IP access controls system was invoked by the characteristics of an incoming or outgoing packet.

Cache

Use the **cache** command to display the IP routing cache which contains recently routed destinations. If a destination is not in the cache, the router looks up the destination in the routing information table in order to make a forwarding decision.

Syntax: cache

Example: **cache**

```
Destination     Usage          Next hop
128.185.128.225 1              128.185.138.180 (Eth/0)
192.26.100.42   1              128.185.138.180 (Eth/0)
128.185.121.1   18             128.185.123.18  (PPP/0)
128.185.129.219 76             128.185.125.25  (PPP/1)
128.185.129.41  130            128.185.125.25  (PPP/1)
128.185.129.134 546            128.185.125.40  (PPP/1)
128.185.129.221 1895           128.185.125.40  (PPP/1)
128.185.129.193 96             128.185.125.40  (PPP/1)
128.197.3.4     4              128.185.123.18  (PPP/0)
128.185.128.25  98             128.185.125.41  (PPP/1)
128.185.124.121 4              128.185.124.121 (Eth/0)
128.185.136.203 95             128.185.125.39  (PPP/1)
128.185.194.4   581            128.185.125.39  (PPP/1)
128.185.123.17  2              128.185.123.17  (PPP/0)
192.26.100.42   1              128.185.125.38  (PPP/1)
128.52.22.6     2              128.185.123.18  (PPP/0)
128.197.3.2     1              128.185.123.18  (PPP/0)
128.185.126.24  61             128.185.125.25  (PPP/1)
128.185.138.150 482            128.185.125.39  (PPP/1)
128.185.123.18  152            128.185.123.18  (PPP/0)
```

Destination IP destination host.

Usage Number of packets recently sent to the destination host.

Next hop IP address of the next router on the path toward the destination host. Also displayed is the network name of the interface used by the sending router to forward the packet.

Counters

Use the **counters** command to display the statistics related to the IP forwarding process. This includes a count of routing errors, along with the number of packets that have been dropped due to congestion.

Syntax: counters

Example: **counters**

```
Routing errors
Count   Type
    0   Routing table overflow
 2539   Net unreachable
    0   Bad subnet number
    0   Bad net number
    0   Unhandled broadcast
58186   Unhandled multicast
    0   Unhandled directed broadcast
 4048   Attempted forward of LL broadcast

Packets discarded through filter  0
IP multicasts accepted:           60592

IP input packet overflows
    Net   Count
   TKR/0  0
    FR/0  0
```

Routing table overflow
Lists the number of routes that have been discarded due to the routing table being full.

Net unreachable
Indicates the number of packets that could not be forwarded due to unknown destinations. This does not count the number of packets that have been forwarded to the authoritative router (default gateway).

Bad subnet number
Counts the number of packets or routes that have been received for illegal subnets (all ones or all zeroes).

Bad net number
Counts the number of packets or routes that have been received for illegal IP destinations (for example, class E addresses).

Unhandled broadcasts
Counts the number of (non-local) IP broadcasts received (these are not forwarded).

Unhandled multicasts
Counts the number of IP multicasts that have been received, but whose addresses were not recognized by the router (these are discarded).

Unhandled directed broadcasts
Counts the number of directed (non-local) IP broadcasts received when forwarding of these packets is disabled.

Attempted forward of LL broadcast
Counts the number of packets that are received having non-local IP addresses but were sent to a link level broadcast address. These are discarded.

Packets discarded through filter
Counts the number of received packets that have been addressed to filtered networks/subnets. These are discarded silently.

IP multicasts accepted
Counts the number of IP multicasts that have been received and successfully processed by the router.

IP packet overflows

Counts the number of packets that have been discarded due to congestion at the forwarder's input queue. These counts are sorted by the receiving interface.

Dump Routing Table

Use the **dump** command to display the IP routing table. A separate entry is printed for each reachable IP network/subnet. The IP default gateway in use (if any) is listed at the end of the display.

Syntax: dump

Example: **dump**

```
Type   Dest net        Mask     Cost Age   Next hop(s)
SPE1   0.0.0.0         00000000  4    3    128.185.138.39  (2)
SPF*   128.185.138.0   FFFFFF00  1    1    Eth/0
Sbnt   128.185.0.0     FFFF0000  1    0    None
SPF    128.185.123.0   FFFFFF00  3    3    128.185.138.39  (2)
SPF    128.185.124.0   FFFFFF00  3    3    128.185.138.39  (2)
SPF    192.26.100.0    FFFFFF00  3    3    128.185.131.10  (2)
RIP    197.3.2.0       FFFFFF00  10   30   128.185.131.10
RIP    192.9.3.0       FFFFFF00  4    30   128.185.138.21
Del    128.185.195.0   FFFFFF00  16   270  None


Default gateway in use.

Type Cost Age  Next hop
SPE1 4    3    128.185.138.39

Routing table size: 768 nets (36864 bytes), 36 nets known
```

Type (route type)

Indicates how the route was derived.

Sbnt - Indicates that the network is subnetted; such an entry is a place-holder only.

Dir - Indicates a directly connected network or subnet.

RIP - Indicates the route was learned through the RIP protocol.

Del - Indicates the route has been deleted.

Stat - Indicates a statically configured route.

BGP - Indicates routes learned through the BGP protocol.

BGPR - Indicates routes learned through the BGP protocol that are readvertised by OSPF and RIP.

Fltr - Indicates a routing filter.

SPF - Indicates that the route is an OSPF intra-area route.

SPIA - Indicates that it is an OSPF inter-area routes.

SPE1, SPE2 - Indicates OSPF external routes (type 1 and 2 respectively).

Rnge - Indicates a route type that is an active OSPF area address range and is not used in forwarding packets.

Dest net

IP destination network/subnet.

Mask

IP address mask.

Cost

Route Cost.

Age For RIP and BGP routes, the time that has elapsed since the routing table entry was last refreshed.

Next Hop IP address of the next router on the path toward the destination host. Also displayed is the interface type used by the sending router to forward the packet.

An asterisk (*) after the route type indicates the route has a static or directly connected backup. A percent sign (%) after the route type indicates that RIP updates will always be accepted for this network/subnet.

A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination. The first hops belonging to these routes can be displayed with the IP **route** command.

Interface Addresses

Use the **interface addresses** command to display the router's IP interface addresses. Each address is listed together with its corresponding hardware interface and IP address mask.

Hardware interfaces having no configured IP interface addresses will not be used by the IP forwarding process; they are listed as Not an IN net. There is one exception. Serial lines need not be assigned IP interface addresses in order to forward IP traffic. Such serial lines are called unnumbered. They show up as having address 0.0.0.0.

Syntax: interface

Example: **`interface`**

```
Interface  IP Address(es)   Mask(s)
   TKR/0   133.1.169.2      255.255.252.0
    FR/0   133.1.167.2      255.255.254.0
```

Interface Indicates the hardware type of the interface.

IP addresses Indicates the IP address of the interface.

Mask Indicates the subnet mask of the interface.

Packet-filter

Use the **packet-filter** command to display information defined for a specific packet filter, or for all filters. Packet-filters are interface-specific lists of access control records.

Syntax: packet-filter [*name*]

Example: **packet-filter pf-in-0**

```
Name               Direction     Interface   #Access-Controls
pf-in-0            In            0           2

Access Control currently enabled
Access Control run 8 times, 7 cache hits

List of access control records:

                                                   Beg End  Beg  End
  Ty  Source       Mask     Destination  Mask      PPP PPP  Port Port  Use
0 I   0.0.0.0      00000000 192.67.67.20 00000000  6   6    25   25    0
1 E   150.150.1.0  FFFFFF00 150.150.2.0  00000000  0   255  0    655   0
2 I   0.0.0.0      00000000 0.0.0.0      00000000  89  89   0    655   27
```

Ping

Use the **ping** command to have the router send ICMP Echo Requests to a given destination (that is, “pinging”) and watch for a response. This command can be used to isolate trouble in an internetwork environment.

This process is done continuously, incrementing the ICMP sequence number with each additional packet. Each matching received ICMP Echo response is reported with its sequence number and the round-trip time. The granularity (time resolution) of the round trip time calculation is usually around 20 milliseconds, depending on the platform.

To stop the pinging process, type any character at the console. At that time, a summary of packet loss, round-trip time, and number of unreachable ICMP destinations will be displayed.

When a multicast address is given as destination, there may be multiple responses printed for each packet sent, one for each group member. Each returned response is displayed with the source address of the responder.

The size of the ping (number of data bytes in the ICMP message, excluding the ICMP header), TTL value, and frequency of pinging are all user configurable. The source IP address is also configurable. If not specified, the router uses its local address on the outgoing interface to the specified destination. If you are validating connectivity from any of the router’s other interfaces to the destination, enter the IP address for that interface as the source address.

The default values are a size of 56 bytes, a TTL of 64 seconds, and a frequency of 1 ping per second. Only the destination value is required, all other values are optional.

Syntax: ping *dest-addr [src-addr size ttl rate]*

Example: **ping 128.185.142.06 128.185.142.11 56 60 1**

```
PING 128.185.142.11 -> 128.185.142.06: 56 data bytes, ttl = 60 every 1 sec.
56 bytes from 128.185.142.06: icmp_seq=0 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=1 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=2 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=3 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=4 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=5 time=0 ms

----128.185.142.06 PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0 ms.
```

Route

Use the **route** command to display the route (if one exists) to a given IP destination. If a route exists, the IP addresses of the next hops are displayed, along with detailed information concerning the matching routing table entry. (See the IP **dump** command.)

Syntax: route *ip-destination*

Example: **`route 133.1.167.2`**

```
Destination:    133.1.166.0
Mask:           255.255.254.0
Route type:     SPF
Distance:       1
Age:            1
Tag:            0
Next hop(s):    133.1.167.2       (FR/0)
```

Example: **`route 128.185.230.0`**

```
Destination:   128.185.230.0
Mask:          255.255.255.0
Route type:    SPF
Distance:      1
Age:           1
Next hop(s):   128.185.230.0    (TKR/0)
```

Example: **`route 128.185.232.0`**

```
Destination:   128.185.232.0
Mask:          255.255.255.0
Route type:    RIP
Distance:      3
Age:           0
Next hop(s):   128.185.146.4   (Eth/0)
```

Sizes

Use the **sizes** command to display the configured sizes of specific IP parameters.

Syntax: sizes

Example: **`sizes`**

```
Routing table size:        768
Table entries used:        3
Reassembly size:           12000
Largest reassembled pkt:   0
Size of routing cache:     64
# of cache entries in use: 0
```

Routing table size The configured number of entries that the routing table will maintain.

Table entries used The number of entries used from the routing table. This number includes both active and inactive entries. The value displayed using the “dump” command as “xx nets known” is the number of active routing table entries. The configured routing table size should be large enough to maintain current active entries as well as other anticipated routing entries.

Reassembly buffer size The configured size of the reassembly buffer that is used to reassemble fragmented IP packets.

Largest reassembled pkt The largest IP packet that this router has had to reassemble.

Size of routing cache The configured size of the routing cache.

# of cache entries in use The number of entries currently being used from the cache.

Static Routes

Use the **static routes** command to display the list of configured static routes. Configured default gateways and default subnet gateways are also listed.

Each static route’s destination is specified by an address-mask pair. Default gateways appear as static routes to destination 0.0.0.0 with mask 0.0.0.0. Default subnet gateways also appear as static routes to the entire IP subnetted network.

The example below shows a configured default gateway, a configured default subnet gateway (assuming 128.185.0.0 is subnetted), and a static route to network 192.9.10.0.

Syntax: static

Example: **static**

```
Net            Mask           Cost  Next hop
0.0.0.0        0.0.0.0        1     128.185.123.18
128.185.0.0    255.255.0.0    1     128.185.123.22
192.9.10.0     255.255.255.0  10    128.185.123.22
```

Net The network address of the route.

Mask The subnet mask of the IP address.

Cost The cost of using this route.

Next Hop The next router a packet would pass through using this route.

Traceroute

Use the **traceroute** command to display the entire path to a given destination, hop by hop. For each successive hop, **traceroute** sends out three probes, and prints the IP address of the responder, together with the round trip time associated with the response. If a particular probe receives no response, an asterisk is displayed. Each line in the display relates to this set of three probes, with the leftmost number indicating the distance from the router executing the command (in router hops).

The traceroute is done whenever the destination is reached, an ICMP Destination Unreachable is received, or the path length reaches 32 router hops.

When a probe receives an unexpected result, several indications can be displayed. “!N” indicates that an ICMP Destination Unreachable (net unreachable) has been received. “!H” indicates that an ICMP Destination Unreachable (host unreachable) has been received. “!P” indicates that an ICMP Destination Unreachable (protocol unreachable) has been received; because the probe is a UDP packet sent to a strange port, a port unreachable is expected "!" indicates that the destination has been reached, but the reply sent by the destination has been received with a TTL of 1. This usually indicates an error in the destination, prevalent in some versions of UNIX, whereby the destination is inserting the probe's TTL in its replies. This unfortunately leads to a number of lines consisting solely of asterisks before the destination is finally reached.

Syntax: traceroute *interface-address*

Example: **`traceroute 128.185.142.239`**

```
TRACEROUTE 128.185.124.110: 56 data bytes
 1 128.185.142.7 16 ms 0 ms 0 ms
 2 128.185.123.22 16 ms 0 ms 16 ms
 3 * * *
 4 * * *
 5 128.185.124.110 16 ms ! 0 ms ! 0 ms !
```

TRACEROUTE Displays the destination area address and the size of the packet being sent to that address.

1 The first trace showing the destination's NSAP and the amount of time it took the packet to arrive at the destination. The packet is traced three times.

Destination unreachable Indicates that no route to destination is available.

*1 * * **
*2 * * ** Indicates that the router is expecting some form of response from the destination, but the destination is not responding.

UDP-Forwarding

Use the **UDP-forwarding** command to display the UDP port and addresses that you added using the **add udp-destination** command or the **enable udp-forwarding** command.

Syntax: udp-forwarding

Example: **`udp-forwarding`**

```
UDP Port   IP Address
    35      20.2.1.1
    20      22.2.1.2
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Chapter 18. Using and Configuring IPX

This chapter describes how to use the IPX protocol on your 2216 and how to configure the IPX protocol using the IPX configuration commands. It includes the following sections:

- "IPX Overview"
- "Configuring IPX" on page 18-2
- "Optional Configuration Tasks" on page 18-3
- "Accessing the IPX Configuration Environment" on page 18-15
- "IPX Configuration Commands" on page 18-15

IPX Overview

IBM's implementation of IPX allows the router to function as a Novell NetWare internetwork router. It has these characteristics:

- Compatibility with all previous Novell NetWare version environments.
- Compatibility with the bridging function in a NetWare file server, plus a standalone NetWare bridge.
- Support for the Novell NetBIOS emulator.

IPX Addressing

Every IPX interface must have a unique address that corresponds to the network number assigned to the attached IPX network. A simple solution is to use multipart addresses like the city-street-house address on a piece of mail. For example, IPX refers to network numbers (city), host numbers (street), and socket numbers (house).

Network Numbers

An IPX network number specifies the location of a particular network in an internetwork. These addresses allow communication between two entities on different networks.

Host Numbers

Each IPX interface needs a 6-byte host (node) number.

Token-Ring and Ethernet interfaces use their hardware MAC address as their host number, and you cannot change them.

Because serial lines have no hardware MAC addresses, you must specify a unique host number.

ATM interfaces use their End System Identifier (ESI) as their host number. Their burned-in ESI will be used if one has not been configured.

Configuring IPX

This section describes how to initially configure IPX. The following sections describe optional parameters you can set.

1. Display the IPX configuration prompt as shown here:

   ```
   * talk 6
     Config> protocol ipx
     IPX protocol user configuration
     IPX config>
   ```

2. Enable the IPX protocol on the router.

   ```
   IPX Config>enable ipx
   ```

3. Enable IPX and assign an IPX network number on each interface on which you want to run IPX. Every interface must have a unique network number that corresponds to the network number that is assigned to the attached IPX network.

   Repeat this step for each interface.

   ```
   IPX Config>enable interface
   Which interface [0]? 0
   Configure an IPX network number for this interface.
   Network number in hex [1]? 180
   ```

   Note: You cannot enter 0 or 0xFFFFFFFF as network numbers. Novell has reserved these numbers.

4. If you have enabled IPX to run over a serial interface, assign a unique host number to the router.

   ```
   IPX config>set host-number
   Host number for serial lines (in hex) []? 2
   ```

5. Optionally change the frame type for Ethernet or Token Ring. You do not have to set the frame type for interfaces other than Ethernet or Token Ring. See "Frame" on page 18-23 for a description of available frame types.

   The default encapsulation formats are:

   - Ethernet - Ethernet_8023
   - Token Ring - Token-ring MSB

   Use the **frame** command as shown here:

   ```
   IPX config> frame ethernet_8023
   Which interface [0]? 1
   ```

6. If you are running IPX over PPP, configure IPXWAN as follows:

   a. Enable IPXWAN.

   ```
   IPX config>enable ipxwan
   Which interface [0]? 2
   Connection Timeout (in sec) [60]? 60
   Retry timer (in sec) [60]? 60
   ```

   b. Set the router name. IPXWAN requires that a router have a name. You can enter up to 47 characters including A through Z, underscore (_), hyphen (-), and the "at" sign (@).

   ```
   IPX config>set name
   Router Name []? newyork_accounting
   ```

c. Assign a primary network number. IPXWAN requires the router to have a primary network number. Enter a 1- to 8-digit hexadecimal number.

```
IPX config>set nodeid
Node ID in hex [1]? 23
```

Optional Configuration Tasks

Optional settings that you can adjust are described in the following sections.

- "Specifying the Size of IPX RIP Network Table"
- "Specifying RIP Update Interval"
- "Specifying the Size of IPX SAP Services Table" on page 18-4
- "Specifying SAP Update Interval" on page 18-4
- "IPX Keepalive and Serialization Packet Filtering" on page 18-4
- "Configuring Multiple Routes" on page 18-5
- "Configuring Global IPX Filters (IPX Access Controls)" on page 18-5
- "Global SAP Filters" on page 18-7
- "IPX Interface Filters - Overview" on page 18-8
- "IPX Performance Tuning" on page 18-11
- "Split-Horizon Routing" on page 18-13

Specifying the Size of IPX RIP Network Table

The IPX RIP network table contains information about each IPX network. The default table size is 32. You can configure the table size from 1 to 2048; however, there may be memory limitations on the router that can prevent the maximum table size from being used.

```
IPX config>set maximum networks
New Network table size [32]? 32
```

Specifying RIP Update Interval

IPX uses RIP to maintain routes in its routing tables. A route indicates the path a packet follows. The RIP update interval determines how often the router broadcasts its routing information tables to its interfaces. It also determines how long a RIP entry remains before being aged-out.

Valid entries remain in the routing tables for a period of three multiples of the RIP update interval, and the router broadcasts its RIP tables once every update interval.

For example, the default interval is 1 minute, which allows a valid entry to remain in the table for 3 minutes. After this time, if an entry is not refreshed by a RIP update, the route is marked with a hop count of infinity (16) and then it is deleted. Every 60 seconds the router broadcasts its RIP tables to corresponding interfaces.

You can configure the RIP interval from 1 to 1440 minutes (24 hours). Increasing the RIP interval reduces traffic on WAN lines and dial circuits. It also prevents dial-on-demand circuits from dialing out as often.

Note: While complete RIP advertisements are controlled by the interval, the router still propagates network topology changes as quickly as it learns them.

The RIP interval is not configurable on the Novell file server.

```
IPX config>set rip-update-interval
Which interface [0]? 2
RIP timer value(minutes) [1]? 2
```

Specifying the Size of IPX SAP Services Table

The IPX Service Advertising Protocol (SAP) services table is a distributed database used to find NetWare Services, such as file servers. Services are uniquely identified by a 2-byte numeric type and a 47-character name. Each service provider advertises its services, specifying service type, name, and address. The router accumulates this information in a table and sends it to other routers. The default table size is 32.

You can configure the table size from 1 to 2048; router memory constraints may prevent the maximum table size from being used.

```
IPX config>set maximum services
New Service table size [32]? 32
```

Specifying SAP Update Interval

The IPX Service Advertising Protocol (SAP) interval lets you configure the time between IPX SAP updates on a per-interface basis. All router interfaces on the same network must use the same SAP interval. This interval determines both the age-out time for table information, and the interval between broadcasts to router interfaces.

You can configure the SAP interval from 1 to 1440 minutes (24 hours). Increasing the SAP interval reduces traffic on WAN lines and dial circuits. It also prevents dial-on-demand circuits from dialing out as often.

Note: While complete SAP advertisements are controlled by this interval, the router still propagates network topology changes as quickly as it learns them.

The SAP interval is not configurable on the Novell file server.

```
IPX config>set sap-update
Which interface [0]? 2
SAP timer value(minutes) [1]? 4
```

IPX Keepalive and Serialization Packet Filtering

You can configure IPX to prevent keepalive and serialization packets from continually activating a dial-on-demand link or to minimize traffic over a dial-on-demand link.

In Figure 18-1 on page 18-5, for example, if the Novell Client logs into the Novell Server and then remains idle, the server sends periodic keepalive requests to the client and the client replies with keepalive replies. Keepalive filtering causes the routers to enter the first keepalive reply into their keepalive tables and then forward the reply. After that, the routers do not forward keepalive traffic for that client-server connection over the WAN link. Instead, Router A replies to keepalive requests it receives from the server and Router B sends keepalive requests to the Novell Client.

Keepalive filtering also prevents the routers from forwarding NetWare serialization packets over the WAN link.

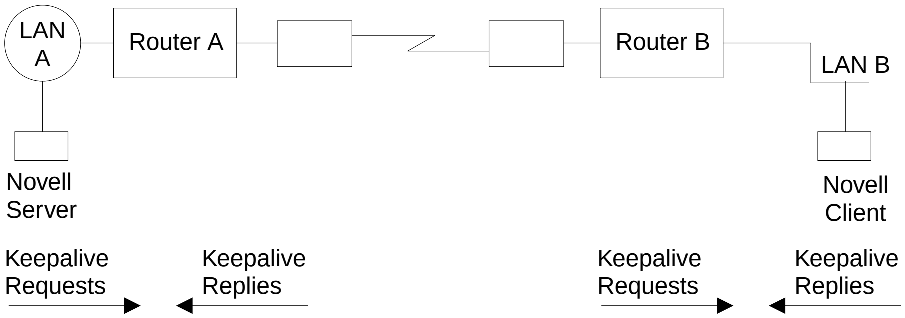


Figure 18-1. Keepalive Filtering

To set up Keepalive filtering, enable it on the dial circuits.

```
IPX Config> enable keepalive-filtering
which interface [0]? 5
```

Configuring Multiple Routes

You can configure IPX so that it keeps more than one routing table entry for the same destination network. The benefit of this feature is that if a route goes down, the alternate route is used immediately. The router does not have to wait for a RIP broadcast, which could take from a few seconds to a minute, to learn a new route. The router stores only equal-cost paths in the routing table.

Use the following command to configure the maximum number of routes that will be stored in the routing table for each destination. The range is 1 to 64. The default is 1.

```
IPX config>set maximum routes-per-destination
New maximum number of routes per destination net [1]? 4
```

Use the following command to set the total number of entries kept in the routing table. The range is 1 to 4096. The default is 32. Set the number of entries to at least the same size as the RIP network table. (Configure the size of the RIP network table using the **set maximum networks** command explained in this chapter.)

```
IPX config> set maximum total-route-entries
New route table size [32]? 40
```

Configuring Global IPX Filters (IPX Access Controls)

Global IPX filters are applied to all IPX interfaces. They can be used to prevent the router from forwarding packets based on IPX addresses (network/host/socket). You can use global IPX filters to provide security or to stop the forwarding of packets from “noisy” applications beyond the area of interest.

Global IPX filters are based on the originating IPX source address and the ultimate destination IPX address. Intermediate hop addresses are not important.

An IPX address (source or destination) for a global filter consists of an IPX network number, an IPX host number, and a range of IPX socket numbers that are specified in hexadecimal. The network number and host number can be specified as 0, which is a wildcard that matches all network and host numbers, respectively. A range of 0 to FFFF is a wildcard for sockets.

The global filter list is an ordered list of entries. Each global filter entry can be configured as inclusive or exclusive. The router compares packets it receives against the global filter list.

- If a packet matches an inclusive entry, the router forwards the packet.
- If a packet matches an exclusive entry, the router drops the packet.
- If the router reaches the end of the list without matching the packet to an entry, the router drops the packet. (This is equivalent to having a wildcard exclusive entry at the end of the list.)

When creating global filter lists, consider the following things about IPX:

- First, never block the RIP and SAP sockets (0x0453 and 0x0452). RIP and SAP are required to correctly forward IPX packets.
- Remember that the global filter list applies to all interfaces. You will have to use source and/or destination network numbers in the global filters to enact directional controls.
- Understand where the services you are trying to protect are located. At the `IPX>` prompt, enter the **slist** command to determine the address of a service.

  Note: All services on a Novell file server (version 3.0 or higher) are on the server's internal network, usually at host 0000 0000 0001. Because that internal network number is unique over an entire IPX network, you can protect it by blocking all packets to the internal network socket range 0–FFFF. To block only the file server, use a socket range of 0451–0451.
- When extracting socket numbers from an **slist** to build a global filter list, remember that some services have fixed socket numbers and some have dynamic (temporary) socket numbers. Because sockets in the range 4000–7FFF are dynamic, there is no guarantee that the service will have the same socket number the next time the file server is rebooted. However, socket numbers in the range 8000–FFFF are assigned by Novell, and will generally remain constant.

Note: The global filters and interface filters are mutually-exclusive. If global SAP filtering is enabled, interface SAP filters cannot be enabled (and vice versa). If global IPX filtering is enabled (*access-controls*), interface IPX filters cannot be enabled (and vice versa).

The router examines each IPX frame to see if it matches an entry in the global filter list. It applies the first match, therefore the order of global filters is critical. The router examines IPX packets for the following criteria:

1. Type of global filter (two types):
   a. Inclusive, indicating that if the packet matches the following criteria, forward it
   b. Exclusive, indicating that if the packet matches the following criteria, discard it
2. Destination network - taken directly from the packet's IPX destination network field.
3. Destination host - taken directly from the packet's IPX destination host field.

4. Starting/Ending destination socket - taken directly from the packet's IPX destination socket field (not host field). (The socket number is the location within the protocol that binds the packet to an application service.)
5. Source network - taken directly from the packet's IPX source network field.
6. Source host - taken directly from the packet's IPX source host field.
7. Starting/Ending source socket - taken directly from the packet's IPX source socket field.

The result of the following example would be to forward only those IPX packets from any client on IPX net 1871, destined for the NCP application, on the Novell File Server 0000 C93A 0912, on network 18730. All other traffic would be dropped.

```
IPX config>add access control
Enter type [E]? I
Destination network number (in hex) [ ]? 18730
Destination host number (in hex) [ ]? 0000C93A0912
Starting destination socket number (in hex) [ ]? 0451
Ending destination socket number (in hex) [ ]? 0451
Source network number (in hex) [ ]? 1871
Source host number (in hex) [ ]? 0
Starting source socket number (in hex) [ ]? 4000
Ending source socket number (in hex) [ ]? 7FFF
```

Global SAP Filters

Global SAP filters apply to all interfaces. They can be used to prevent service advertising information from being propagated through the router. There are four primary reasons to use global SAP filters:

- You are using servers with small bindery sizes (for example, NetWare Version 2.15 or lower) and must limit the amount of information in the SAP database.
- You do not want to advertise certain services outside the local area, because remote access to them would be inappropriate.
- You want to remove clutter from the SAP table.
- You want to reduce needless SAP advertisements on WAN links, since SAP advertisements can consume a considerable amount of WAN bandwidth.

Note: None of these reasons explicitly mentions security. Global SAP filters cannot protect a service. All that SAP does is provide a name-to-address translation for services. If a potential intruder knows the address of the service, blocking its advertisement via global SAP filters will not protect the service. Only access controls can provide security.

The global SAP filter is based on setting a maximum hop count for a particular service, or group of services. Any matching service advertisement received with the specified hop count (or less) is accepted into the SAP table. Others are ignored. Only those services in the SAP database are re-advertised or used to answer queries.

Note: The router allows you to enter service names in 7-bit ASCII only. Some service names use binary data, in violation of Novell SAP specifications. You will not be able to filter those services by name.

A global SAP filter can apply to all services of a type. Novell assigns 4-digit hexadecimal type numbers for each type of service. Alternately, a global SAP filter can apply to one particular service of a type. This is done by specifying the name of the service.

There can be several servers of the same service type, each with a unique service name. In this case, you can configure multiple global SAP filters with the same service type to filter unique service names, or you can configure a single SAP filter which filters the service type for all service names (wildcard filter).

Creating Global SAP Filters

To configure global SAP filters:

1. Enter **add filter** at the `IPX Config>` prompt. You must specify several key entries that are normally found in the SAP broadcasts:
   a. Number of hops. This entry indicates the hop count allowed for a SAP entry (if higher, discard).
   b. Service type
   c. Service name
2. Enter **set filter on** at the `IPX Config>` prompt to enable the filter.

The following example shows the creation of a global SAP filter against a specific print server.

```
IPX config>add filter
Maximum number of hops allowed [1]? 2
Service type [4]? 0047
Optional service name [ ]? rem-ptr1
IPX config> set filter on
```

This global SAP filter causes the router to ignore SAP advertisements from any print server (service type 0047) named **rem-ptr1** that is more than two hops away. The filter prevents the router from propagating advertisements that match these criteria.

Determining the Service Type for a Global SAP Filter

To determine the SAP type for a filter you want to establish, follow these steps:

1. At the * prompt, enter **talk 5**. Then, at the + prompt, enter **protocol ipx**.

   At the `IPX>` prompt enter **slist**. Note the entry for the services you want to filter.
2. At the `*` prompt, enter **talk 6**. Then, at the `Config>` prompt, enter **protocol ipx**. Add the appropriate global SAP filter and the appropriate hop count for the service you want to filter.
3. After creating the filter, restart the router.
4. If you have successfully filtered a service, it should no longer be listed. Check that the service is no longer listed by entering **slist** at the `IPX>` prompt.

IPX Interface Filters - Overview

The IPX routing feature supports four types of interface-based filters: ROUTER, RIP, SAP, and IPX. One *input* and one *output filter* can be defined per interface. Filter criteria, referred to as *items*, are assembled into *filter-lists* and are then attached to the input and/or output filters. A filter-list can be attached to more than one filter. This prevents you from having to configure the same filter criteria on multiple interfaces.

Note: The global filters and interface filters are mutually-exclusive. If global SAP filtering is enabled, interface SAP filters cannot be enabled (and vice versa). If global IPX filtering is enabled (*access-controls*), interface IPX filters cannot be enabled (and vice versa).

Configuring IPX Interface Filters

To configure IPX Interface Filters:

1. Create a filter-list and give it a name, using the **create list** command.
2. Modify the filter-list using the **update** command and its subcommands to specify the filter criteria and whether this filter-list is inclusive or exclusive.
3. Create a filter on the desired interface using the **create filter** command, specifying whether it is an input or output filter.
4. Enable the filter using the **enable** command.
5. Attach filter-lists to the filter using the **attach** command.
6. Set the default action for the filter using the **default** command. The default action will be taken if no match is made on any of the attached filter-lists.

There are also commands to delete a filter on a network interface, disable a filter on a network interface (or all network interfaces), detach a filter-list from a filter, move the filter-lists within the filter (because the filter-lists are ordered), list a filter, and set the size of the filter cache (for IPX Filtering only).

ROUTER Filtering

The ROUTER Filter operates on the IPX header of all received RIP response packets. Output ROUTER filtering is not supported. ROUTER filtering can be used to group individual IPX networks into several distinct IPX internets by controlling which routers are allowed to exchange routing information.

RIP Router Filters are kept in ordered lists of items by interface. The items are applied in order to each received RIP response packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = discard packet, Include = receive packet for processing). Because Excluded packets are discarded, the information contained in their network entries is not entered into the RIP routing tables. If no match is found, the specified default filter action is performed.

RIP Filtering

The RIP filter operates on the network entries of RIP response packets. It can be used to control the extent to which routing information about selected networks is disseminated. As an *input* filter, this filter can prevent the *storing* of routing information about selected networks. This prevents ***all*** other networks from learning about the selected networks (at least through this router).

RIP filters (input) are kept in ordered lists of items by interface. The items are applied in order to each network entry in each received RIP response packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = ignore network entry, Include = process network entry). Because Excluded network entries are ignored, they are not entered into the RIP routing tables. If no match is found, the specified default filter action is performed.

As an *output* filter, this filter can prevent the *advertising* (as opposed to the storing) of routing information about selected networks. It prevents *some* (as opposed to all) networks from learning about the selected networks (at least through this router).

RIP filters (output) are kept in ordered lists of items by interface. The items are applied in order to each network entry to be transmitted in a RIP response packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = exclude network entry from packet, Include = include network entry in packet). This filter has no effect on the contents of the RIP routing tables. If no match is found, the specified default filter action is performed.

SAP Filtering

The SAP filter operates on the server entries of all SAP response packets. It can be used to control the extent to which information about services is disseminated, and can reduce the amount of SAP traffic on lower speed WANs.

As an *input* filter, this filter can prevent the *storing* of service information about selected servers. This prevents ***all*** other networks from learning about the selected servers (at least through this router).

SAP filters (input) are kept in ordered lists of items by interface. The items are applied in order to each server entry in each received SAP response packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = ignore server entry, Include = process server entry). Because Excluded server entries are ignored, they are not entered into the SAP services table. If no match is found, the specified default filter action is performed.

As an *output* filter, this filter can prevent the *advertising* (as opposed to the storing) of service information about selected servers. This prevents *some* (as opposed to all) networks from learning about the selected servers (at least through this router).

SAP filters (output) are kept in ordered lists of items by interface. The items are applied in order to each server entry in each SAP response packet to be transmitted. If a match is found, the action specified in the matching filter-list is performed (Exclude = exclude server entry, Include = include server entry in packet). This filter has no effect on the contents of the SAP services table. If no match is found, the specified default filter action is performed.

IPX Filtering

The IPX Filter operates on the IPX header of IPX packets. It can be used to control the extent to which selected servers and workstations are allowed to communicate with other selected servers and workstations, based on source and destination network, node, and socket fields, as well as protocol type and hop count.

As an *input* filter, a match that indicates that the packet should be discarded prevents the packet from being transmitted on ***all*** interfaces.

IPX Filters (input) are kept in ordered lists of items by interface. The items are applied in order to each received IPX packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = discard packet, Include = receive packet for processing or forwarding). If no match is found, the specified default filter action is performed.

As an *output* filter, the decision whether to forward the packet is made based on the output interface, and therefore might allow a received packet to be forwarded out on one interface but not out on some other interface.

IPX filters (output) are kept in ordered lists of items by interface. The items are applied in order to each IPX packet to be transmitted. If a match is found, the action specified in the matching filter-list is performed (Exclude = discard packet, Include = transmit packet). If no match is found, the specified default filter action is performed.

Because IPX filters are invoked for each received packet, it is recommended that they be used only where a high degree of specificity is required (that is, where the RIP Router, RIP and SAP filters cannot be used). Generally, the RIP filters deal with internetworking between ***all*** stations on a particular set of networks; the SAP filters control which servers are reachable by workstations throughout the internetwork; the IPX filters deal with internetworking between ***individual*** workstations (or individual applications on individual workstations).

“ IPX Interface Filter Configuration Commands” on page 18-30 describes in more detail the commands used to configure IPX Interface Filters.

IPX Performance Tuning

The IPX router implements a dual path for packet forwarding, a fast path and a slow path, to route traffic more efficiently.

The fast path forwards only data packets, while a slower path handles administration packets, such as RIP and SAP packets. Fast path uses an address cache that enables the router to forward a packet quickly.

The slower routing table lookups are performed only during the creation of a cache entry. The cache has an aging mechanism that allows overflows to be dealt with intelligently. You can configure the cache size through the IPX configuration menu.

The IPX fast path cache includes two entries: local and remote. Each entry can handle the requirements of that type of addressing.

The cache commands are used to set a limit on the maximum number of entry types allowed in the cache.

Local Cache

The size of the local cache should equal the total number of clients on each router’s local or client network plus a 10% buffer to prevent excessive purge requests. Using the example in Figure 18-2 on page 18-13, router 5 (RTR R5) has 9 clients (C) plus the server (S) for a total of 10. Based on this total:

1. Multiply by 10% (10 in our example).
2. Add that total (1) to the client total (for a safety margin).
3. Use the new total (11) for the number of local cache entries.

For example:

```
IPX config>set local-cache size
New IPX local node cache size [32]? 11
```

When all cache entries are in use, the least frequently used entries are purged.

Remote Cache

The size of the remote cache should equal the total number of remote networks used by the router plus a 10% buffer to prevent excessive purge requests. In Figure 18-2 on page 18-13, there are 10 IPX networks that RTR R5 can read via IPX network 5. Therefore, RTR/R5 has a total of 10 clients. Based on this total:

1. Multiply by 10% (10 in our example).
2. Add that total (1) to the remote network total (10) for a safety margin.
3. Use the new total (11) for the number of remote cache entries.

For example:

```
IPX config>set remote-cache size
New IPX remote network cache size [32]? 11
```

You can view the cache entries using the IPX monitoring **sizes** command.

```
IPX>sizes
Current IPX cache size:
Remote network cache size (max entries): 45
0 entries now in use

Local node cache size(max entries): 86
0 entries now in use
```

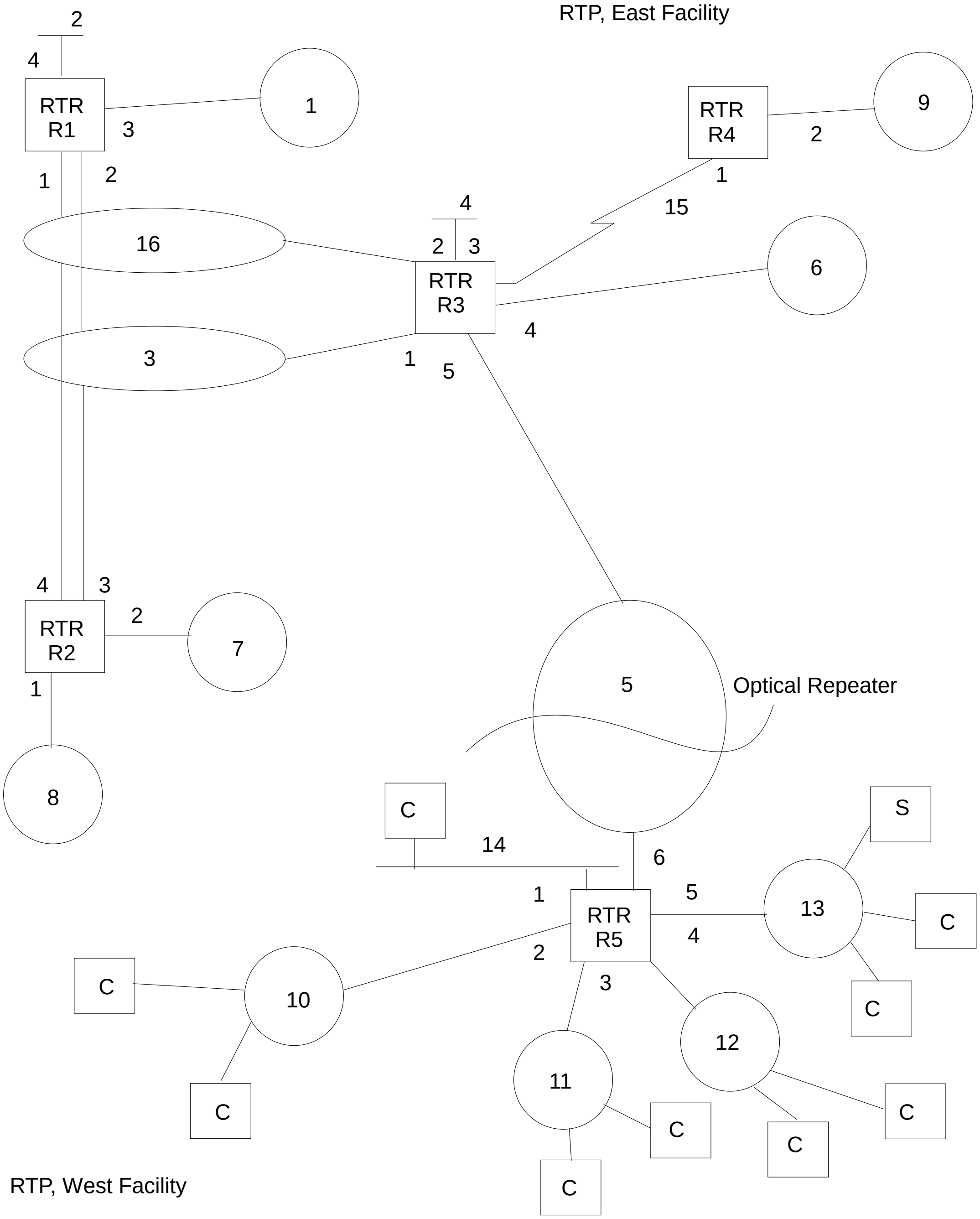


Figure 18-2. Sample IPX Network

Split-Horizon Routing

Split-horizon is a method of routing that avoids broadcasting RIP and SAP updates to the router from which they were learned.

Generally, split-horizon should be enabled on every interface to prevent packets from counting to infinity and to avoid unnecessary RIP and SAP advertisements. However, there are some cases, such as partially-meshed frame-relay, ATM, and X.25 configurations, where it may be necessary to disable split-horizon.

A Partially-meshed RFC 1483-Supported IPX Routing configuration is another case where it may be necessary to disable split-horizon.

In a partially-meshed frame-relay network, as shown in Figure 18-3 on page 18-14, the routers at the branches cannot communicate with each other unless the router at headquarters broadcasts all routing information to all other routers. In this case, split-horizon should be disabled on the frame-relay interface at headquarters, and enabled at each of the branches to keep them from generating unnecessary traffic.

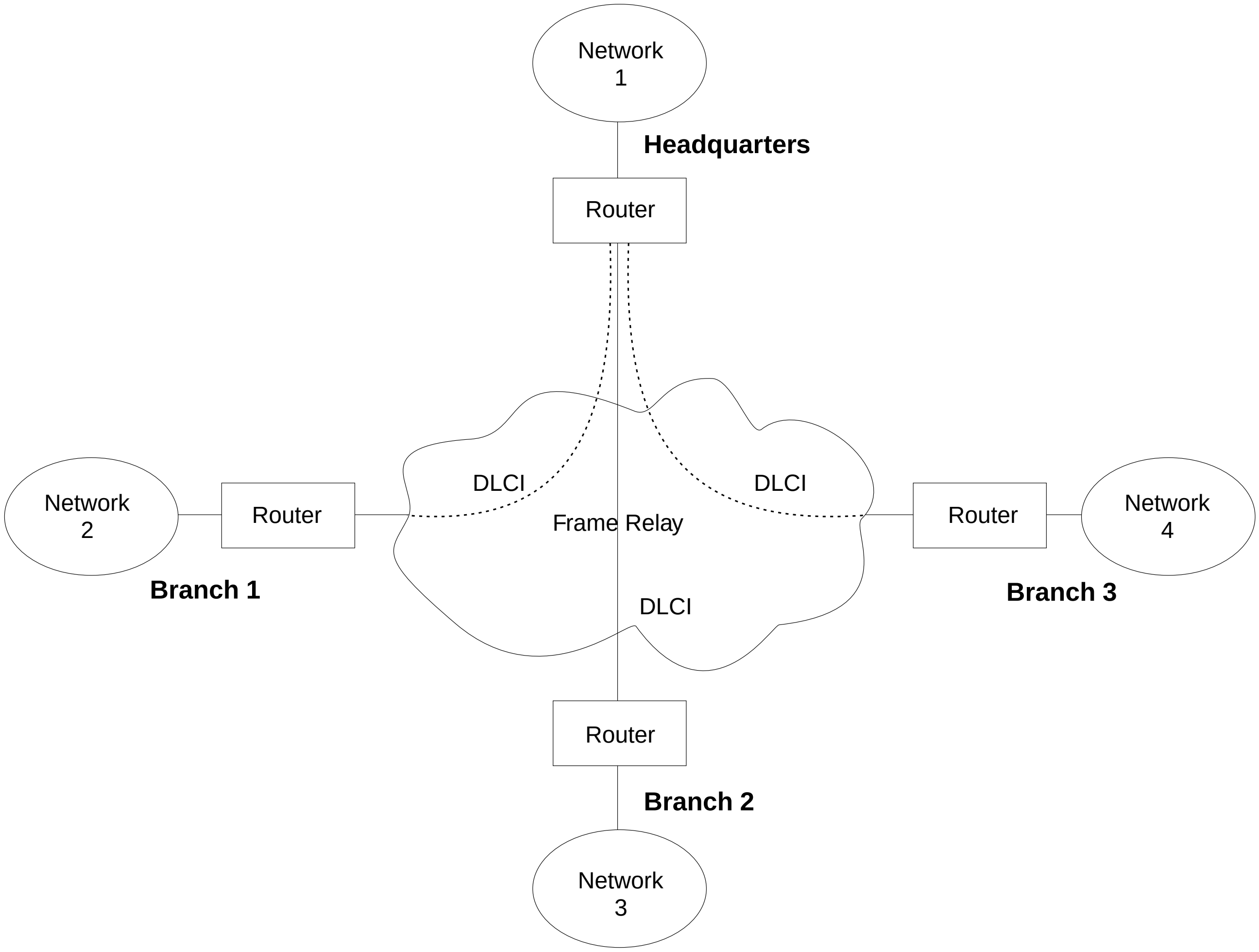


Figure 18-3. Partially Meshed Frame-Relay Network

If you do need to change the split-horizon setting, use the **set split-horizon** command as follows:

```
IPX Config>set split-horizon enabled
Which interface [0]? 2

IPX Config>set split-horizon disabled
Which interface [0]? 2

IPX Config>set split-horizon heuristic
Which interface [0]? 2
```

Accessing the IPX Configuration Environment

To access the IPX configuration environment, enter the following command at the `Config>` prompt:

```
Config> protocol IPX
IPX Protocol user configuration
IPX Config>
```

IPX Configuration Commands

This section discusses the IPX configuration commands. Table 18-1 lists the IPX configuration commands. These commands specify the network parameters for router interfaces transmitting IPX packets. These commands are entered at the `IPX config>` prompt. To activate the configuration changes, restart the router.

Table 18-1. IPX Configuration Commands Summary

Command	Function
? (Help)	Lists all of the IPX configuration commands or lists the options associated with specific commands.
Add	Adds global IPX filters (access controls), and global SAP filters.
Delete	Deletes global IPX filters (access controls), and global SAP filters.
Disable / Enable	Disable or enable IPX globally or on specific interfaces, disable or enable Keepalive filtering, disable or enable reply to SAP get-nearest-server requests, disable or enable RIP-SAP broadcast pacing, and disable or enable IPXWAN on specific interfaces.
Filter-lists	Accesses the `IPX` *filter-type*`-List Config>` prompt. This is the environment where the IPX interface filters (Router, RIP, SAP, and IPX) are configured.
Frame	Specifies the data link format for Ethernet and Token-Ring interfaces. This also applies to Token-Ring and Ethernet LAN Emulation Clients.
List	Displays the current IPX configuration.
Move	Changes the line numbers set when adding access control.
Set	Sets the host number, IPXWAN router name and node ID, IPXWAN connection timeout and retry timer, IPX network numbers, maximum RIP and SAP table sizes, local and remote cache sizes, global IPX filter (access controls) and global SAP filter states, cache sizes, RIP and SAP update intervals, Keepalive filtering table size, and split-horizon usage.
Exit	Exits the IPX configuration process and returns to the CONFIG environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
disable
enable
exit
frame
list
set
add
delete
move
filter-lists
```

Add

Use the **add** command to add global IPX filters (access controls) to your IPX configuration, which determines whether the router drops or forwards IPX packets. The **add** command also adds global SAP filters to your IPX configuration; this determines which SAP service advertisements will be ignored or accepted by the router.

Syntax: add access-control . . .
filter . . .

access-control *type dest-net dest-host dest-socket-range src-net src-host src-socket-range*

Determines whether to pass a packet at the IPX level. IPX access controls provide a global access control function at the IPX packet level for the IPX protocol. The access control list is an ordered set of entries that the router uses to filter packets. Each entry can be either Inclusive or Exclusive. Each entry has source and destination network numbers, host addresses, and socket ranges.

When a packet is received from a network for the IPX protocol, and access control is enabled, it is checked against the access control list. It is compared with the net/address/socket pairs in the list until there is a match. If there is a match and the entry is of the Inclusive type, reception of the packet (and potential forwarding) proceeds. If the matching entry is of the Exclusive type, the packet is dropped. If there is no match, the packet is also dropped.

After you create an access-control list with the **add access-control** command, enable the entries with the **set access-control on** command. Use the **move** command to change the order of the access-control list.

Note: Access controls apply to all received packets. If you do not enable reception of RIP (socket 453 hexadecimal) or SAP (socket 452 hexadecimal) packets, the IPX forwarder will be nonfunctional.

```
add access I 0 0 453 453 0 0 0 FFFF
add access I 0 0 452 452 0 0 0 FFFF
```

```
Enter type [E] i
Destination network number (in hex) [0]? 0
Destination host (in hex) [ ]? 0
Starting destination socket number in hex [0]? 452
Ending destination socket number in hex [0]? 453
Source network number (in hex) [0]? 0
Source host number (in hex) [ ]? 0
Starting source socket number in hex [0]? 0
Ending source socket number in hex [452]? FFFF
```

Type

Identifies whether packets are sent or dropped for a specific address or set of addresses. Enter I for include. This causes the router to receive the packet and to forward it if it matches criteria in the remaining arguments. Enter E for exclude. This causes the router to discard the packets.

Dest-net

Network number of the destination. Enter the network number in hexadecimal.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Zero (0) specifies all networks.

Default Value: 0

Dest-host

Host number on the destination network. Enter the host number in hexadecimal.

Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Zero (0) specifies all hosts on the network.

Default Value: None

Dest-socket-range

Two numbers that specify an inclusive range of destination sockets. The destination socket value is used for filtering IPX packets.

Valid Values: X'0000' to X'FFFF'

Default Value: 0

Src-net

Network number of the source. Enter the network number in hexadecimal.

This parameter defines the network number of the source IPX network whose packets are filtered by this router.

If you choose to filter on *only* the source network value, the filter applies to all source sockets, source networks, packet types, and number of hops.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Zero (0) specifies all networks.

Default Value: 0

Src-host

Host number on the source network. Enter the host number in hexadecimal.

Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Zero (0) specifies all hosts on the network.

Default Value: None

Src-socket-range
Two numbers that specify an inclusive range of source sockets.

Valid Values: X'0000' to X'FFFF'

Default Value: 0

Note: It is not necessary to use access controls and SAP filters for IPX to work in a NetWare environment. Use them only if necessary.

Example: `add access-control E 201 1 451 451 329 0 0 FFFF`

This access control prevents all nodes on network 329 from accessing the file server with internal network number 201.

filter *hops service-type service-name*
Prevents NetWare bindery overflows for users on large networks by enabling you to determine the number of hops reasonable for a given service. IPX SAP filters allow the protocol to be configured to ignore certain entries in SAP advertisements. This is done to limit the size of the SAP database. This could be necessary due to size limitations in older versions of NetWare file servers. This could also be necessary to limit the amount of SAP data sent across WAN links.

The SAP filters are a global ordered list of filter entries. Each filter entry has a maximum hop count, a service type, and an optional service name. When a SAP response packet is received, each SAP entry is compared with the filter list. If the SAP entry matches an entry in the filter list and is greater than the specified hops, it is ignored and not entered into the local SAP database. If the SAP entry matches an entry in the filter list, and is less than or equal to the specified hops, it is accepted and entered into the local SAP database. If there is no match, the SAP entry is accepted. The arguments for this command are as follows:

Hops
Maximum number of hops permitted for the service.

Valid Values: An integer in the range of 0 - 16.

Default Value: 1

Service-type
Numeric service class.

Valid Values: A hexadecimal value in the range of 0000 - FFFF.

Use a value of X'0000' to filter all service types.

Default Value: 4

You can see a list of service types by entering the **slist** command at the `IPX>` prompt.

Service-name

Identifies the name of the server. In general, this field is not entered.

Valid Values:

A string of 1 to 47 ASCII characters (X'20' through X'7E'), with the exception of the following special characters: plus (+), minus (-), comma (,), semicolon (;), colon (:), slash (/), and back slash (\).

The question mark (?) and asterisk (*) characters serve as wildcard characters. The question mark may be used multiple times to represent any single character within the server name. The asterisk may be used multiple times to represent any portion of the server name. The question mark and asterisk may also be used together.

Default Value: none

Example: `add filter 2 039B NOTES-CHICAGO`

This example ignores all SAP advertisements for the Lotus Notes server “NOTES-CHICAGO” at more than 2 hops.

Delete

Use the **delete** command to delete a global IPX filter (access control) or global SAP filter.

Syntax: delete access-control . . .
filter . . .

access-control *line#*

Deletes the access control that matches the line number you enter. Enter the **list** command to display the current line numbers.

Example: `delete access-control 2`

filter *hops service-type service-name*

Deletes the specified SAP filter. You must type the SAP filter exactly as it appears when you run the list command. The arguments are as follows:

Hops

Maximum number of hops permitted for the service.

Valid Values: 0 to 16

Default Value: 16

Service-type

Numeric service class. Enter a 2-byte hexadecimal number.

Valid Values: X'0000' to X' FFFF'

Default Value: none

Service-name

If the entry you are deleting has a name, specify the name.

Valid Values: A string of 1 to 47 ASCII characters (X'20' through X'7E'), with the exception of the following special characters: plus (+), minus (-), comma (,), semicolon (;), colon (:), slash (/), and back slash (\).

The question mark (?) and asterisk (*) characters serve as wildcard characters. The question mark may be used multiple times to represent any single character within the server name. The asterisk may be used multiple times to represent any portion of the server name. The question mark and asterisk may also be used together.

Default Value: none

Example: `delete filter 2 039B NOTES-CHICAGO`

Disable

Use the **disable** command to disable IPX on specific interfaces, or to globally disable the IPX protocol. Also, use the **disable** command to disable replies to SAP get-nearest-server requests, RIP-SAP Broadcast Pacing on specific interfaces, or IPXWAN on specific interfaces.

Syntax: disable interface . . .
ipx
ipxwan . . .
reply-to-get-nearest-server . . .
rip-sap-pacing . . .
keepalive-filtering . . .

interface *interface#*

Prevents the router from sending IPX packets over specific interfaces.

Example: `disable interface 2`

ipx

Prevents the router from sending IPX packets over any of the interfaces.

Example: `disable ipx`

ipxwan *interface#*

Disables IPXWAN on specific serial interfaces using the point-to-point protocol (PPP).

Example: `disable ipxwan 2`

reply-to-get-nearest-server *interface#*

Prevents the router from responding to SAP get-nearest-server requests from workstations that are attempting to locate a server.

Note: Disabling this feature should be done with great caution. This command should be used only when there are multiple routers (or servers) on an IPX network and it is known that the “best” server is not behind this router.

Example: `disable reply  3`

rip-sap-pacing *interface#*

Prevents the router from pacing RIP and SAP periodic broadcast packets on specific interfaces. When pacing is disabled, RIP and SAP periodic broadcasts are transmitted on the interface with a 55 msec interpacket gap (the default setting). Enable pacing only on interfaces where RIP and SAP broadcasts might cause congestion (for example, you can enable pacing on frame-relay or X.25 interfaces with many virtual circuits).

Example: `disable rip-sap-pacing 3`

keepalive-filtering *interface#*

Disables Keepalive filtering as described on page 18-4.

Example: **`disable keepalive-filtering`**

```
Which interface [0]? 5
```

Enable

Use the **enable** command to enable IPX on specific interfaces, or to globally enable the IPX protocol. Also use the **enable** command to enable replies to SAP get-nearest-server requests, RIP-SAP Broadcast Pacing on specific interfaces, or IPXWAN on specific interfaces.

Syntax: enable interface . . .
ipx
ipxwan . . .
keepalive-filtering . . .
reply-to-get-nearest-server . . .
rip-sap-pacing . . .

interface *interface# network#*

Allows the router to send IPX packets over specific interfaces. Every interface must have a unique network number that corresponds to the network number that is assigned to the attached IPX network. You will be prompted for a valid IPX network number if one has not already been configured.

Example: **`enable interface 2  4`**

ipx

Allows the router to send IPX packets over all of the interfaces on which IPX has been enabled.

Example: **`enable ipx`**

ipxwan *interface# timeout retry_timer*

Enables IPXWAN on specific interfaces using the Point-to-Point protocol (PPP). This command also queries for a connection timer value and a retry timer value. The enable command prompts for the same parameters as the **set ipxwan** command. This allows you to initially set IPXWAN parameters without having to use the set command. If you need to modify pre-configured parameters, then use the **set ipxwan** command.

timeout

This value specifies the IPX connection timeout period. A connection will time out if no IPXWAN packets are exchanged within the number of seconds specified by this parameter.

Valid values: An integer number of seconds in the range of 5 - 300.

Default Value: 60 seconds

retry_timer

This parameter specifies the amount of time to wait after a connection is timed out before trying to re-establish the connection.

Valid values: An integer number of seconds in the range of 5 - 600.

Default Value: 60 seconds

Example: `enable ipxwan 0 60 60`

```
Which interface [0]? 0
Connection Timeout (in sec) [60]?60
Retry Timer (in sec) [60]? 60
```

keepalive-filtering *interface#*

Enables Keepalive filtering as described on page 18-4.

Example: `enable keepalive-filtering`

```
        Which interface [0]? 5
```

reply-to-get-nearest-server *interface#*

Allows the router to respond to SAP get-nearest-server requests from workstations that are attempting to locate a server. This is the default setting.

Note: Disabling this feature should be done with great caution. The **disable reply-to-get-nearest-server** command should be used only when there are multiple routers (or servers) on an IPX network and it is known that the “best” one is not behind this router.

Example: `enable reply  2`

rip-sap-pacing *interface#*

Enables pacing of RIP and SAP periodic broadcast packets on specific interfaces. When pacing in enabled, RIP and SAP periodic broadcasts are transmitted on the interface with an interpacket gap calculated by the router (a value from 55 msec to 5 seconds).

Note: The router calculates an interpacket gap that guarantees that broadcast completion within the configured RIP and SAP update intervals. Configuring these intervals to a larger value may be necessary for the router to calculate a sufficiently large interpacket gap.

Pacing should be enabled only on interfaces where RIP and SAP broadcasts might cause congestion (for example, on frame-relay or X.25 interfaces with many virtual circuits).

Example: `enable rip-sap-pacing 3`

Filter-lists

Use the **filter-lists** command to access the `IPX` *`filter-type`*`-List Config>` prompt. Valid filter list types are router, rip, sap, and ipx.

For information about the commands available at the `IPX` *`filter-type.`*`-List Config>` prompt, see “ IPX Interface Filter Configuration Commands” on page 18-30.

Syntax: filter-lists router
rip
sap
ipx

Example: `filter-lists router`

Frame

Use the **frame** command to specify the packet format for IPX interfaces. (Encapsulation can also be set using the CONFIG **network** command.)

Note: When there are incorrect or invalid configuration records, the default frame values are used.

```
Syntax:  frame    ethernet_II . . .
                  ethernet_8022 . . .
                  ethernet_8023 . . .
                  ethernet_SNAP . . .
                  token-ring MSB . . .
                  token-ring LSB . . .
                  token-ring_SNAP MSB. . .
                  token-ring_SNAP LSB. . .
                  fddi
                  fddi_snap
```

ethernet_type *interface#*

Selects the Ethernet encapsulation format. This is required if you are using NetWare-VMS on the Ethernet, and is often used when there are ISO nodes on the same Ethernet. The following options are available:

- ethernet_II (default of NetWare 4.0 and greater) - uses Ethernet version 2.0 protocol 81-37.
- ethernet_8022 - uses Ethernet 802.3 with 802.2 SAP E0.
- ethernet_8023 (default of pre-NetWare 4.0 and lower) - uses Ethernet 802.3 without any 802.2 header.
- ethernet_SNAP - uses 802.3, 802.2 with SNAP PID 00-00-00-81-37.

  Note: The ethernet_SNAP encapsulation it is not architecturally valid and is not fast-pathed. No cache entries will appear for network entries using this encapsulation.

The default value for Ethernet frames is “ethernet_8023.”

Example: `frame ethernet_II 1`

token-ring_type *interface#*

Selects the token-ring encapsulation format. The default value is “token-ring MSB.” The following options are available:

- token-ring MSB (router default) - uses 802.5 with 802.2 SAP E0, and uses the noncanonical format for host addresses in the IPX packet header. The router builds outgoing packets with a three-byte 802.2 header (0xE0, 0xE0, 0X03).
- token-ring LSB - uses 802.5 with 802.2 SAP E0, and uses the canonical format for host addresses in the IPX packet header.
- token-ring_SNAP MSB - uses 802.5, 802.2 with SNAP PID 00-00-00-81-37, and uses the noncanonical format for host addresses in the IPX packet header.
- token-ring_SNAP LSB - uses 802.5,802.2 with SNAP PID 00-00-00-81-37, and uses the canonical format for host addresses in the IPX packet header.

Example: `frame token-ring_SNAP MSB 3`

```
which interface [0]? 3
```

Example: **`frame token-ring SNAP PID 00-00-00-81-37`**

fddi

Selects the IPX encapsulation to FDDI IEEE 802.2.

Example: `IPX Config>` **`fddi`**

fddi_snap

Selects the IPX encapsulation to FDDI SNAP.

Example: `IPX Config>` **`fddi_snap`**

List

Use the **list** command to display the current IPX configuration.

Syntax: list

Example: **`list`**

```
IPX globally                enabled
Host number (serial line)   020000000200
Router Name (IPXWAN)
NodeID (IPXWAN)             0
Maximum networks            32
Maximum total route entries             128
Maximum routes per dest. network        3
Maximum services                        32
Maximum Network Cache entries           64
Maximum Local Cache entries             64

List of configured interfaces:
                Frame                  SAP nearest   Split
Ifc  IPX net #  Encapsulation          server reply  Horizon    IPXWAN

0    177        TOKEN-RING      MSB    Enabled       Heuristic  N/A
1    183        N/A                    Enabled       Heuristic  N/A
5    184        N/A                    Enabled       Heuristic  N/A

RIP/SAP Timer Intervals and Pacing:

                 SAP Interval  RIP Interval
Ifc  IPX net #   (Minutes)     (Minutes)     Pacing
---  ---------   ------------  ------------  -------
0        177        1             1          Disabled
1        183        1             1          Disabled
5        184        30            30         Enabled


IPX SAP Filter is: disabled
No IPX SAP Filter records in configuration.
IPX Access Controls are: disabled
No IPX Access Control records in configuration.
IPX KeepAlive Proxy Reply connection table size: 32
IPX Keepalive Filtering/Proxy Reply on interface 5
```

IPX globally Indicates whether IPX is globally enabled or disabled.

Host number The IPX host number to be assigned to serial interfaces. You can change this number with the IPX **set** command.

Router name The user-assigned IPXWAN router name.

Node ID The user-assigned IPXWAN node-id.

Maximum networks The size of the IPX RIP network table, which is the maximum number of IPX networks.

Maximum routes
The size of the IPX RIP routes table, which is the maximum number of routes to IPX networks.

Maximum routes-per-network
The configured number of maximum routes-per-network.

Maximum services
The size of the IPX SAP service table, which is the maximum number of IPX servers.

Maximum network cache entries
The maximum number of network cache entries.

Maximum local cache entries
The maximum number of local cache entries.

List of configured interfaces
The following is displayed for each interface on which IPX is enabled:

- Interface number
- IPX network number
- Type of encapsulation
- Whether reply to SAP get-nearest-server requests is enabled
- Whether split-horizon is enabled, disabled, or heuristic.
- Whether IPXWAN is enabled.

RIP/SAP Timer Intervals and Pacing
The following is displayed for each interface on which IPX is enabled:

- Interface number
- IPX network number for a particular interface
- Delay in minutes between complete RIP advertisements
- Delay in minutes between complete SAP advertisements
- Whether RIP-SAP broadcast pacing is enabled.

IPX SAP filter
Indicates whether the Global SAP filter is enabled or disabled, and lists the configured Global SAP filters.

IPX access controls
Indicates whether the Global IPX filters (access controls) are enabled or disabled, and lists the configured Global IPX filters (access controls).

IPX KeepAlive Proxy Reply connection table size
Number of entries the IPX KeepAlive table holds. Change this size using the **set keepalive-table-size** command.

IPX KeepAlive Filtering/Proxy Reply on interface
Shows which interfaces have IPX Keepalive and Packet Serialization filtering enabled.

Move

Use the **move** command to change the line numbers for the Global IPX filters (access controls) After you move the lines, they are renumbered to reflect the new order.

Syntax: move   access-control *line#* *line#*

Example: **move 5 2**

```
About to move:
 # T Dest Net         Host  Sck  Sck  Src Net         Host  Sck  Sck
 5 E       30 020000000006   30   32        2 020000000004   45   46
to be after:
 2 I     2487 020300000008    0    0    45230 020000000042    0    0
            Are you sure this is what you want to do(Yes or [No]):
```

Set

Use the **set** command to configure many of the operational parameters of the IPX protocol, such as the IPX network number for each interface, the sizes of the RIP and SAP tables, the sizes of the routing caches, and the IPXWAN parameters.

Syntax: set

access-control . . .
filter . . .
host-number . . .
ipxwan . . .
local-cache size . . .
maximum routes-per-destination . . .
maximum networks . . .
maximum services . . .
maximum total-route-entries . . .
name . . .
net-number . . .
node-id . . .
remote-cache size . . .
rip-update-interval . . .
sap-update-interval . . .
split-horizon . . .
keepalive-table-size . . .

access-control *on* or *off*

Turns the global IPX filters (access controls) on or off. Enter **on** or **off**.

Example: `set access-control on`

filter *on* or *off*

Turns the global SAP filters on or off. Enter **on** or **off**.

Example: `set filter on`

host-number *host#*

Specifies the host number used for serial interfaces running IPX. Each IPX router operating over serial interfaces must have a unique host number. This is required because serial interfaces do not have hardware node addresses from which to build a host number.

Valid Values: An 12-digit hexadecimal number in the range of 0000 0000 0001 - FFFF FFFF FFFE.

Default Value: none

This number must be unique on each router.

Example: `set host-number 0000000000F4`

Note: IPXWAN requires a router node-ID and name to be configured. Use the **set node-ID** and **set name** commands to configure these parameters. In addition, in order to interoperate with the IBM 6611, the node address used on the IPXWAN interface must be set to the node-ID ***followed by four zeros***. For example, if the IPXWAN node-ID is 454, the host-number must be set to 4540000. Use the **set host-number** command to configure this parameter.

ipxwan *interface# timeout retry_timer*

Sets up or modifies an interface to use the IPXWAN protocol when starting IPX on a serial interface using the PPP. Before the **set ipxwan** command can be invoked, IPXWAN must be enabled using the **enable ipxwan** command. This command also queries for a connection timer value and a retry timer value.

connection timeout

This value specifies the IPX connection timeout period. A connection will time out if no IPXWAN packets are exchanged within the number of seconds specified by this parameter.

Valid values: An integer number of seconds in the range of 5 - 300.

Default Value: 60 seconds

retry timer

This parameter specifies the amount of time to wait after a connection is timed out before trying to re-establish the connection.

Valid values: An integer number of seconds in the range of 5 - 600.

Default Value: 60 seconds

Example: **`set ipxwan`**

```
Which interface [0]? 1
Connection Timeout (in sec) [60]?60
Retry timer (in sec) [60]? 60
```

local-cache size *size*

Specifies the size of the local cache routing table.

The size of the local cache should equal the total number of clients on each router’s local or client network plus a 10% buffer to prevent excessive purge requests.

Valid Values: The range is 1 to 10000.

Default Value: 64. For more information, see “Local Cache” on page 18-11 and “Remote Cache” on page 18-12.

Example: **`set local-cache size`**

```
New IPX local node cache size [64]? 80
```

maximum routes-per-destination *routes*

Specifies the maximum number of routes per destination network to store in the IPX RIP routes table.

Valid Values: An integer in the range of 1 - 64.

Default Value: 1. For additional information on multiple routes, see “Configuring Multiple Routes” on page 18-5.

Example: **`set maximum routes-per-destination 8`**

maximum networks *size*

Specifies the size of the IPX RIP network table. This reflects the number of networks in the internet on which IPX operates.

Valid Values: 1 - 2048

Router memory constraints can prevent the maximum table size from being used.

Default Value: 32 This value cannot be larger than the maximum total-route-entries *size*.

Example: **`set maximum networks 30`**

maximum services *size*

Specifies the size of the IPX SAP service table. This reflects the number of SAP services in the internetwork on which IPX operates.

Valid Values: 1 - 2048

Router memory constraints can prevent the maximum table size from being used.

Default Value: 32

Example: `set maximum services 30`

maximum total-route-entries *size*

Specifies the size of the IPX RIP routes table. This reflects the total number of routes, including alternate routes, in the internetwork on which IPX operates.

Valid Values: 1 to 4096

Default Value: 32

This value must be at least as large as the *maximum networks size*. For additional information of multiple routes, see “Configuring Multiple Routes” on page 18-5.

Example: `set maximum total-route-entries 40`

name *router_name*

Lets you assign a symbolic name to the router. IPXWAN requires a router to have a primary network number and a name.

Valid Values: A variable length string of 1 to 47 characters.

The router_name can contain the characters A through Z, 0 through 9, underscore (_), hyphen (-), and "at" sign (@).

Default Value: none.

Example: `set name newyork_accounting`

net-number *interface#* *ipx-net#*

Assigns an IPX network number to the associated directly-connected network. Every IPX interface must have a unique network number.

Valid Values: 1 to 8 hexadecimal digits in the range of 1 - FFFF FFFE.

Do not assign 0 or FFFF FFFF.

Default Value: none

Example: `set net-number 2 180`

node-id *primary-net#*

Lets you assign a primary network number. IPXWAN requires a router to have a primary network number and a name. The “node-id” is the primary network number for the router and must be assigned before the exchange of IPXWAN packets can begin.

Valid Values: 1 to 8 hexadecimal digits in the range of 1 - FFFF FFFE.

Default Value: none

Do not assign 0 or FFFF FFFF as the *primary-net#.* This number is for the router as a whole. In NetWare file server terms, it is the “internal” network number. This number must be unique among all the network numbers in the IPX internet.

Example: **`set node-id 23`**

remote-cache size *size*

Specifies the size of the remote cache routing table.

The size of the remote cache should equal the total number of remote networks used by the router plus a 10% buffer to prevent excessive purge requests.

Valid Values: The range is 1 to 10000.

Default Value: 64.

Example: **`set remote-cache size`**

```
        New IPX remote network cache size [64]? 80
```

rip-update-interval *interface# minutes*

Specifies the time delay in minutes between complete RIP updates given on an interface.

Increasing the RIP interval reduces traffic on WAN lines and dial circuits. It also prevents dial-on-demand circuits from dialing out so often.

Note: While complete RIP advertisements are controlled by the interval, the router still propagates network topology changes as quickly as it learns them.

Valid Values: The range is from 1 to 1440 minutes.

Default Value: 1 minute. For additional information on RIP interval, see “Specifying RIP Update Interval” on page 18-3.

Example: **`set rip-update-interval`**

```
  Which interface [0]? 0
  RIP Timer Value (minutes) [1]?2
```

sap-update-interval *interface# minutes*

Specifies the time delay in minutes between complete SAP updates given on an interface.

Valid Values: The range is from 1 to 1440 minutes.

Default Value: 1 minute.

Example: **`set sap-update-interval`**

```
  Which interface [0]? 0
  SAP Timer Value (minutes) [1]? 2
```

split-horizon *value interface#*

Specifies the type of split-horizon on the specified interface.

The value *enabled* enables split-horizon on the specified interface. The value *disabled* disables split-horizon on the specified interface. The value *heuristic* enables split-horizon on all types of interfaces except frame-relay. For frame-relay, split-horizon is enabled only if the specified interface has exactly one PVC defined, otherwise split-horizon is disabled.

Generally, split-horizon should be set to *enabled*. It is sometimes necessary to disable split-horizon for partially-meshed frame-relay, X.25, and ATM configurations. For additional information on split-horizon, see "Split-Horizon Routing" on page 18-13.

Example: **`set split-horizon enabled 0`**

keepalive-table-size *value interface#*

Sets the number of entries that the Keepalive table holds. These entries include all current client/server and server/server pairs connected over the WAN link. The default is 32. The range is 1 to 250.

Example: **`set keepalive-table-size`**

```
which interface[0]? 0
Number of entries [32]?
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Accessing the IPX Interface Filter Configuration Environment

To access the IPX Interface Filter configuration environment, enter the following command at the `IPX config>` prompt:

```
IPX Config> filter-lists type
IPX type-List Config>
```

Where *type* is the type of IPX filter to be configured. Valid types are *router, rip, sap,* and *ipx*.

IPX Interface Filter Configuration Commands

This section lists and then explains the commands to configure the IPX interface-based filters; ROUTER, RIP, SAP, and IPX. To configure these filters, enter the `filter-lists` *`type`* command at the `IPX Config>` prompt, and then enter the configuration commands at the `IPX` *`type`*`-List Config>` prompt.

Table 18-2 (Page 1 of 2). IPX Filter Configuration Command Summary

Command	Function
? (Help)	Lists all interface filter configuration commands or lists the options associated with specific commands.
Attach	Attaches a specified filter-list to a specified filter.

Table 18-2 (Page 2 of 2). IPX Filter Configuration Command Summary

Command	Function
Create	Creates a filter or filter-list.
Default	Sets the default action of a filter to *include* or *exclude*
Delete	Deletes a filter or filter-list.
Detach	Detaches a filter-list from a filter.
Disable	Disables filtering.
Enable	Enables filtering.
List	Displays the current filtering configuration.
Move	Reorders filter-lists attached to a filter.
Set-cache	Sets the caching size for a specified filter.
Update	Accesses the `IPX` *`type`*`-List` *`filter-list`* `Config>` prompt.
Exit	Exits the IPX Interface Filter configuration environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
attach
create
default
delete
detach
disable
enable
list
move
set-cache
update
exit
```

Attach

Use the **attach** command to attach a filter-list to a filter.

Syntax: attach *list-name filter#*

list-name

Specifies the name of the filter-list. The **list** command can be used to display a list of the configured filter-list names.

Valid Values: Any alphanumeric string up to 16 characters

Default Value: None

filter#

Specifies the number of the filter. A numbered list of configured filters can be obtained using the list command.

Example: attach test_list 1

Create

Use the **create** command to create a filter-list or filter.

Syntax: create   list ...
                             filter ...

list *list-name*

Creates a list with the specified name.

Valid Values: Any alphanumeric string up to 16 characters

Default Value: none

You can also enter the **create list** command with no list name. You will then be prompted for the list name.

Example: `create list example_list`

filter *direction interface_number*

Creates a filter for the specified direction on the specified interface. Specify *input* to filter packets received on the specified interface. Specify *output* to filter packets to be sent by the specified interface.

A number is automatically assigned to a filter when it is created and from that point on is used to identify the filter, rather than having to key in the interface and direction (input or output) for all subsequent commands.

Example: `create filter input 1`

Default

Use the **default** command to set the default action for a filter. The default action is taken when no match is found for any of the filter items.

Syntax: default   *action filter#*
Example: `default exclude 1`

action

Specifies the default action. **Include** specifies that when no match is found to any of the filter items, the packet is processed. **Exclude** indicates that when no match is found, the packet is dropped.

filter#

Specifies the number of the filter. Use the **list** command to display a numbered list of configured filters.

Delete

Use the **delete** command to delete a filter-list or filter.

Syntax: delete   list ...
                            filter ...

list *list-name*

Deletes the specified list. The list command can be used to display the configured filter list names.

Example: `delete list example_list`

filter *filter#*

Deletes the specified filter. The list command can be used to display a numbered list of configured filters.

Example: `delete filter 1`

Detach

Use the **detach** command to detach a filter-list from a filter.

Syntax: detach *list-name filter#*

list-name

Specifies the name of the filter-list. The list command can be used to display a list of the configured filter names.

Valid Values: Any alphanumeric string up to 16 characters

Default Value: None

filter#

Specifies the number of the filter. The list command can be used to display a numbered list of configured filters.

Example: detach test_list 1

Disable

Use the **disable** command to disable filtering globally or for a specified filter.

Syntax: disable all
filter ...

all

Disables all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: disable all

filter *filter#*

Disables the specified filter. Use the list command to display a numbered list of configured filters.

Example: disable filter 1

Enable

Use the **enable** command to enable filtering globally or for a specified filter.

Syntax: enable all
filter ...

all

Enables all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: enable all

filter *filter#*

Enables the specified filter. Use the list command to display a numbered list of configured filters

Example: enable filter 1

List

Use the **list** command to globally display the state of the current filtering type, or to display information about a specific filter.

Syntax: list all
filter ...

all

Lists information about the state of all filters of the current type.

Example: list all

```
IPX IPX-List Config>list all
Filtering: ENABLED

Filter Lists:
Name                           Action
------------------------------ ----------
ipx01                          EXCLUDE
ipx02                          INCLUDE
ipx03                          EXCLUDE

Filters:
Id   Default    State      Ifc  Direction  Cache
---- ---------- ---------- ---- ---------- -----
1    INCLUDE    ENABLED    0    INPUT      10
2    INCLUDE    ENABLED    0    OUTPUT     10
3    INCLUDE    DISABLED   1    INPUT      10
4    INCLUDE    DISABLED   1    OUTPUT     10
```

filter *filter#*

Lists information about the specified filter. Use the list command to display a numbered list of configured filters.

Example: list filter 1

```
Filter:
Id   Default    State      Ifc  Direction  Cache
---- ---------- ---------- ---- ---------- -----
1    INCLUDE    ENABLED    0    INPUT      10

Filter Lists:
Name                           Action
------------------------------ ----------
ipx01                          EXCLUDE
ipx02                          INCLUDE
```

Move

Use the **move** command to change the order of filter lists within a filter. Packets are evaluated against the filter lists in the order the lists occur. The first match stops the filtering process.

Syntax: move *src-list-name dst-list-name filter#*

src-list-name

Specifies the list to be moved within the filter.

dst-list-name

Specifies the list before which the src-list-name will be moved.

filter#

Specifies the filter to which the lists belong. The list command can be used to display a list of the configured filters and their attached filter lists.

Example: `move test-list-1 test-list-2 2`

Set-cache

Use the **set-cache** command to set the size of the filter cache. A filter cache is only supported for the IPX interface filter; the ROUTER, RIP and SAP interface filters do not support a cache.

Syntax: set-cache *size filter#*
Example: `set-cache 10 1`

size
Specifies the size of the filter cache (in number of entries).

Valid Values: 4 to 64 cache entries.

Default Value: 10 entries.

filter#
Specifies the number of the filter. The list command can be used to display a numbered list of configured filters.

Example: `set-cache 10 1`

Update

The **update** command accesses the `IPX` *`type`*`-List` *`list-name`* `Config>` prompt. From this prompt you can issue commands to add, delete, or move items within the list being updated. From this prompt you can also set the action for the filter-list being updated.

Syntax: update *list-name*

list-name
Specifies the name of the filter-list. The list command can be used to display the configured filter-list names.

Example: update test-list

Add (Update subcommand)

Use the **add** subcommand to add items to a filter-list. The list item parameters vary based on the type of interface filter (ROUTER, RIP, SAP, or IPX) being configured. For all types of interface filter, the **add** command can be entered without parameters. You will then be prompted for the required parameters.

Add (ROUTER)

Syntax: add *address mask*

address
Specifies the value to be compared against the source node address of the router which sent the RIP response packet (after being ANDed with the mask). If you want to match on a single address, set the address parameter to the address and set the mask to FFFF FFFF FFFF. If you want to match on all addresses, set the address parameter and the mask parameter to 0000 0000 0000.

Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Default Value: none

mask

Specifies the value to be ANDed with the source node address of the router which sent the RIP response packet (before being compared with the address parameter).

If you want to match on a single address, set the address parameter to the address and set the mask to FFFF FFFF FFFF. If you want to match on all addresses, set the address parameter and the mask parameter to 0000 0000 0000.

Valid Values: X'0000 0000 0000' to X'FFFF FFFF FFFF'

Default Value: X'FFFF FFFF FFFF'

Example: `add 400000001000 ffffffff0000`

Add (RIP)

Syntax: add *net-range-start net-range-end*

net-range-start

Specifies the start of a range (inclusive) of IPX network numbers to be filtered. If you want to match on a single network number, set the net-range-start and net-range-end parameters to that network number. If you want to match on all network numbers, set the net-range-start to 0000 0001 and the net-range-end to FFFF FFFE.

Valid Values: X'1' to X'FFFFFFFE'

Default Value: X'1'

net-range-end

Specifies the end of a range (inclusive) of IPX network numbers to be filtered.

Valid Values: X'1' to X'FFFFFFFE'

Default Value: X'1'

Example: `add 00000001 FFFFFFFE`

Add (SAP)

Syntax: add *comparitor hops sap-type name*

comparitor

Specifies the type of hop count comparitor for this list item.

Valid Values:

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

Default Value: <= The comparitor and hops parameters are ignored on output filters.

hops

Specifies the hop count for this list item. If you do not want to filter based on hop count, enter <= 16 for the comparitor and hop count. The comparitor and hops parameters are ignored on output filters.

Valid Values: 0 to 16

Default Value: 16

sap-type

Specifies the service type to be filtered. Enter the service type, or 0000 for all service types.

Valid Values: X'0' to X' FFFF'

Default Value: 4

name

Specifies the service name to be filtered.

Valid Values:

A string of 1 to 47 ASCII characters (X'20' through X'7E'), with the exception of the following special characters: plus (+), minus (-), comma (,), semicolon (;), colon (:), slash (/), and back slash (\).

The question mark (?) and asterisk (*) characters serve as wildcard characters. The question mark may be used multiple times to represent any single character within the server name. The asterisk may be used multiple times to represent any portion of the server name. The question mark and asterisk may also be used together.

Default Value: none

Example: `add < 6 0004 *`

Add (IPX)

Syntax: add *comparitor hops ipx-type dst-net-range-start dst-net-range-end*
dst-address dst-mask dst-sck-range-start dst-sck-range-end
src-net-range-start src-net-range-end src-address src-mask
src-sck-range-start src-sck-range-end

comparitor

Specifies the type of hop count comparitor for this list item. The comparitor and hops parameters are ignored on output filters.

Valid Values:

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

Default Value: <=

hops

Specifies the hop count for this list item. If you do not want to filter based on hop count, enter <= 16 for the comparitor and hop count. The comparitor and hops parameters are ignored on output filters.

ipx-type

Specifies the IPX packet type to be filtered. Enter the packet type, or 00 for all packet types.

Valid Values: X'0' - X'FF'

Default Value: X'0'

dst-net-range-start

Specifies the start of a range (inclusive) of destination IPX network numbers to be filtered. If you want to match on a single network number, set the dst-net-range-start and dst-net-range-end parameters to that network number. If you want to match on all network numbers, set the dst-net-range-start to 0000 0001 and the dst-net-range-end to FFFF FFFE.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Default Value: X'0000 0000'

dst-net-range-end

Specifies the end of a range (inclusive) of destination IPX network numbers to be filtered. If you want to match on a single network number, set the dst-net-range-start and dst-net-range-end parameters to that network number. If you want to match on all network numbers, set the dst-net-range-start to 0000 0001 and the dst-net-range-end to FFFF FFFE.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Default Value: X'0000 0000'

dst-address

Specifies the value to be compared against the destination node address (after being ANDed with the dst-mask). If you want to match on a single address, set the dst-address parameter to the address and set the dst-mask to FFFF FFFF FFFF. If you want to match on all addresses, set the dst-address parameter and the dst-mask parameter to 0000 0000 0000.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Default Value: X'0000 0000'

dst-mask

Specifies the value to be ANDed with the destination node address (before being compared with the dst-address parameter). If you want to match on a single address, set the dst-address parameter to the address and set the dst-mask to FFFF FFFF FFFF. If you want to match on all addresses, set the dst-address parameter and the dst-mask parameter to 0000 0000 0000.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Default Value: X'0000 0000'

dst-sck-range-start

Specifies the start of a range (inclusive) of destination IPX sockets to be filtered. If you want to match on a single socket, set the dst-sck-range-start and dst-sck-range-end parameters to that socket. If you want to match on all sockets, set the dst-sck-range-start to 0000 and the dst-sck-range-end to FFFFF.

Valid Values: 0 - 255

Default Value: 0

dst-sck-range-end

Specifies the end of a range (inclusive) of destination IPX sockets to be filtered. If you want to match on a single socket, set the dst-sck-range-start and dst-sck-range-end parameters to that socket. If you want to match on all sockets, set the dst-sck-range-start to 0000 and the dst-sck-range-end to FFFFF.

Valid Values: 0 - 255

Default Value: 0

src-net-range-start

Specifies the start of a range (inclusive) of source IPX network numbers to be filtered. If you want to match on a single network number, set the src-net-range-start and src-net-range-end parameters to that network number. If you want to match on all network numbers, set the src-net-range-start to 0000 0001 and the src-net-range-end to FFFF FFFE.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Default Value: X'0000 0000'

src-net-range-end

Specifies the end of a range (inclusive) of source IPX network numbers to be filtered. If you want to match on a single network number, set the src-net-range-start and src-net-range-end parameters to that network number. If you want to match on all network numbers, set the src-net-range-start to 0000 0001 and the src-net-range-end to FFFF FFFE.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Default Value: X'0000 0000'

src-address

Specifies the value to be compared against the source node address (after being ANDed with the src-mask). If you want to match on a single address, set the src-address parameter to the address and set the src-mask to FFFF FFFF FFFF. If you want to match on all addresses, set the src-address parameter and the src-mask parameter to 0000 0000 0000.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Default Value: X'0000 0000'

src-mask

Specifies the value to be ANDed with the source node address (before being compared with the src-address parameter). If you want to match on a single address, set the src-address parameter to the address and set the src-mask to FFFF FFFF FFFF. If you want to match on all addresses, set the src-address parameter and the src-mask parameter to 0000 0000 0000.

Valid Values: X'0000 0000' to X'FFFF FFFF'

Default Value: X'0000 0000'

src-sck-range-start

Specifies the start of a range (inclusive) of source IPX sockets to be filtered. If you want to match on a single socket, set the src-sck-range-start and src-sck-range-end parameters to that socket. If you want to match on all sockets, set the src-sck-range-start to 0000 and the src-sck-range-end to FFFFF.

Valid Values: X'0000' to X'FFFF'

Default Value: X'0000'

src-sck-range-end

Specifies the end of a range (inclusive) of source IPX sockets to be filtered. If you want to match on a single socket, set the src-sck-range-start and src-sck-range-end parameters to that socket. If you want to match on all sockets, set the src-sck-range-start to 0000 and the src-sck-range-end to FFFFF.

Valid Values: X'0000' to X'FFFF'

Default Value: X'0000'

Example:

```
 add  <= 16  0  00000004 00000004 000000000000 000000000000
0000 FFFF 0000005A 0000006A 000000000000 000000000000 0000 FFFF
```

This example filters all packets from IPX networks 5A through 6A to IPX network 4.

Delete (Update subcommand)

Use the **delete** subcommand to delete an item from the current filter-list.

Syntax: delete *item#*

Example: `delete 4`

item#

Specifies the number of the item in the list. The number can be obtained by using the list command to list the items in the filter-list.

List (Update subcommand)

Use the **list** subcommand to display the filter-list action and list filter items.

Syntax: list

Example: `list`

```
IPX IPX-List 'ipx01' Config>list
Action: EXCLUDE
Id   Hops Type Net Range           Address      Mask         Sock Range
---- ---- ---- ------------------- ------------ ------------ -----------
1    <=16    0     4320 -     4324 4000003A0002 FFFFFFFFFFFF    0 - FFFF (Dest)
                   3A33 -    13A33 400000010000 FFFFFFFF0000    0 - FFFF (Source)
```

Move (Update subcommand)

Use the **move** subcommand change the order of filter items. After you change the order of filter items, they are renumbered to reflect the new order. The list command can be used to display a numbered list of configured filter items.

The *src-line#* parameter indicates the line to be moved. This line will be moved to precede the item specified by the *dest-line#* parameter.

Syntax: move *src-line#* *dest-line#*

Example: `move 5 2`

Set-action (Update subcommand)

Use the **set-action** subcommand to indicate the action to be taken when a match is made to a filter-list.

Syntax: set-action include
exclude

include
Specifies that if a match is found for the current filter, the packet will be processed (included) for ROUTER and IPX filters. For RIP and SAP filters, **include** specifies that the RIP or SAP entry will be processed.

Example: `set-action include`

exclude
Specifies that if a match is found for the current filter, the packet will be dropped (excluded) for ROUTER and IPX filters. For RIP and SAP filters, **exclude** specifies that if a match is found, the RIP or SAP entry will be ignored.

Example: `set-action exclude`

Exit

Use the **exit** subcommand to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 19. Monitoring IPX

This chapter describes how to monitor IPX protocol activity and use the IPX console commands. It includes the following sections:

- "Accessing the IPX Console Environment"
- "IPX Console Commands"

Accessing the IPX Console Environment

For information on how to access the IPX console environment, refer to "Getting Started (Introduction to the User Interface)" in the *Software User's Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

IPX Console Commands

Table 19-1 lists the IPX console commands. The IPX console commands allow you to view the parameters and statistics of the interfaces and networks that transmit IPX packets. Console commands display configuration values for the physical, frame, and packet levels. You also have the option of viewing the values for all three protocol levels at once.

Enter the IPX console commands at the `IPX>` prompt. Table 19-1 summarizes the IPX monitoring commands.

Table 19-1 (Page 1 of 2). IPX Console Command Summary

Command	Function
? (Help)	Lists all the IPX console commands or lists the options associated with specific commands.
Access-controls	Displays whether the global IPX filter (access control) is enabled, the IPX access-control statements, and the number of packets that have matched each access-control statement.
Cache	Lists the current contents of the routing cache.
Config	Lists the IPXWAN router name and node-ID. Lists the number, name and type of each interface on which IPX is enabled, as well as the IPX network number and host number, the frame encapsulation type, whether RIP/SAP Broadcast Pacing is enabled on those interfaces, and the RIP and SAP update intervals on those interfaces.
Counters	Displays the number of routing errors and packet overflows.
Delete	Deletes a Keepalive filtering table entry.
Disable	Disables specific IPX interfaces or globally disables IPX.
Dump routing tables	Displays the contents of the IPX RIP table.
Enable	Enables IPX on specific interfaces or globally enables IPX.
Filters	Displays whether global SAP filtering is enabled, the SAP filter statements, and a count of the SAP advertisements which have been filtered.

Table 19-1 (Page 2 of 2). IPX Console Command Summary

Command	Function
Filter-Lists	Accesses the `IPX` *type*`-Lists>` prompt. This is the environment where information regarding the Interface filters (ROUTER, RIP, SAP, and IPX) can be displayed.
IPXWAN	Lists IPXWAN configuration information for each serial interface on which IPXWAN is enabled.
Keepalive	Shows the status of Keepalive filtering on each interface and displays the status of each active client/server connection.
Ping	Sends IPXPING packets to another host once a second and watches for a response. This command can be used to isolate trouble in an internetwork environment.
Sizes	Displays the configured sizes of the local node and remote network caches, and the number of cache entries currently in use.
Slist	Displays the contents of the IPX SAP server table.
Exit	Exits the IPX console process and returns to the GWCON environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
access controls
cache
config
counters
delete
disable
dump routing tables
enable
exit
filters
filter-lists
ipxwan
keepalive
ping
sizes
slist
```

Access Controls

Use the **access-controls** command to list the status of global IPX filters (access controls), the IPX access control statements, and a count of how many times each control statement has been followed.

Syntax: access-controls

Example: **access-controls**

```
Access Control currently enabled
List of Access Control records:
# T   Dest Net  Host          Sck  Sck  Src Net  Host         Sck Sck  Count
1 E        179  123456789ABC 1234 1234      176  000000000000   0   0    3
2 I          0  000000000000    0  FFF        0  000000000000 426 426    0
```

Type Identifies whether packets are sent or dropped for a specific address or set of addresses. I means include. This allows the packets to be sent. E means exclude. This causes the router to discard the packets.

Dest-net Network number of the destination. Zero (0) means all networks.

Dest-host Host number on the destination network (0) means all hosts on the network.

Des-sck Two numbers that specify an inclusive range of destination sockets.

Src-net Network number of the source. Zero (0) means all networks.

Src-host Host number on the source network. Zero means all hosts on the network.

Src-sck Two numbers that specify an inclusive range of source sockets.

Count Specifies the number of incoming IPX packets that have matched each access-control statement, causing the associated Type (Include or Exclude) to be performed.

Cache

Use the **cache** command to display the contents of the IPX routing cache.

Syntax: cache

Example: **cache**

```
Dest Net/Node        Use Count    via Net/Node        via Int
152                   56000       161/000000000006     PPP/0
162                   56476       162/000000000000     Eth/0
162/0000C0239F71      56476       162/0000C0239F71     Eth/0
```

The first entry shows that the remote network 152 can be reached over the serial interface with IPX network number 161. The second entry is the IPX network 162. It is an Ethernet directly attached to the router. This entry is a general local network entry. There will be one general local network entry for each of the directly attached networks after they have begun forwarding IPX packets. The last entry is a local entry on an Ethernet. This IPX cache entry has been used to send 56,476 packets to the IPX node number 0000C0239F71 on net number 162.

Config

Use the **config** command to list the number, name, and type of each network interface on which IPX is enabled, as well as listing the IPX network number and host number, the frame encapsulation type and the RIP and SAP update intervals on those interfaces. The **config** command also displays whether RIP/SAP Broadcast Pacing is enabled on specific interfaces, and the IPXWAN router name and node-ID.

Syntax: config

Example: **config**

```
             Router Configuration
             IPX Name:  Node ID: 0
             Net   Name   Type                     Network/Address
             0     Eth/0  Ethernet/802.3           162/000093908468
             1     PPP/0  8 port X.21 Adapter      161/000000000001
             2     PPP/1  8 port X.21 Adapter      160/000000000001

             IPX Encapsulation/Frame Types
             Net   Name   Type                     Encapsulation
             0     Eth/0  Ethernet/802.3           ETHERNET_802.3
             1     PPP/0  8 port X.21 Adapter      161/000000000001
             2     PPP/1  8 port X.21 Adapter      160/000000000001

RIP/SAP Timer Intervals and Pacing
Net  Name   Type                            SAP Interval  RIP Interval Pacing
0    Eth/0  Ethernet/802.3                        1             1      Disabled
1    PPP/0  8 port X.21 Adapter                   1             1      Disabled
2    PPP/1  8 port X.21 Adapter                   5             5      Enabled
```

Router Configuration The current router configuration information.

IPX Name The IPXWAN router name.

Node ID The IPXWAN node-id (primary network number).

Net The interface number.

Name The interface name.

Type The hardware type of the interface.

Network/Address The user-assigned network number and host number. Except for serial interfaces, the host number is the node address of the network interface. For serial interfaces, it is the user-configured IPX host number.

IPX Encapsulation/Frame Types The encapsulation type for each interface on which IPX is enabled.

Net The interface number.

Name The interface name.

Type The hardware type of the interface.

Encapsulation The encapsulation type configured for the interface.

RIP/SAP Timer Intervals and Pacing The delay between the transmission of complete RIP and SAP advertisements on an interface, and whether RIP/SAP Broadcast Pacing is enabled on an interface.

Net The interface number.

Name The interface name.

Type The hardware type of the interface.

SAP Interval The number of minutes between complete SAP advertisements on the interface. The range is 1 through 1440. The default is 1.

RIP Interval The number of minutes between complete RIP advertisements on the interface. The range is from 1 to 1440. The default is 1.

Pacing Indicates whether RIP/SAP Broadcast Pacing is enabled.

Counters

Use the **counters** command to display the number of routing errors and packet overflows that have occurred. In the example, the counters show no recorded errors.

Syntax: counters

Example: **counters**

```
Routing errors
Count     Type
   0      Unknown
   0      Checksum error
   0      Destination unreachable
   0      Hop count expired
   0      Interface size exceeded


Destination errors
Count      Type
   0       Unknown
   0       Checksum error
   0       Nonexistent socket
   0       Congestion


IPX input packet overflows
    Net     Count
   Eth/0    0
   FR/0     0
   PPP/0    0
```

Routing Errors

Unknown An unspecified error occurred before reaching the destination.

Checksum The checksum is incorrect, or the packet had some other serious inconsistency before reaching the destination.

Destination unreachable The destination host cannot be reached from here.

Hop count expired The packet has passed through 15 internet routers without reaching its destination.

Interface size exceeded The packet is too large to be forwarded through some intermediate network.

Destination errors

Unknown An unspecified error was detected at destination.

Checksum The checksum is incorrect, or the packet has some other serious inconsistency detected at destination.

Nonexistent socket The specified socket does not exist at the specified destination host.

Congestion The destination cannot accept the packet due to resource limitations.

IPX Input Packet Overflows

Net Specifies the interface name.

Count Specifies the number of packets that could not be received due to resource limitations.

Delete

Use the **delete** command to remove a Keepalive filtering table entry.

Syntax: delete *entry#*

entry#

Specifies the table entry to be deleted. The **keepalive** command can be used to list the contents of the Keepalive filtering table.

Example: **`delete 1`**

Disable

Use the **disable** command to disable IPX on specific interfaces, or to disable IPX globally on all interfaces.

Syntax: disable interface ...
ipx

interface *interface#*

Disables IPX on the specified interface. IPX can be re-enabled on the interface using the **enable** command.

Example: **`disable interface 0`**

ipx

Disables IPX globally on all interfaces. IPX can be globally re-enabled using the **enable** command.

Example: **`disable ipx`**

Dump

Use the **dump** command to display the contents of the current IPX RIP routing tables.

Syntax: dump

Example: **`dump`**

The screen displays the following information:

```
 11 route entries used out of 32
 11 net entries used out of 32

Type  Dest net  Hops  Delay  Age(M:S)   via Router
 Dir      124     0      1       0: 0     124/AA0004001A04
 Dir      131     0      1       0: 0     131/00000000001A
 Dir      177     0      1       0: 0     177/00000000001A
 Dir       41     0      1       0: 0      41/4000C90401FA
Dir      249     0      1       0: 0     249/0000C9084F34
RIP      250     1      2       0:10     249/0000C9093250
RIP 2C39ABE9     2      3       0:10     249/0000C9093250
RIP       BB     1      2       0:50      41/4000C9050971
RIP        1     2      3       0:50      41/4000C9050971
RIP       31     2      3       0:50      41/4000C9050971
RIP      703     1      2       0:20      41/4000C9041243
```

Type
Specifies one of the following:

Dir - specifies that this network is directly connected to the router.

RIP - specifies that this route was provided by the IPX routing protocol, RIP.

Old - specifies that this route has timed out and is no longer being used. The route remains in the table briefly to inform other routers that the route is no longer valid; after this brief interval, it is no longer displayed.

Dest net
Specifies the destination network number.

Hops
Specifies the number of router hops to this destination.

Delay
Specifies the estimate of how long it takes the router to transmit and for the packet to arrive at its destination. The unit of delay is the number of IBM PC clock ticks to send a 576-byte packet, which is 18.21 clock ticks per second. The minimum delay is 1 unit.

Age
Specifies the age of the routing information in minutes and seconds. If an entry in the routing table is not updated, the router takes the following actions:

- After three RIP update intervals have passed, the route is specified as Old and the router advertises that the route is no longer valid. The RIP update interval can be displayed using the IPX **config** command. For additional information on RIP intervals, see “Specifying RIP Update Interval” on page 18-3.
- After an additional 60 seconds, the route is deleted and does not appear in the dump display.

Via router
Specifies the next hop for packets going to networks that are not directly connected. For directly connected networks, this is the address of the router interface that transmits the packet.

At the top of the display is the number of route and network entries used and the total available. If all the network entries are used, it is likely that the routing table is not large enough. Use the IPX configuration **set maximum networks** command to increase the size.

If all of the route entries are used, then there may be routes to IPX networks that cannot be kept, including new, incoming networks. If you do not want to increase the number of available routes, reduce the number of maximum routes per network.

Enable

Use the **enable** command to enable IPX on specific interfaces or to globally enable IPX on all interfaces on which IPX has been configured and enabled.

Syntax: enable interface ...
ipx

interface *interface#*

Enables IPX on the specified interface. An IPX network number must have been configured for the interface before IPX can be enabled. For serial interfaces, a host-number must have been configured before IPX can be enabled.

Example: **`enable interface 0`**

ipx

Enables IPX on all interfaces on which IPX has been configured and enabled.

Example: **`enable ipx`**

Filters

Use the **filters** command to display whether global SAP filtering is enabled, the SAP filter statements, and a count of the SAP advertisements that have been filtered.

Syntax: filters

Example: **`filters`**

```
IPX SAP Filter currently enabled
List of IPX SAP Filter records:
Count  Max Hops  Type  Service Name
  0          8     4  ?
  0          1  1234  SomeServer
```

Count	Indicates the number of SAP advertisements that have been filtered (discarded).
Max Hops	Indicates the maximum number of hops permitted for the service.
Type	Is the numeric service class.
Service name	Is the name of the service if it has a name.

Filter-lists

Use the **filter-lists** command to access the `IPX` *`type`*`-Lists>` prompt. Valid types are: router, rip, sap, and ipx.

For information about the commands available from this prompt, see "IPX Interface Filter Monitoring Commands" on page 19-14.

Syntax: filter-lists router
rip
sap
ipx

Example: **`filter-lists router`**

IPXWAN

Use the **ipxwan** command to list the current configuration information for serial interfaces on which IPXWAN is enabled.

Syntax: ipxwan detailed . . .
summary

detailed *interface#*

Lists the complete current configuration information for the specified serial interface on which IPXWAN is enabled.

Example: **ipxwan detailed**

```
Network number [0]? 2
Detailed information for IPXWAN link over interface 2, PPP/1
This side is the IPXWAN slave
Neighbor Name: SKYSURF2
Neighbor Node ID: 727299
Negotiated Routing Type: RIP/SAP
Link Delay: 330 1/18th sec ticks
Common Net#: 132
Connection Timeouts: 0
Connection Retries: 0
Timer Requests Sent: 1
Timer Requests Received: 1
Timer Responses Sent: 1
Timer Responses Received: 0
Info Requests Sent: 0
Info Requests Received: 1
Info Responses Sent: 1
Info Responses Received: 0
```

Network number The network interface number.

Neighbor Name The router name of the neighbor as received in the RIP/SAP Information Request Packet.

Neighbor Node ID The node ID (also known as the primary network number) of the neighbor. This is a IPX network number unique to the entire internetwork. It is a 32-bit quantity.

Negotiated Routing Type The negotiated routing type. Currently supported is RIP/SAP. The default is RIP/SAP.

Link Delay The link delay in 1/18th second ticks calculated by the master. It is a 16-bit quantity. It is always calculated, therefore there is no default.

Common Net# The network number agreed upon by both ends of the link. This number must be unique to the entire internetwork. It is a 32-bit quantity. There is no default, it must be negotiated.

Connection Timeouts The number of times the connection timed out. A connection will timeout periodically if the exchange of IPXWAN packets does not proceed. You can configure the timeout period using the **set ipxwan** command. The default for the timeout period is 60 seconds.

Connection Retries The number of times the connection is retried after timing out. The amount of time to wait (before retrying) is configurable by using the **set ipxwan** command. It defaults to 60 seconds.

Timer Requests Sent The number of IPXWAN Timer Request packets sent.

Timer Requests Received The number of IPXWAN Timer Request packets received.

Timer Responses Sent The number of IPXWAN Timer Response packets sent.

Timer Responses Received The number of IPXWAN Timer Response packets received.

Info Requests Sent The number of IPXWAN Information Request packets sent.

Info Requests Received The number of IPXWAN Information Request packets received.

Info Responses Sent The number of IPXWAN Information Response packets sent.

Info Responses Received The number of IPXWAN Information Response packets received.

summary

Lists a summary of the current configuration information for each serial interface on which IPXWAN is enabled. Summary information is displayed only for links that have successfully completed the IPXWAN protocol exchange.

Example: **ipxwan summary**

```
Net  Name   Common Net#  NodeID  Neighbor Name

2    PPP/1  132          727299  SKYSURF2
```

Net Network interface number.

Name Network interface name.

Common Net# Network number agreed upon by both ends of the link. This number must be unique to the entire internetwork.

NodeID Node ID (also known as the primary network number) of the neighbor. This is a IPX network number unique to the entire internetwork.

Neighbor Name Router name of the neighbor as received in the RIP/SAP Information Request Packet.

Keepalive

Shows the status of Keepalive filtering on each interface and displays the status of each active client/server connection.

Syntax: keepalive

Example: **keepalive**

```
 Net #     Keepalive Status
 -----     ----------------
     0     DISABLED
     1     ENABLED
     2     DISABLED
     3     DISABLED
     4     DISABLED

 Keepalive
 Conn #     Net  /   Node     /Sock        Net  /   Node     /Sock
-----------------------------------------------------------------
     0    272727/000000000001/4001 <->      302/0000C911EF1C/4004
       (server conn # 1, conn type: passive, last heard 1:00 ago)
     1    272727/000000000001/4001 <->      302/0000C911B0D9/4004
       (server conn # 2, conn type: passive, last heard 1:00 ago)
```

Ping

Use the **ping** command to make the router send IPXPING packets to a given destination once a second ("pinging") and watch for a response. This command can be used to isolate trouble in an internetwork environment.

This process is done continuously. Matching received responses are displayed with the sender's IPX network number and node number, the number of hops, and the round trip time in milliseconds.

To stop the pinging process, type any character at the console. At that time, a summary of packet loss, round trip time, and number of unreachable destinations will be displayed.

When a multicast address is given as destination, there may be multiple responses for each packet sent, one for each group member. Each returned response is displayed with the source address of the responder.

Note: Care should be taken when specifying the broadcast address (FFFF FFFF FFFF) as this could generate a large number of IPXPING response packets, which would degrade network and routing software performance.

Each packet contains 56 bytes of data that include a 4-byte time stamp (to determine the round-trip time) and 52 bytes of sequential numbers, which are checked when receiving IPXPING response packets.

Syntax: ping *ipx-net# ipx-node#*

ipx-net#
: Specifies the destination IPX network number. Valid values are in the range 0000 0001 - FFFF FFFE.

ipx-node#
: Specifies the destination IPX node number. Valid values are in the range 0000 0000 0001 - FFFF FFFF FFFF

Example: **`ping 00000004 40000000409A`**

```
IPXPING 00000004/40000000409A: 56 data bytes
56 data bytes from 00000004/40000000409A: hops=2, time=220 ms
56 data bytes from 00000004/40000000409A: hops=2, time=200 ms
56 data bytes from 00000004/40000000409A: hops=2, time=210 ms

3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms) min/ave/max = 200/210/220
```

Sizes

Use the **sizes** command to display the configured sizes of the local node and remote network caches, and the number of cache entries currently in use. (This command does not display the contents of the caches.)

Syntax: si̲zes

Example: **`sizes`**

```
Current IPX cache size:
Remote network cache size (max entries): 64
       2 entries now in use

Local node cache size (max entries): 128
       1 entries now in use
```

Slist

Use the **slist** command to display the contents of the IPX SAP server table.

Syntax: slist

Example: **slist**

```
State Typ Service Name Hops Age(M:S)   Net/   Host     /Sock
 SAP 0004 PCS12        3   0:50        1/000000000048/0451
 SAP 0004 ACMPCS       3   0:50        1/00000000004A/0451
 SAP 0004 DEVEL2       1   0:50       11/0000000000B4/0451
 SAP 0004 PLANNING     2   0:50       BB/0000000000B7/0451
 SAP 0004 DEVEL        2   0:50       BB/0000000000EE/0451
 SAP 0004 SOFT2        1   0:30      704/000000000094/0451
 SAP 0004 SKYSURF1     2   0: 5  2C39ABE9/000000000001/0451
 SAP 0278 DIRTREE      2   0: 5  2C39ABE9/000000000001/4005
 SAP 026B DIRTREE      2   0: 5  2C39ABE9/000000000001/0045

9 services used out of 32
```

State Specifies one of the following parameters:

SAP - indicates that this service was obtained by the SAP routing protocol.

Old - indicates that this service has timed out and is no longer being used. The service is kept briefly in the table to inform other routers that the service is no longer valid. After that, it is deleted and is no longer displayed.

Typ Specifies the server type in hexadecimal. File servers are type 0004. Other type numbers are assigned by Novell.

Service name Specifies the server's unique name for this type of server. Only the first 30 characters of the 47-character name are displayed to conserve space.

Hops Specifies the number of router hops from this router to the server.

Age Specifies the age of the service information. If an entry in the SAP table is not updated, the router takes the following actions:

After 3 SAP update intervals have passed, the service is specified as Old and the router advertises that the service is no longer valid. The SAP update interval can be displayed using the IPX **config** command.

After an additional 60 seconds, the service is deleted and does not appear in the **slist** display.

Net/Host/Sock Specifies the address of the service. The address includes the following parameters:

Network number

Net host number (the address of the first interface on the network)

Socket number at which the service can be reached

At the bottom of the display is the number of entries used and the total available. If all the entries are used, it is likely that the service table is not large enough. Use the IPX configuration **set maximum services** command to increase the size.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

IPX Interface Filter Monitoring Commands

Table 19-2 lists the commands available from the `IPX *type*-Lists>` prompt. Each of these commands is explained in detail in this section.

To access the `IPX *type*-Lists>` prompt, enter **`filter-lists`** *type* at the `IPX>` prompt. Valid types are router, rip, sap, and ipx.

Table 19-2. IPX Interface Filter Command Summary

Command	Function
Cache	Displays the contents of the filter cache for the specified interface. Only the IPX filter supports a filter cache.
Clear	Clears the counters of the specified filter, or clears the counters of all filters of the current type (ROUTER, RIP, SAP, or IPX).
Disable	Disables a specified filter, or all filters of the current type.
Enable	Enables a specified filter, or all filters of the current type.
List	Lists a specified filter, or all filters of the current type.
Exit	Returns to the previous prompt level.

Cache

Use the **cache** command to display the contents of the filter cache. Only the IPX filter supports a cache. ROUTER, RIP, and SAP filters do not support a filter cache.

Syntax: cache filter *filter#*

filter#
: Specifies the number of the filter. The list command can be used to display a numbered list of configured filters.

Example: **`cache filter  1`**

```
IPX IPX-Lists>cache filter 1
Hops Type Dst Net  Address      Sock Src Net  Address      Sock Action
---- ---- -------- ------------ ---- -------- ------------ ---- -------
   4   00 04000000 400003900000  802 03000040 400003004400  966 EXCLUDE
   2   00 0004A300 400000233D00  952 0763A020 4000000DD100  920 INCLUDE
```

Clear

Use the **clear** command to clear the counters of the specified filter, or to clear the counters of all filters of the current type (ROUTER, RIP, SAP, or IPX).

Syntax: clear all
filter ...

all
: Clears the counters of all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: **`clear all`**

filter *filter#*
: Clears the counters of the specified filter number. The list command can be used to display a numbered list of configured filters.

Example: `clear filter 1`

Disable

Use the **disable** command to disable specific filters or to disable all filters of the current type (ROUTER, RIP, SAP, or IPX).

Syntax: disable all
filter *filter#*

all
Disables all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: `disable all`

filter *filter#*
Disables the specified filter number. The list command can be used to display a numbered list of configured filters.

Example: `disable filter 1`

Enable

Use the **enable** command to enable specific filters or to enable all filters of the current type (ROUTER, RIP, SAP, or IPX).

Syntax: enable all
filter *filter#*

all
Enables all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: `enable all`

filter *filter#*
Enables the specified filter number. The list command can be used to display a numbered list of configured filters.

Example: `enable filter 1`

List

Use the **list** command to display information about specific filters, or about all filters of the current type (ROUTER, RIP, SAP, or IPX).

Syntax: list all
filter *filter#*

all
Lists the configuration of all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: `list all`

```
IPX IPX-Lists>list all
Filtering: ENABLED

Filter Lists:
Name                           Action
------------------------------ ----------
ipx01                          EXCLUDE
ipx02                          INCLUDE
ipx03                          EXCLUDE

Filters:
Id   Default    State      Ifc  Direction  Cache
---- ---------- ---------- ---- ---------- -----
1    INCLUDE    ENABLED    0    INPUT      10
2    INCLUDE    ENABLED    0    OUTPUT     10
3    INCLUDE    DISABLED   1    INPUT      10
4    INCLUDE    DISABLED   1    OUTPUT     10
```

filter *filter#*

Lists the configuration of the specified filter number. The list command can be used to display a numbered list of configured filters.

Example: `list filter 1`

```
IPX IPX-Lists>list filter 1

Filters:
Id   Default    State      Ifc  Direction  Cache
---- ---------- ---------- ---- ---------- -----
1    INCLUDE    ENABLED    0    INPUT      10

Filter Lists:
Name                           Action     Count
------------------------------ ---------- ----------
ipx01                          EXCLUDE    43
ipx02                          INCLUDE    23453
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 20. Using and Configuring ARP

This chapter describes how to use the Address Resolution Protocol (ARP) and the Inverse Address Resolution Protocol (Inverse ARP) on your router. It includes the following sections:

- “ARP Overview”
- “Inverse ARP Overview” on page 20-3
- “Classical IP and ARP Over ATM Overview (RFC 1577)” on page 20-4
- “IPX and ARP Over ATM Overview (RFC 1483)” on page 20-10
- “Accessing the ARP Configuration Environment” on page 20-10
- “ARP and Inverse ARP Configuration Commands” on page 20-11
- “ARP Over ATM Configuration Commands” on page 20-16

Note: If the device's software load does not contain Asynchronous Transfer Mode (ATM), ATM-related commands are not valid and are not displayed at the ARP configuration and console prompts.

ARP Overview

The ARP Protocol is a low-level protocol that dynamically maps network layer addresses to physical medium access control (MAC) addresses. Given only the network layer address of the destination system, ARP locates the MAC address of the destination host within the same network segment.

For example, a router receives an IP packet destined for a host connected to one of its LANs. The packet contains only a 32-bit IP destination address. To construct the data link layer header, a router acquires the physical MAC address of the destination host. Then, the router maps that address to the 32-bit IP address. This function is called *address resolution*. Figure 20-1 on page 20-2 illustrates how ARP works.

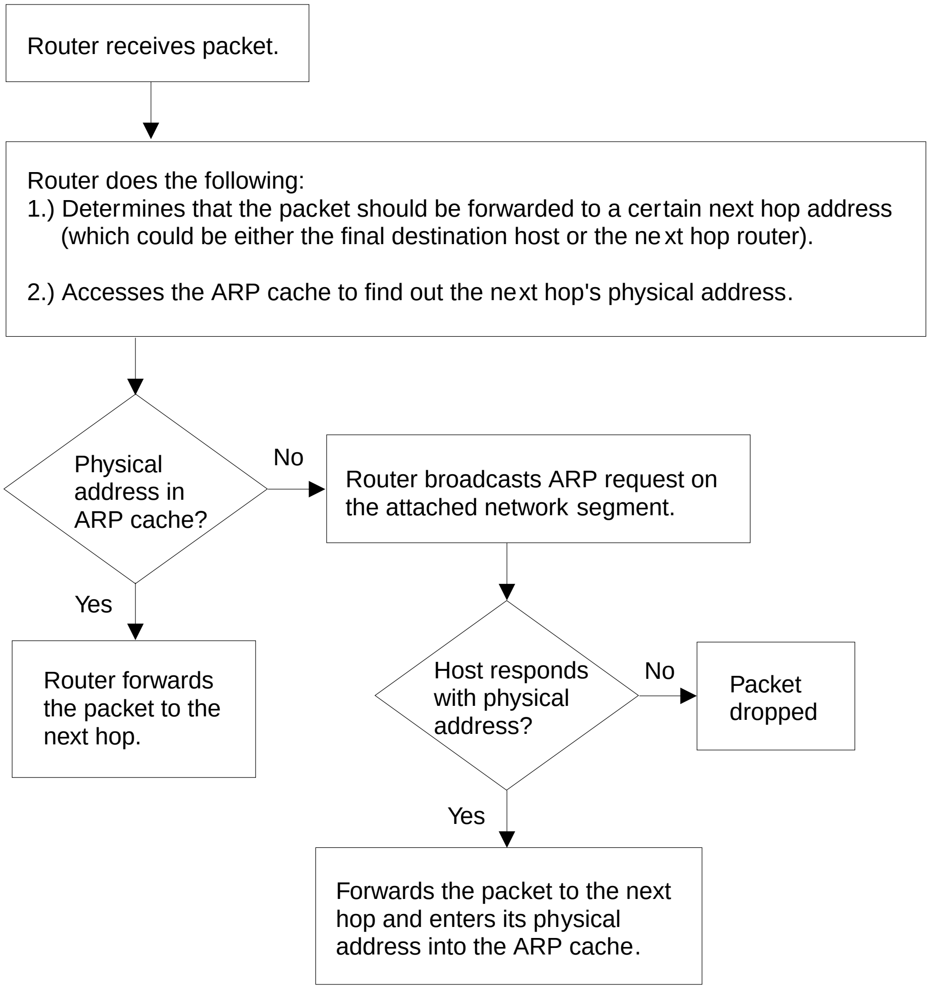

Router receives packet.
Router does the following:
1.) Determines that the packet should be forwarded to a certain next hop address (which could be either the final destination host or the next hop router).
2.) Accesses the ARP cache to find out the next hop's physical address.
Physical address in ARP cache?
No
Router broadcasts ARP request on the attached network segment.
Yes
Router forwards the packet to the next hop.
Host responds with physical address?
No
Packet dropped
Yes
Forwards the packet to the next hop and enters its physical address into the ARP cache.

Figure 20-1. ARP Address Resolution Broadcast

When a router translates a network layer address to a physical address, the router accesses the ARP (translation) cache. The ARP cache contains the physical MAC address that corresponds to that network layer address. If the address is missing, the router broadcasts an ARP request to all hosts on the attached network segment to locate the correct physical MAC address. The node with the correct physical MAC address responds to the router. The router then sends the packet to the node and enters the physical MAC address into the translation cache for future use.

RFC 1577, Classical IP & ARP over ATM, extends the ARP protocol with a different packet format and with the addition of an entity known as the ARP server as described in “Classical IP and ARP Over ATM Overview (RFC 1577)” on page 20-4.

Inverse ARP Overview

Inverse ARP, described in RFC 1293, was created for Frame Relay networks. This protocol defines a method for routers on a Frame Relay network to learn the protocol addresses of other routers in a way that very efficiently reduces traffic by eliminating the need to use broadcast ARP packets for address resolution. Inverse ARP discovers a protocol address by sending Inverse ARP request packets to the hardware address (for Frame Relay circuits the circuit identifier is the Frame Relay equivalent of a hardware address), address; for ATM, an ATM address is exchanged), as soon as the circuit becomes active. The remote router responds with its protocol address and the resulting mapping is stored in the ARP cache.

In ATM, the inverse ARP packet has been extended to handle the variable-sized ATM addresses of the source and destination. Addresses learned by inverse ARP are aged out in the same way as those learned by ARP.

The protocol address-to-hardware address entries learned by Inverse ARP do not time out when the ARP refresh timer expires. The mappings do not age at all except when the Frame Relay circuit goes down. This means that the router does not need to transmit any ARP broadcasts to update the ARP cache. However, the router permits updates to an entry when the other (remote) router changes its protocol address.

Support for both ARP and Inverse ARP greatly enhances the router’s interoperability with other vendors’ routers over Frame Relay for dynamic mapping of protocol and hardware addresses. If other Frame Relay-attached routers support Inverse ARP, then the mappings are dynamically learned as described above. If the attached routers do not support Inverse ARP but support “traditional” ARP on Frame Relay, then the mappings still could be learned dynamically using ARP exchanges (see Figure 20-1 on page 20-2).

If needed, you can manually configure the protocol addresses of other routers using the Frame Relay configuration command **add protocol-address**. For additional information, see the chapter on configuring Frame Relay interfaces in the Software User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1.

Classical IP and ARP Over ATM Overview (RFC 1577)

The Internet Engineering Task Force (IETF) has standardized its solution for sending IP traffic over an ATM interface in RFC 1577, “Classical IP & ARP over ATM.” This document, created by the IP over ATM working group of the IETF, strives to keep the ATM infrastructure transparent to IP. Most applications that run today in a LAN or WAN environment will see no difference in function; however, their performance and throughput gains may be substantial, as described in “Advantages of Classical IP.”

For additional information on Classical IP & ARP over ATM, and for illustrations showing logical and physical network configurations, refer to *Configuration Program User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

Classical IP (CIP) Logical IP Subnets (LIS)

In Classical IP (CIP), IP stations are grouped in Logical IP Subnets (LIS). Classical IP servers and clients are defined to support these subnets similar to the way that LAN Emulation servers and clients are defined to LAN Emulation Services as described in the “Using and Configuring LAN Emulation Services (LES)” chapter of *Software User’s Guide*.

For many configuration commands, you will be prompted to answer questions that are identical to those for LAN Emulation Clients and Servers. Questions that require ATM address ESIs and selectors, for example, will be asked in a similar manner whether you are configuring Classical IP or LAN Emulation.

Each of these configuration questions is based on the client definition. A client is defined as an interface number (ATM only) and an IP address.

In its simplest form, the IP client has no server and can talk only to those that contact its *automatically*-assigned ATM address. If PVCs have been assigned, then they will be operational.

For a more detailed description of ATM, refer to the “Using, Configuring, and Monitoring ATM” chapter in *Software User’s Guide*.

Advantages of Classical IP

Classical IP has several advantages over conventional IP:

- Higher line speeds provided by ATM
- More efficient use of available bandwidth

  Classical IP requires less framing bytes than, for example, LANs (which contain source and destination MAC addresses), so less of the bandwidth is used for overhead and more is used for data.
- No broadcast traffic required for resolution of ARP frames

  In a broadcast environment, ARP traffic can adversely affect all stations. In Classical IP, the ARP traffic affects only the ARP Server and the client requesting the information. All other stations on the subnet are unaffected by this traffic.
- Independent conversation channels

When IP is used over a shared medium such as token ring or Ethernet, frames transmitted between two stations preclude other stations on the same physical network from sending messages. This is true even when the traffic is nonbroadcast. In Classical IP, independent channels are established between hosts having the conversation. These channels can be established with traffic parameters that protect the conversation from being impacted by other conversations.

- Simpler method for adding, deleting, moving, or changing stations

  The same benefits of moves, adds, deletes, etc., described for LAN emulation over ATM also apply to the CIP Logical IP Subnet (LIS). Refer to “Using, Configuring, and Monitoring ATM” chapter in *Software User’s Guide*.

  Membership in a LIS is not based on physical location. Logically related stations may be grouped into the same LIS. The ease with which a client can register with the ARP Server makes additions and changes trivial. The deletion will occur naturally as the ARP Server ages its entries.

  While all members of a LIS must support the Classical IP model, the 2216 can easily route between CIP Logical IP Subnets (LIS) and emulated LAN subnets. Some equipment may be more adept at CIP while other equipment may be more adept at LAN emulation. The flexibility of the 2216 allows you to place that equipment where it is most effective.

Classical IP Components

The Logical IP Subnet contains all of the properties of a normal IP subnet whether it is Ethernet, Token-Ring, or Frame Relay. However, because ATM is a Non-Broadcast Multiple Access (NBMA) network, the existing broadcast method for resolving addresses cannot be performed. To solve the addressing problem, RFC 1577 describes a registration/request procedure and introduces the notion of an ARP Server and ARP clients.

One ARP Server is defined per LIS. The server maintains the translation of IP addresses to ATM addresses. It allows CIP Clients to register by receiving incoming VCCs and querying the client for the appropriate information. The ARP Server also responds to ATMARP requests for ATM addresses corresponding to IP addresses requested by the client. Finally, the ARP server manages and updates its tables through aging ARP entries and managing incoming VCCs.

The client is the entity that always places calls. A client, as it IMLs, will place a call to the ARP Server, and through the exchange of InATMARP requests and replies will register with the ARP Server. The call placed by the client to the server is called a control channel. When the client has traffic to transmit to another client on the LIS, the client sends an ARP request to the ARP Server with the target IP address. The server sends back either a reply (if the server has the information in its table) or a NAK (if no information is available). The client uses this ATM address to place a call to the target client (this call is referred to as a data channel). Once the call is established, IP datagrams may traverse the link at any time.

Within the CIP model, there are two forms of request/replies: ATM ARP request/replies (referred to as ARPs), and InATMARP request/replies. One could consider InATMARPs as gathering first-hand information. That is, InATMARP is used to query the other end of a VCC for its IP address and ATM address. InATMARP also informs the other end who it is (its IP address and ATM address).

ATMARP could be considered surrogate information. A CIP client sends an ATMARP to the ARP Server to find the ATM address corresponding to the specified IP address. The server replies with the requested information, or with a NAK if the information is not available. However, the RFC requires all clients and servers to respond to ARPs and InATMARPs with the appropriate response.

A user may configure up to 32 LISs per ATM interface. For each LIS, the device can appear as a client only, or can appear as both a client and an ARP Server on that LIS. The device does not support an ARP Server only as this goes against the recommendation of RFC 1577 that each ARP Server should contain an IP address.

Timeouts and Refresh

Both the CIP client and ARP Server age their ARP entries. Once the timer for an ARP entry expires, that entry is deleted. If traffic is flowing when an ARP entry gets aged, that traffic will cease for a period until a new ARP entry is created. To avoid any interruption in service, the device provides an automatic refresh option. This option allows the client to transmit either an ARP request to the ARP Server or a positive InATMARP response only to the target client some time before the ARP entry expires. If the target replies, the timer of the ARP entry is reset. If the target does not, the entry is deleted. The ARP Server automatically sends out an InATMARP message before aging an entry in its table. The Client and ARP Servers default to aging periods of 5 minutes and 20 minutes respectively. These times are configurable for each LIS (client or client/server pair).

IP Addresses and CIP Components

IP addresses are key to IP routing. When configuring the device, the act of adding an IP address to an interface (ATM port), automatically creates a CIP client. The client is defined further by adding ATM ARP client information, but it is the adding of the IP address that creates the client.

Each server, since it contains an IP address, implicitly contains a client as well. When configuring the server, you must configure an IP address. Again, this automatically creates a client. The server is not created until you customize the client configuration. In particular, you must specify that this client is also the server for the LIS. The required databases are then created and maintained to service incoming requests.

The IP address configured does not necessarily imply that the device will act as a router. To act as a router, a higher level routing protocol such as OSPF must be configured. However, if the device is attached to multiple subnets, and if packets are sent to it from one subnet destined to a station on one of the other attached subnets, the device will forward that packet without having any routing protocol configured. Further, if a packet is sent to the device, but the destination of the packet is not the device, and the destination is on the same subnet as the source, the device will send an ICMP redirect message to the originator, and will forward the packet to the proper host.

Because CIP treats ATM as an NBMA network, there is no notion of broadcast. RIP, which is a routing protocol that assumes broadcast will not function in this environment. OSPF, which adapts itself to several types of networks, treats ATM as either a point-to-multipoint network or a Non-Broadcast Multi-Access (NBMA) network which it manages quite well.

ATM Addresses of CIP Components

Each client receives a unique ATM address. As described earlier, only NSAP addresses are supported. The End System Identifier (ESI) and the Selector can be chosen by the person configuring or it may be generated automatically during initialization time. If a device is configured as a client-only on a LIS, then configuring the ESI or Selector is not required (it is recommended that automatic generation be used). If a device is configured as a client/server pair, then it is strongly recommended that you do specify your own Selector, and if necessary, the ESI. (Note that the ESI will default to a burned-in 6-byte value that is unique.) A user will want to specify this information so that the specific ATM address comes up every time for that Server. Clients wishing to connect to this server can rely on the fact that the ATM address of the Server will not change.

If a server/client pair is configured for a specific LIS, then both the server and the client will use the same ATM address. The ATM addresses for each CIP client should be unique.

Virtual Channel Connection (VCC)

A Virtual Channel Connection (VCC) is the lowest common denominator for data transmission. It can either be dynamically created in which case a VCC is a Switched Virtual Circuit (SVC), or it may be configured in the ATM Switch and end stations as a Permanent Virtual Circuit (PVC).

SVCs require a call setup or signalling protocol to establish the connection. Setting up an SVC is similar to placing a phone call. The user dials a phone number and waits for the phone to be answered before communicating to the answering party. If either end hangs up the phone, then the caller must redial the number before talking again. The same is true for ATM SVCs. The host sends out a setup message with a 20-byte ATM address (similar to a phone number), and waits for the other end to connect. Either host can hang-up the channel.

PVCs, on the other hand require no signalling protocol. Nor do they require matching levels of UNI. They are static, and are available to the host from initialization time until power down. The host does not need to take any actions to “set up” the connection. As such, PVCs are simpler and generally more reliable than SVCs.

The device's implementation of Classical IP supports both PVCs and SVCs. SVCs may be generated automatically through the address resolution process and subsequent call setup performed by the Classical IP code, or an SVC may be explicitly configured by the user. Automatic SVCs are brought up and torn down by the ARP subsystem as required for sending IP traffic. A configured SVC is brought up during initialization, and is kept up indefinitely. If the configured SVC does not connect, the device continues to retry the connection periodically until power is turned off.

PVCs and configured SVCs require no ARP Server definition. That is, a LIS could consist of hosts that were interconnected only by configured information. Optionally, the destination IP address of a configured PVC or SVC can be configured as well. If the IP address is not configured, InATMARP packets are used to determine what IP address sits at the opposite end of a VCC. For a network of any size, the amount of manual configuration would become prohibitive.

Automatically generated SVCs drastically reduce the amount of configured information, and provide maximum flexibility for adding and moving hosts.

Automatically generated VCCs can only exist with the assistance of an ARP Server. Each client must be configured with the ARP Server's ATM address. Immediately after initialization, the client will attempt to connect to the server. This connection is referred to as a control channel. The principal use of a control channel is for sending ATMARP and InATMARP requests and replies, although if the ARP Server is also a client, the control channel also can be used for sending IP data. Automatic VCCs generated to send data from one host to another are referred to as data channels.

The attributes of both control and data channels can be tailored to the user's needs. The CIP configuration of the device allows for configuration of the Peak Cell Rate, Sustained Cell Rate, maximum SDU sizes and other characteristics of the control and data channels set up by the device. A user also can choose to limit the cell rates of incoming calls to avoid the problems caused by mismatches in bandwidths of the various ATM attachments.

Key Configuration Parameters for Classical IP

The simplicity of CIP is that very few configuration parameters are required. For a client-only, two pieces of information are required:

1. The IP address and Subnet mask. (add address)
2. The ATM address of the ARP Server. (add arp-server)

The IP address and subnet mask are required to give the client its unique IP identity so that it can send and receive IP datagrams. It also defines the subnet to which this CIP client belongs. The ATM address of the ARP Server is used by the client during initialization to establish a control channel with the ARP Server.

The configuration of the server is similarly simple. Essentially, the server needs to be defined with a fixed, well-known ATM address, and it needs to know which LIS it is serving. The server configuration requires the following:

1. The IP address and Subnet mask. (add address)
2. Answering “Yes” to the question about whether this client is also a server. (add atm-client-configuration)
3. Specifying an explicit selector for the server's ATM address (answering “no” when asked if you wish to use the internally assigned selector). (add atm-client-configuration)

The IP address and Subnet mask tell the server which LIS it is serving. The IP address also gives IP access to the server and routing function if desired (through the implicit client). Questions 2 and 3 are asked, among others, in the “add atm-client-configuration” Question 2 is required to enable the server function for that LIS. Question 3 is used to give the server a predictable ATM address.

How to Enter Addresses

Addresses are entered in two ways, depending on whether the address represents (1) an IP address, or (2) an ATM address, MAC address, or route descriptor, as follows:

1. IP address

IP addresses are entered in dotted decimal format, a four-byte field represented by four decimal numbers (0 to 255) separated by periods (.).

Example of IP Address:

```
01.255.01.00
```

2. ATM or MAC address or route descriptor

   ATM addresses, MAC addresses, and route descriptors are entered as strings of hexadecimal characters with or without optional separator characters between bytes. Valid separator characters are dashes (–), periods (.), or colons (:).

   Examples of ATM address, MAC address or route descriptor:

```
A1FF010203
      or
A1–FF–01–02–03
      or
A1.FF.01.02.03
      or
39.84.0F.00.00.00.00.00.00.00.00.00.03.10.00.5A.00.DE.AD.C8
      or
A1:FF:01:02:03
      or even
A1–FF.01:0203
```

   This applies to addresses entered for ATM, LAN emulation, and Classical IP & ARP over ATM.

IPX and ARP Over ATM Overview (RFC 1483)

The 2216 uses LLC/SNAP encapsulation as specified by RFC 1483 to carry IPX traffic over ATM. 2216s (and other routers that support RFC 1483 LLC/SNAP encapsulation on ATM) can be interconnected in full or partial meshes via manually-configured RFC 1483 connections. Both PVCs and configured SVCs are supported. However, VCCs to IPX routers must be dedicated to IPX; they cannot be shared with other protocols, such as IP.

The 2216 supports a single IPX network per ATM interface. This implies a single ATM ARP client per interface for IPX which must be explicitly configured. Therefore, all interconnected routers on the ATM interface must be part of the same IPX network.

IPX ATM addresses must be unique among all components using RFC 1483 encapsulation (which includes Classical IP components). The ESI and the selector portions of IPX ATM addresses are configured in the same manner as Classical IP ATM addresses. If the 2216 is not initiating the SVC, then at least the selector should be explicitly specified in the current configuration to provide a fixed address that can be configured at the calling router.

IPX protocol addresses have two parts:

- 4-byte network number, and
- 6-byte host number (or host ID)

Network numbers must be unique within IPX routing domains, and host numbers must be unique within a given network. The IPX host number is set (by the 2216) to the ESI component of the associated ATM address. The ESI defaults to the MAC address burned into the ATM interface hardware in case that one is not explicitly configured by the user.

Destination IPX host numbers may be specified during VCC configuration or learned dynamically via InATMARP. You must manually configure the IPX host numbers of destination routers that do not support InATMARP. InATMARP is also used to periodically refresh the 2216's knowledge of a connected router's IPX host number.

Routers that are interconnected in a partial mesh and are providing intermediate routing between routers on the same ATM interface should disable IPX split-horizon on the ATM interface. This ensures RIP and SAP properly inform the interconnected routers of all available routes and services. Routers that are interconnected in a full mesh need not disable split-horizon.

Accessing the ARP Configuration Environment

For information on how to access the ARP configuration environment, see “Getting Started (Introduction to the User Interface)” in the Software User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1.

Use the following procedure to access the ARP *configuration* process.

1. At the OPCON prompt, enter **talk 6**. (For more detail on this command, refer to *The OPCON Process and Commands* in the Software User’s Guide.) For example:

```
* talk 6
Config>
```

After you enter the **talk 6** command, the CONFIG prompt (`Config>`) displays on the console. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the CONFIG prompt, enter the **prot arp** command to get to the `ARP Config>` prompt.

ARP and Inverse ARP Configuration Commands

This section summarizes and then explains all the ARP configuration commands. Table 20-1 lists the ARP configuration commands. You can access ARP configuration commands at the `ARP config>` prompt.

Note: These commands are used to manage the ARP table for emulated LANs. They have no effect on the Classical IP ARP table that is associated with the ATM physical interface.

Table 20-1. ARP Configuration Commands Summary

Command	Function
? (Help)	List the ARP configuration commands or list the options associated with specific commands.
Add Entry	Add a MAC address translation entry.
Change Entry	Change a MAC address translation entry.
Delete Entry	Deletes a MAC address translation entry.
Disable Auto-refresh	Disable ARP auto-refresh.
Enable Auto-refresh	Enable ARP auto-refresh.
List	List ARP configuration data in SRAM.
Set	Set the usage and refreshes timeout values.
Exit	Exit the ARP configuration process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

Example:

```
ARP Config> ?
LIST
ADD
CHANGE
DELETE
DISABLE
ENABLE
SET
EXIT
```

or

`set ?`

Add Entry

Use the **add entry** command to add a “static protocol-to-hardware address mapping” entry. This command is currently supported for IP addresses only.

Syntax: add entry *ifc# prot-type prot-addr MAC-addr*

ifc# **Valid values**: any defined interface

Default value: 0

prot-type
Valid values: any protocol that ARP supports.

Default value: IP

prot-addr
Valid Values: any valid IP address

Default Value: 0

MAC-addr
Valid Values: any valid MAC address

Default Value: none

Example: **`add entry`**

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
Mac Address []?
```

Change Entry

Use the **change entry** command to change a “static protocol-to-hardware address mapping” entry. This command is currently supported for IP addresses only. The hardware address parameter (MAC-addr) should be the address of the node being changed.

ifc# **Valid values**: any defined interface

Default value: 0

prot-type
Valid values: any protocol that ARP supports.

Default value: IP

prot-addr
Valid Values: any valid IP mask

Default Value: none

MAC-addr
Valid Values: any valid MAC address

Default Value: none

Syntax: change entry *ifc# prot-type prot-addr MAC-addr*

Example: **`change entry`**

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
Mac Address []?
```

Delete Entry

Use the **delete entry** command to delete a “static protocol-to-hardware address mapping” entry. This command is currently supported for IP addresses only.

ifc# **Valid values**: any defined interface

Default value: 0

prot-type
Valid values: *IP* or *IPX*

Default value: IP

prot-addr
Valid Values: any valid IP address

Default Value: 0.0.0.0

Syntax: delete entry *ifc# prot-type prot-addr*

Example: **delete entry**

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
```

Disable Auto-Refresh

Use the **disable auto-refresh** command to disable the auto-refresh function. The auto-refresh function is the router’s capability to send an ARP request based on the entry in the translation cache before the refresh timer expires. The request is sent directly to the hardware address in the current translation instead of a broadcast. If auto-refresh is disabled, no ‘preemptive’ ARP request is made, the refresh timer is allowed to expire, and the ARP translation is purged from the table. The next protocol packet to the destination protocol address will then cause a new ARP request to be broadcast on the network.

Syntax: disable auto-refresh

Example: **disable auto-refresh**

Enable Auto-Refresh

Use the **enable auto-refresh** command to enable the auto-refresh function. The auto-refresh function is the router’s capability to send an ARP request based on the entry in the translation cache before the refresh timer expires. The request is sent directly to the hardware address in the current translation instead of a broadcast.

Enabling auto-refresh could cause entries to be retained in the cache regardless of their usage. On networks with a large number of nodes, this can lead to an excessive number of entries in the cache, which might adversely affect router performance. However, on networks with a small number of nodes, this option is useful in reducing broadcast ARP traffic.

Syntax: enable auto-refresh

Example: `enable auto-refresh`

List

Use the **list** command to display the contents of the router's ARP configuration as stored in SRAM. The list command displays the current timeout settings for the refresh and usage timer.

Syntax: list all
config
entry

all

Lists the ARP configuration followed by all of the ARP entries.

Example: `list all`

```
ARP configuration:

Refresh Timeout: 5 minutes
Auto Refresh: disabled

Mac address translation configuration
IF #           Prot #         Protocol --> Mac Address
 0               0            2.2.2.1 --> 0000C90932EF
```

config

Lists the configuration. for the different ARP parameters.

Example: `list config`

```
ARP configuration:

Refresh Timeout: 5 minutes
Auto refresh: disabled
```

entry

Lists the ARP entries in SRAM.

Example: `list entry`

```
Mac address translation configuration

IF #           Prot #         Protocol --> Mac Address
 0               0            2.2.2.1 --> 0000C90932EF
```

Set

Use the **set** command to set an ARP configuration parameter.

Syntax: set refresh-timer

refresh-timer *minutes*

Changes the timeout value for the refresh timer. To change the timeout value for the refresh timer, enter the timeout value in minutes. A setting of zero (0) turns off (disables) the refresh timer.

This timer is used in determining when an ARP translation cache entry is to be refreshed while auto-refresh is enabled, or purged while auto-refresh is disabled. Disabling the timer causes entries to be retained until a newly learned address translation causes entries to be removed, until entries are cleared manually with the ARP **clear** console command, or until the router is restarted.

Valid Values: an integer number of minutes in the range of 0 - 65535

Default Value: 5 minutes

Example: `set refresh-timer 3`

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

ARP Over ATM Configuration Commands

This section summarizes and explains the ARP Over ATM configuration commands. These commands apply to:

- Classical IP & ARP over ATM
- IPX over ATM

Enter the commands at the `ARP Config>` prompt.

Differences for IP and IPX

Configuring IPX over ATM (using RFC 1483) is similar to configuring Classical IP (RFC 1577).

Once you enter "IPX" as the protocol, some subsequent questions are different than those for protocol "IP." Since IPX over ATM does not use ARP servers, questions relating to ARP servers are not asked.

Also, IPX over ATM requires fewer parameters to be configured than Classical IP. The IPX network number and the IPX host number (IPX ATM-ARP-client) are the only required parameters for IPX over ATM. If you need to open a connection to a remote IPX router, you must additionally configure the desired channels (VCCs).

Effect on ARP Table Entries

These commands apply only to the physical ATM interface where the ARP entries reside for ARP over ATM. These commands will have no effect on a non-ATM interface such as an Emulated LAN.

Table 20-2. ARP Over ATM Configuration Command Summary

Command	Function
? (Help)	Lists all of the ARP over ATM configuration commands, or lists the options associated with specific commands. ?
Add	Adds an arp-server, atm-arp-client-configuration, pvc-atm-arp-entry or svc-atm-arp-entry.
Change	Changes the atm-arp-client-configuration.
Delete	Deletes an arp-server, atm-arp-client-configuration, pvc-atm-arp-entry or svc-atm-arp-entry.
List	Lists all (the current ARP over ATM configuration), lists ARP servers (for IP only), or lists pvc-atm-arp-entries and svc-atm-arp-entries.
Exit	Exits the ARP over ATM configuration process and returns to the `Config>` prompt.

? (Help)

Use the **?** (**help**) command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: list output

Example: `ARP Config>` **?**list output

Add

Use the **add** command to add an arp-server, atm-arp-client-configuration, or pvc-atm-arp-entry.

Syntax: add arp-server
atm-arp-client-configuration
pvc-atm-arp-entry
svc-atm-arp-entry

arp-server private-nsapa

Adds an arp-server to the client specified. Only one ARP Server is allowed per client. During initialization, the specified Classical IP client will place a call to the ARP server, and will use it as the mechanism for resolving IP addresses to ATM addresses. If a CIP client is configured to also be a server, then this command will override the client configuration and the client goes to the remote ARP Server to resolve all addresses.

local client IP address

Valid Values: any valid IP mask

Default Value: none

The *private-nsapa* field is the Private Network Specified Access Point Address that is the addressing format specified in the UNI versions 3.0 and 3.1. The first byte of the *nsapa* defines the addressing format, as follows:

First Byte	*NSAP Address Format Specification*
0x39	DCC ATM Format
0x47	ICD ATM Format
0x45	E.164 ATM Format

Note: This setting corresponds to a client's (IP address/port number) pair.

Default value: none

Example:

```
ARP config> add arp-server private-nsapa
Local Client IP Address [0.0.0.0]? 2.2.3.100
Private NSAP Address: Specify 40 digits
ATM Address []? 39840f0000000000000000000410005a3345f3a0
```

Local Client IP Address	Address of this client.
ATM NSAP Address	Address of the Remote ARP Server

atm-arp-client-configuration

Adds atm-arp-client-configuration.

You will be prompted to provide information about the characteristics of the VCCs that will be set up and received by this client, the refresh timeout and auto-refresh settings, how the ATM address for this client is determined, and the frame size that this client can handle.

Note: Any bandwidth or cell parameter that equals zero will be treated as the line speed of the ATM interface.

Example for IP:

```
ARP config> add atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]?
Client IP Address [0.0.0.0]? 2.2.3.100
This client is also a server? [Yes]: no
Refresh timeout (in minutes) [5]?
Enable auto-refresh? [No]:
Refresh by InAtmArp? [Yes]:
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [Yes]:
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Service for Control VCCs? [Yes]:
Peak Cell Rate of outbound control VCCs (Kbps) [0]?
Sustained Cell Rate of outbound control VCCs (Kbps) [0]?
Use Best Effort Service for Data VCCs? [Yes]:
Peak Cell Rate of outbound Data VCCs (Kbps) [0]?
Sustained Cell Rate of outbound Data VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
```

Interface Number Interface number assigned.

Valid values: any interface on the device

Default value: 0

Protocol **Valid values**: *IP* or *IPX*

Default value: IP

Client IP Address Client IP Address (IP only)

Valid Values: any valid IP address

Default value: 0.0.0.0

This client is also a server

Yes or No. If no, client is not a server. (IP only)

Refresh timeout (in minutes)

Refresh timeout value in minutes.

Valid Values: an integer number of minutes in the range of 0 - 64000

Default Value: 5 minutes

Enable auto-refresh YES or NO.

Refresh by InAtmArp YES or NO. If YES, and if auto-refresh is enabled, then InAtmArp requests will be periodically transmitted to confirm the existence of the remote host. If NO, then AtmArp requests will be transmitted to the ARP Server to reconfirm the ARP entry.

Use burned-in ESI as part of the ATM address. You might be given other choices depending on your configuration.

Select ESI Specify the ESI index number from the the number of any defined ESI

Default Value: 1 which specifies to used the burned in address “pick list” (the list of configured ESIs that you previously configured using the **add esi** network configuration command). The default item in the pick list is to use the ATM interface's burned in ESI.

Use internally assigned selector

Use internally assigned selector.

Valid Values: any single octet value that has not been previously used and is within the range defined for the device.

Default Value: none

Validate PCR for best effort VCCs

TRUE or FALSE. When true, Best-Effort VCCs will be rejected if the signaled forward PCR exceeds the Maximum Reserved Bandwidth or the speed of the adapter. If false, Best-Effort PCRs will be rejected without regard to the signaled Peak Cell Rate.

Maximum Reserved Bandwidth for incoming VCCs (Kbps)

Defines the maximum acceptable Sustained Cell Rate (SCR) for an incoming VCC. If SCR is not specified on the incoming call, then this parameter defines the maximum acceptable Peak Cell Rate (PCR). Calls received with traffic parameters specifying higher rates will be released. This parameter is applied to both forward and backward Cell Rate parameters. The constraint imposed by this parameter is applicable to best effort connections (if “validate PCR” is yes) and is compared to the PCR on the incoming call.

Valid Values: any single octet value that has not been previously used and is within the range defined for the device.

Default Value: none

Use Best Effort Service for Control VCCs

Specifies the type of traffic characteristics to be associated with Control VCCs. Bandwidth is not reserved for best effort traffic. **Valid Values**: *Best Effort* or *Reserved Bandwidth*

Default Value: Best Effort

Peak Cell Rate of outbound control VCCs (Kbps)

Specifies the Peak Cell Rate (PCR) traffic parameter for the Control VCC. This PCR value is used for both the forward and backward PCR values of both best effort and reserved bandwidth VCCs.

Valid Values: an integer Kbps in the range of 0 - line speed of the ATM device

Default Value:

- If best effort, default is max data rate
- If reserved, there is no default

Sustained Cell Rate of outbound control VCCs (Kbps)
Specifies the bandwidth reserved by all VCCs on a given ATM device. (Sustained Cell Rate can be considered to reserved bandwidth.) This parameter is applicable only when Best Effort Service is not selected for Control VCCs.

Valid Values: an integer Kbps in the range of 0 - control VCC PCR

Default Value: none

Use Best Effort Service for Data VCCs
Yes or No. Specifies the type of traffic characteristics to be associated with Data VCCs. Bandwidth is not reserved for best effort traffic.

Peak Cell Rate of outbound Data VCCs (Kbps)
Specifies the Peak Cell Rate (PCR) traffic parameter for the Data VCC. This PCR value is used for both the forward and backward PCR values of both best effort and reserved bandwidth VCCs.

Valid Values: an integer Kbps in the range of 0 - control VCC PCR

Default Value: 0

Sustained Cell Rate of outbound Data VCCs (Kbps)
Specifies the Sustained Cell Rate (SCR) traffic parameter for the Data VCC. (Sustained Cell Rate can be considered to reserved bandwidth.) This parameter is applicable only when Best Effort Service is not selected for Data VCCs.

Valid Values: an integer Kbps in the range of 0 - PCR value for Data VCC

Default Value: none

Max SDU size (bytes)
Specifies the Maximum SDU size that will be specified when calls are placed from this client address. It also is used to verify incoming calls. This parameter cannot be set to a value greater than the Maximum SDU size for the physical ATM interface (port).

Valid Values: an integer in the range of 72 - Maximum interface SDU

Default Value: 9188

Example for IPX:

```
ARP config> add atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]? IPX
Refresh timeout (in minutes) [5]?
Enable auto-refresh? [Yes]:
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [Yes]:
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Service for Control VCCs? [Yes]:
Peak Cell Rate of outbound control VCCs (Kbps) [0]?
Sustained Cell Rate of outbound control VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
```

For field descriptions, refer to the preceding example for IP.

pvc-atm-arp-entry

Adds a PVC and optionally creates permanent ARP Entry if the destination protocol address is specified. For virtual ATM interfaces, you should check the configuration of the real ATM interface where the AVI sits and all the other AVIs configured on the real ATM interface. A new VPI/VCI pair is needed for a new PVC unless you specifically want to share the new PVC traffic with the traffic of an existing PVC.

interface number

Valid values: the number of the interface over which the device will boot or dump.

Default value: 0

protocol

Valid values: *IP* or *IPX*

Default value: IP

local client IP address

Required for IP

Valid Values: any valid IP address

Default Value: 0.0.0.0

destination protocol address

Valid Values: any valid IP address

Default Value: 0.0.0.0

destination ATM address

Valid Values: any valid IP address

Default Value: none

permanent virtual circuit VPI

Valid Values: any valid value in the range of 0 - 255

Default Value: 0

permanent virtual circuit VCI

Valid Values: any value in the range of 0 - 65535

Default Value: 0

Example for IP:

```
ARP config> add pvc-atm-arp-entry
Interface Number [0]?
Protocol [IP]?
Local client IP address [0.0.0.0]? 2.2.3.100
Specify destination protocol address? [Yes]: no
Permanent Virtual Circuit VPI, Range 00..FF [00]?
Permanent Virtual Circuit VCI, Range 0000..FFFF [0000]? 0029
```

Interface Number Interface number assigned.

Protocol IP or IPX

Specify destination protocol address
If yes, you will be required to enter the destination protocol address.

Example for IPX:

```
ARP config> add pvc-atm-arp-entry
Interface Number [0]?
Protocol [IP]? IPX
Specify destination protocol address? [Yes]: no
Permanent Virtual Circuit VPI, Range 00..FF [00]?
Permanent Virtual Circuit VCI, Range 0000..FFFF [0000]? 0037
```

If you choose to specify a destination protocol address, enter a valid 6-byte IPX host address.

For field descriptions, refer to the preceding example for IP.

svc-atm-arp-entry
Adds an SVC and optionally creates a permanent ARP Entry.

interface number
Valid values: the number of the interface over which the device will boot or dump.

Default value: 0

protocol
Valid values: *IP* or *IPX*

Default value: IP

local client IP address
Required for IP

Valid Values: any valid IP address

Default Value: 0.0.0.0

destination protocol address
Valid Values: any valid IP address

Default Value: 0.0.0.0

destination ATM address
Valid Values: any valid IP address

Default Value: none

Example for IP:

```
ARP config> add svc-atm-arp-entry
Interface Number [0]?
Protocol [IP]?
Local client IP address [0.0.0.0]? 2.2.3.100
Specify destination protocol address? [Yes]: no
Destination ATM Address []? 39840f000000000000000000210005a00dead03
```

Interface Number Interface number assigned.

Protocol IP or IPX.

Specify destination protocol address
If no, it is resolved by InATMARP requests/replies.

Destination ATM Address Destination ATM address.

Example for IPX:

```
ARP config> add svc-atm-arp-entry
Interface Number [0]?
Protocol [IP]? IPX
Specify destination protocol address? [Yes]: no
Destination ATM Address []? 39840f000000000000000000210005a00dead03
```

If you choose to specify a destination protocol address, enter a valid 6-byte IPX host address.

For field descriptions, refer to the preceding example for IP.

Change

Use the **change** command to change the ATM-ARP configuration.

Syntax: change
entry
atm-arp-client-configuration

atm-arp-client-configuration
Changes the atm-arp-client-configuration.

interface number
Valid values: any interface on the device

Default value: 0

protocol
Valid values: *IP* or *IPX*

Default value: IP

If you enter IP, you must enter the client IP addresss.

client IP address
Valid Values: any valid IP address

Default value: 0.0.0.0

refresh timeout
Valid Values: an integer number of minutes in the range of 0 - 65535

Default Value: 5 minutes

ESI a defined ESI index number.

maximum reserved bandwidth for incoming VCCs

Valid values: the number of any defined ESI

Default Value: 1 which specifies to used the burned in address

selector value

You enter this if you respond **no** to the “Use internally assigned selector” question.

Valid Values: any single octet value that has not been previously used and is within the range defined for the device.

Default Value: none

maximum reserved bandwidth for incoming VCCs

Valid Values: an integer in the range of 0 to the line speed of the ATM device.

Default Value: 0

use best effort service for control VCCs

Valid Values: *Best Effort* or *Reserved Bandwidth*

Default Value: Best Effort

peak cell rate of outbound control VCCs

Valid Values: an integer Kbps in the range of 0 - line speed of the ATM device

Default Value:

- If best effort, default is max data rate
- If reserved, there is no default

sustained cell rate of outbound control VCCs

Valid Values: an integer Kbps in the range of 0 - control VCC PCR

Default Value: none

peak cell rate of outbound data VCCs

Valid Values: an integer Kbps in the range of 0 - control VCC PCR

Default Value: 0

sustained cell rate of outbound data VCCs

Valid Values: an integer Kbps in the range of 0 - PCR value for Data VCC

Default Value: 0

max service data unit (SDU) size

Valid Values: an integer in the range of 72 - Maximum interface SDU

Default Value: 9188

Specify the IP address of the atm-arp-client.

Example for IP:

```
ARP config> change atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]?
Client IP Address [0.0.0.0]? 1.1.1.100
This client is also a server? [No]:
Refresh timeout (in minutes) [5]?
Enable auto-refresh? [No]: yes
Refresh by InAtmArp? [Yes]:
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [Yes]:
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Service for Control VCCs? [Yes]:
Peak Cell Rate of outbound control VCCs (Kbps) [0]?
Sustained Cell Rate of outbound control VCCs (Kbps) [0]?
Use Best Effort Service for Data VCCs? [Yes]:
Peak Cell Rate of outbound Data VCCs (Kbps) [0]?
Sustained Cell Rate of outbound Data VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
```

For field descriptions, refer to the example for **add atm-arp-client-configuration**.

Example for IPX:

```
ARP config> change atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]? IPX
Refresh timeout (in minutes) [5]?
Enable auto-refresh? [Yes]:
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [No]:
Selector Only, Range 00..FF [00]? 20
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Service for Data VCCs? [Yes]:
Peak Cell Rate of outbound data VCCs (Kbps) [0]?
Sustained Cell Rate of outbound Data VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
```

Since only one IPX ATM-ARP client configuration record exists for an ATM interface, you are not prompted to enter a protocol address.

For field descriptions, refer to the preceding example for IP.

Use of this command is required only if you wish to use values other than the defaults for the traffic parameters.

For field descriptions, refer to the preceding example for IP.

Delete

Use the **delete** command to delete an arp-server, atm-arp-client-configuration, pvc-atm-arp-entry, or svc-atm-arp-entry.

Syntax: delete
entry
arp-server
atm-arp-client-configuration
pvc-atm-arp-entry
svc-atm-arp-entry

arp-server

Deletes an arp-server.

Specify the address of the arp-server.

Valid Values: any valid IP address

Default Value: 1.1.1.100

Example for IP:

```
ARP config> del arp-server
IP Address [1.1.1.100]? 2.2.3.100
Arp Server entry found and deleted
```

atm-arp-client-configuration

Deletes an atm-arp-client-configuration.

Specify the interface number, protocol, and Client IP address.

interface number

Valid values: any defined interface

Default value: 0

protocol

Valid values: *IP* or *IPX*

Default value: IP

client IP address

Valid Values: any valid IP address

Default Value: 1.1.1.100

Example for IP:

```
ARP config> del atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]?
Client IP Address [1.1.1.100]? 2.2.3.100
ATM ARP Client Config record deleted
```

Example for IPX:

```
ARP config> del atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]? IPX
ATM ARP Client Config record deleted
```

Since only one IPX ATM-ARP client configuration record exists for an ATM interface, you are not prompted to enter a protocol address.

For field descriptions, refer to the preceding example for IP.

pvc-atm-arp entry

Deletes a pvc-atm-arp-entry.

Specify the entry number for the pvc-atm-arp-entry you want deleted.

Example for IP and IPX:

```
ARP config> del pvc

ATM Arp Permanent Virtual Circuit Definitions
No. IF# Prot# P/S     Protocol     -> VPI / VCI
1    0    0    P         0.0.0.0   ->  00 / 0029
2    0    7    P 00.00.00.00.00.00 ->  00 / 0037
Which Arp entry do you want to delete [0]? 1
ATM Arp entry 1 being deleted
```

No. 1 is an IP PVC and No. 2 is an IPX PVC.

svc-atm-arp-entry

Deletes an svc-atm-arp-entry.

Specify the entry number for the svc-atm-arp-entry you want deleted.

Example for IP and IPX:

```
ARP config> del svc

ATM Arp Switched Virtual Circuit Definitions
No. IF# Prot# P/S     Protocol     -> Destination ATM Address
1    0    0    S         0.0.0.0   ->
                39.84.0F.00.00.00.00.00.00.00.00.00.02.10.00.5A.00.DE.AD.03
2    0    7    P 00.00.00.00.00.00 ->
                39.84.0F.00.00.00.00.00.00.00.00.00.02.11.00.B7.38.AA.BB.12

Which Arp entry do you want to delete [0]? 1
ATM Arp entry 1 being deleted
```

No. 1 is an IP SVC and No. 2 is an IPX SVC.

List

Use the **list** command to display the contents of the router's ARP configuration as stored in SRAM. The list command also displays the current settings for the refresh and usage timer.

Syntax: list

entry
all
arp-servers
atm-arp-client-configuration
pvc-atm-arp-entry
svc-atm-arp-entry

all Lists the ARP configuration followed by all of the ARP entries.

Example: `list all`

```
ARP config> list all
ARP configuration:

Refresh timeout: 5 minutes
Auto refresh: disabled

Mac address translation configuration

No arp entries defined

ATM Arp Server List:
   IP Address   L/R    Address / Sub Address
   1.1.1.100  R  39.84.0F.00.00.00.00.00.00.00.00.00.02.10.00.5A.00.DE.AD.02
```

arp-servers

Lists arp-servers.

```
 ARP config> list arp-servers

 ATM Arp Server List:
IP Address   L/R    Address / Sub Address
   1.1.1.100  R  39.84.0F.00.00.00.00.00.00.00.00.00.02.10.00.5A.00.DE.AD.02
```

atm-arp-client-configuration

Lists the atm-arp-client-configuration.

```
ARP config> list atm-arp-client-configuration

ATM Arp Clients:
--------------------------------------------------
If: 0  Prot: 0  Addr: 1.1.1.100       ESI: burned-in         Sel: auto
Server: no   Refresh T/O: 5    AutoRefr: no    By InArp: yes  Validate PCR: no
Use Best Effort: yes/yes  (Control/Data)   Max B/W(kbps):      0
Cell Rate(kbps):  Peak:      0/     0    Sustained:      0/     0
Max SDU(bytes):   9188
--------------------------------------------------
If: 0  Prot: 0  Addr: 3.3.3.3         ESI: burned-in         Sel: auto
Server: yes  Refresh T/O: 5    AutoRefr: no    By InArp: yes  Validate PCR: no
Use Best Effort: yes/yes  (Control/Data)   Max B/W(kbps):      0
Cell Rate(kbps):  Peak:      0/     0    Sustained:      0/     0
Max SDU(bytes):   9188
--------------------------------------------------
If: 0  Prot: 0  Addr: 4.4.4.4         ESI: burned-in         Sel: auto
Server: yes  Refresh T/O: 5    AutoRefr: no    By InArp: yes  Validate PCR: no
Use Best Effort: yes/yes  (Control/Data)   Max B/W(kbps):      0
Cell Rate(kbps):  Peak:      0/     0    Sustained:      0/     0
Max SDU(bytes):   9188
```

If: Interface Number

Prot: 0 = IP, 7 = IPX

Addr: IP Address

ESI: End System Identifier

Sel: Selector (the last byte in the ATM address, following the ESI). If AUTO, the selector is generated at run-time.

Server: YES, this client is also a server; NO, this client is not a server.

Refresh T/O: Refresh Timeout value in minutes.

AutoRefr YES or NO.

By InArp: YES or NO. If YES, and if auto-refresh is enabled, then InAtmArp requests will be periodically transmitted to confirm the existence of the remote host. If NO, then AtmArp requests will be transmitted to the ARP Server to reconfirm the ARP entry.

Validate PCR: TRUE or FALSE. When true, Best-Effort VCCs will be rejected if the signaled forward PCR exceeds the Maximum Reserved Bandwidth or the speed of the adapter. If false, Best-Effort PCRs will be rejected without regard to the signaled Peak Cell Rate.

Use Best Effort: Specifies the type of traffic characteristics to be associated with the Control or Data VCCs.

Max B/W(kbps): Maximum bandwidth (kbps).

Cell Rate(kbps): Peak Cell Rate for the Control or Data VCC.

Max SDU(bytes): The maximum SDU size that is specified.

pvc-atm-arp-entry
Lists ARP PVCs.

```
ARP config> list pvc

ATM Arp Permanent Virtual Circuit Definitions
No. IF# Prot# P/S     Protocol     -> VPI / VCI
1    0    0    P         0.0.0.0   ->  00 / 0029
```

ATM No.	ATM interface number
ARP IF#.	ARP Interface Number
Prot#	Protocol number (Prot# 0 = IP, 7 = IPX)
P/S:	P for PVC, S for SVC
Protocol	IP Address
VPI/VCI	The decimal value of the defined channel.

svc-atm-arp-entry

Lists ARP SVCs.

```
ARP config> list svc

ATM Arp Switched Virtual Circuit Definitions
No. IF# Prot# P/S     Protocol     -> Destination ATM Address
2    0    0    S         0.0.0.0   ->
                 39.84.0F.00.00.00.00.00.00.00.00.00.02.10.00.5A.00.DE.AD.03
```

ATM No.	ATM interface number
ARP IF#.	ARP Interface Number
Prot#	Protocol number (Prot# 0 = IP, 7 = IPX)
P/S:	P for PVC, S for SVC
Protocol	IP Address
Destination ATM Address	Destination ATM Address

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 21. Monitoring ARP

This chapter describes how to monitor ARP protocol activity and how to use the ARP console commands and includes the following sections:

- “Accessing the ARP Console Environment”
- “ARP Console Commands” on page 21-2
- “ARP Over ATM Console Commands” on page 21-6

Note: If the device's software load does not contain Asynchronous Transfer Mode (ATM), ATM-related commands are not valid and are not displayed at the ARP configuration and console prompts.

Accessing the ARP Console Environment

Use the following procedure to access the ARP console commands. This process gives you access to the ARP *monitoring* process.

1. At the OPCON prompt, enter **talk 5**. (For more detail on this command, refer to *The OPCON Process and Commands* in the Software User’s Guide.) For example:

   ```
   * talk 5
   +
   ```

   After you enter the **talk 5** command, the GWCON prompt (+) displays on the console. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the + prompt, enter the **protocol arp** command to get you to the `ARP>` prompt.

   Example:

   ```
   + prot arp
   ARP>
   ```

ARP Console Commands

This section summarizes and then explains the ARP console commands. You can access ARP console commands at the `ARP>` prompt. Table 21-1 shows the commands.

Table 21-1. ARP Console Command Summary

Command	Function
? (Help)	List the ARP console commands or list the options associated with specific commands.
Clear	Clear the cache for a specified interface.
Dump	Display the cache for a specified interface.
Hardware	List each ARP-configured network.
Protocol	List each ARP-configured protocol.
Statistics	Display ARP information.
Exit	Exit the ARP console process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: `?`

Clear

Use the **clear** command to flush the ARP cache for a given network interface. The **clear** command can be used to force the deletion of bad transactions.

To clear a particular interface, enter the interface or network number as part of the command. To obtain the interface number, use the CONFIG **list devices** command.

Syntax: clear *interface#*

Example: `clear 1`

Dump

Use the **dump** command to display the ARP cache for a given network/protocol combination. To display the ARP cache for a particular interface, enter the interface or network number as part of the command. To obtain the interface number, use the CONFIG **list devices** command.

If there is more than one protocol on that network, the protocol number must also be given. This causes the console to display the hardware address-to-protocol mappings stored in that database. If ARP is in use by only one protocol on the specified interface, then the protocol number is optional. To obtain the protocol number, use the CONFIG **protocol** command.

The **dump** command display shows the hardware address, the protocol address, and the refresh timer parameter for each mapping.

Syntax: dump *interface# protocol#*

Example: **dump 2 ip**

```
Hardware Address    IP Address      Refresh
02-07-01-00-00-01   192.9.1.2       Permanent
a1-b2-c3-4d-5e-6f   128.185.214.36  5
100                 128.185.123.51  Not Aging
16                  128.185.214.38  Not Aging
```

Valid refresh timer parameters are:

Permanent A statically configured mapping between hardware address and protocol address (entered using the ARP **add entry** command, or the frame-relay **add protocol** command, or the X25 **add address** command). These entries do not age and are not overwritten by dynamically learned mappings.

minutes to expire The number of minutes until this mapping expires due to aging or until this mapping is refreshed (if auto-refresh is enabled). This parameter is expressed as a numeric value.

Not Aging A fixed SVC or PVC mapping learned through Inverse ARP. It begins to age only when the circuit goes down. The mapping can be overwritten by a newer learned address and can be cleared by the ARP **clear** console command.

Hardware

Use the **hardware** command to display the networks registered with ARP. The **hardware** command lists each ARP-registered network, and displays each network's hardware address space (Hardware AS) and local hardware address.

Syntax: hardware

Example: **hardware**

```
 Network       Hardware AS    Hardware Address

 1 FR/0          000F            1023
 5 TKR/0         0006            00:00:C9:09:32:EF
 8 Eth/0         0001            AA-00-04-00-26-14
 9 IPPN/0        2048            128.185.214.38
10 BDG/0         0001            00-00-93-90-4C-F7
```

Note: The IPPN entry refers to IP Tunneling where the hardware address field indicates the IP address of the IP Tunnel.

Ping

Use the **ping** command to have the router send ICMP Echo Requests to a given destination. For more information on the **ping** command, see "Ping" on page 17-7.

Protocol

Use the **protocol** command to display (by network) the protocols that have addresses registered with ARP. This command displays the network, protocol name, protocol number, protocol address space (in hexadecimal), and local protocol addresses.

Syntax: protocol

Example: **protocol**

```
Network  Protocol (num)  AS    Protocol Address(es)
5 TKR/0  IP       (00)   800    128.185.209.38
6 TKR/1  IP       (00)   800    10.1.181.38
8 Eth/0  IP       (00)   800    128.185.221.38
8 Eth/0  AP2      (22)   80F3   221/38
```

Note: SR entries refer to Source Routing - the protocol address is used to indicate the MAC address. Use the token-ring **dump** command to view actual RIF entries.

Statistics

Use the **statistics** command to display a variety of statistics about the operation of the ARP module.

Syntax: statistics

Example: **statistics**

```
ARP input packet overflows
Net   Count
 PPP/0  0
 PPP/1  0
 TKR/0  0
IPPN/0  0
 BDG/0  0ARP

ARP cache meters
Net Prot  Max Cur Cnt    Alloc  Refresh: Tot  Failure  TMOs: Refresh
0   0     1   1   1       17            0       0             13
0  22     1   0   0        6            0       0              6
1   0     1   1   2       27            0       0             25
1  16     3   3   7      291            0       0              0
2   0     1   0   0        2            0       0              2
2  16     1   0   0        1            0       0              0
8   0     1   1   1       11            0       0             10
```

ARP input packet overflows Displays counters that represent the number of ARP packets discarded on input because the ARP layer was too busy. The counts shown are per network interface.

ARP cache meters Consists of a variety of meters on the operation of the ARP cache. The counts shown are all per protocol, per interface.

Net Displays the interface numbers.

Prot Displays the protocol numbers.

Max Displays the all-time maximum length hash chain.

Cur Displays the current maximum length hash chain.

Cnt Displays the count of entries currently active.

Alloc Displays the count of entries created.

Rfrsh:Tot Displays the number of refresh requests sent for this network interface and protocol.

Fail Displays the number of auto-refresh attempt failures due to unavailability of internal resources. This count is not related to whether or not an entry was refreshed.

TMOs:Rfrsh Displays the count of entries deleted due to a timeout of the refresh timer.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

ARP Over ATM Console Commands

This section summarizes and then explains the ARP over ATM (CIP) console commands. It describes the console commands for:

- Classical IP & ARP over ATM
- IPX over ATM

The console commands for IPX and ARP over ATM are essentially the same as those for Classical IP and ARP. The main difference is the format of protocol addresses:

- Protocol addresses for IP are specified as 4-byte fields in dotted decimal notation.
- Protocol addresses for IPX are specified as 6-byte fields in hexadecimal characters.

Note: The **ping** command for IPX over ATM is different from that used for Classical IP and ARP. The IPX version of the **ping** command is available at the IPX console. You can access ARP console commands at the `ARP>` prompt. Table 21-2 shows the commands.

For further information, refer to “Classical IP and ARP Over ATM Overview (RFC 1577)” on page 20-4 and “IPX and ARP Over ATM Overview (RFC 1483)” on page 20-10. For additional information on ARP over ATM, and for illustrations showing logical and physical network configurations, refer to *Configuration Program User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

Table 21-2. ARP Over ATM Console Command Summary

Command	Function
? (Help)	List the ARP console commands or list the options associated with specific commands.
Delete	Immediately bring down an active channel. A new channel may or may not be brought up to replace the old one depending on the conditions.
Display	Display all of the channels (VCCs) associated with a single ATM interface.
Dump	Show which ATM channels are being used for sending datagrams and show their corresponding IP addresses.
Hardware	List each ARP-configured network.
Ping	Verify connectivity between the device and the specified end station.
Protocol	List each ARP-configured protocol.
Statistics	Display statistics of the ARP code over all of the network interfaces.
Exit	Exit the ARP console process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: List output

Delete

Use the **delete** command to immediately bring down an active channel. A new channel may or may not be brought up to replace the old one depending on the conditions.

Delete a specific channel off of the Active Channel List. One should use great care when invoking this option. The channel specified by the VPI/VCI is deleted if it is found on the active channel list. Before deletion, the channel is released with a normal hang-up cause code. All ARP entries that are dependent on this particular channel are also deleted.

Syntax: delete

Example: **delete**

```
ARP> del 0
VPI, Range 00..FF [00]?
VCI, Range 0000..FFFF [0000]? 0020
Channel found and deleted
```

Display

Use the **display** command to display all of the channels (VCCs) associated with a single ATM interface.

Syntax: display

Example: **display**

```
ARP> display 0
Active Channel List : Net 0
   P/S FLAGS LIST  VPI/VCI    FwdPcr    FwdScr     MaxSDUsz Control P2P
  0) S   80    01   00/0020  155000000 155000000      9188      T     T
        Tgt Addr. 39.84.0F.00.00.00.00.00.00.00.00.00.02.10.00.5A.00.DE.AD.02
        Client Address (owner): 1.1.1.100
        Target Protocol Addresses: 1.1.1.2
New Channel List : Net 0
PVC Channel List : Net 0
   P/S FLAGS LIST  VPI/VCI    FwdPcr    FwdScr     MaxSDUsz Control P2P
  1) P   80    03   00/0085  155000000 155000000      9188      F     T
        Tgt Addr:
        Client Address (owner): 3.5.5.5
        Target Protocol Addresses: 3.4.4.4
```

P/S P means that this channel is a PVC. S means that this channel is an SVC.

List For internal use.

Flags For internal use.

VPI/VCI Virtual Path Identifier and Virtual Channel Identifier of the channel in use.

FwdPcr The Peak Cell Rate in bits per second.

FwdScr The Sustained Cell Rate in bits per second.

MaxSDUsz The maximum SDU size for this channel. All packets transmitted or received on this interface must be less than or equal to this size less the 8-byte header prefix used by RFC 1483.

Control T if this is a control channel (channel to the ARP server). F if this is a data channel (channel to another client).

P2P T if this channel is point-to-point. F if this channel is point-to-multipoint.

Active Channel List These channels are true connections with the remote party. Data can flow over these connections with the traffic parameters shown.

New Channel List These channels are in the process of being connected with the other end. No data may flow over them until they are moved to the active list.

PVC Channel List These are channels which have been specifically configured as PVCs. They take on the client characteristics for Data Channels as defined in the client configuration.

Dump

Use the **dump** command to show which ATM channels are being used for sending datagrams, and their corresponding IP addresses.

This table represents the entire ARP table for a physical ATM network running Classical IP. The hardware address is the resultant VCC identifier (VPI/VCI) for an active channel. That is, all traffic that is to be sent to the IP address will be transmitted out on the associated channel (listed under Hardware Address).

Note: If the host on the other end of the channel sends either a request or reply with its own address, we will automatically reset the refresh time to its maximum value.

Syntax: dump

Example: **dump**

```
ARP> dump 0
Hardware Address        IP Address                  Refresh
0x00/0x0020             1.1.1.2                     not aging
```

Under Refresh, the time specified is the approximate time before the ARP entry is aged out (in minutes). If autorefresh is turned on, then an ARP request or an InATMARP request will be sent out 30 seconds before the expiration. If a reply is received before expiration, the Refresh time is reset, and the ARP entry remains. If no reply is received, or if autorefresh is turned off, the ARP entry will be deleted when it expires. It will be recreated as required.

If “not aging” appears under Refresh, that entry will remain indefinitely.

Hardware

Use the **hardware** command to list all of the ATM addresses associated with each configured IP client.

Syntax: hardware

Example: **`hardware`**

```
ARP> hardware
    Network         Hardware AS     Hardware Address
    0 ATM/0         0013            39.84.0F.00.00.00.00.00.00.00.00.00.01.
                                    10.00.5A.00.DE.AD.C8  (IP 1.1.1.100)
    1 IPPN/0        0800            1.1.1.100
```

Network: The physical network number.

Hardware AS: The hardware type used in the ARP packets to classify this network. For ARP over ATM, the AS type is 0x13 (decimal 19).

Hardware Address: The hardware address. Typically, this address is a MAC address for other networks, but for ATM, this address is the ATM address associated with a specific client. In the example, the IP client, `1.1.1.100`, is accessed by calling the corresponding ATM address `39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.DE.AD.C8`.

Ping

Use the **ping** command to verify connectivity between the device and the specified end station.

Ping works exactly as it does over any of the other networks. It sends out an ICMP echo request every second, and displays statistics of the corresponding replies. Note that the source address in the request will contain the client's address that most closely matches the subnet of the destination.

Syntax: ping

Example: **`ping`**

```
    ARP> ping 1.1.1.2
    PING 1.1.1.100 -> 1.1.1.2: 56 data bytes, ttl=64, every 1 sec.
    56 data bytes from 1.1.1.2: icmp_seq=0. ttl=64. time=19. ms
    56 data bytes from 1.1.1.2: icmp_seq=1. ttl=64. time=11. ms

    ----1.1.1.2 PING Statistics----
    2 packets transmitted, 2 packets received, 0% packet loss
    round-trip min/avg/max = 11/15/19 ms
```

Protocol

Use the **protocol** command to list all the client addresses on each of the network interfaces. This is exactly the same as for other interfaces. For an ATM interface, the list of Protocol Addresses are all of the CIP clients configured on this interface.

Syntax: protocol

Example: **protocol**

```
    ARP> protocol
    Network  Protocol (num)  AS     Protocol Address(es)
    0 ATM/0  IP       ( 0)   0800   1.1.1.100
```

Statistics

Use the **statistics** command to display statistics of the ARP code over all of the network interfaces. These statistics are the same as the statistics in the ARP code over any of the other interfaces as described in “Statistics” on page 21-4.

Syntax: statistics

Example: **statistics**

```
ARP> statistics

    ARP input packet overflows
        Net   Count
       ATM/0  0
      IPPN/0  0
       BDG/0  0

    ARP cache meters
    Net Prot  Max Cur Cnt    Alloc  Refresh: Tot  Failure  TMOs: Refresh
      0   0     1   1   1        1             0        0              0
```

Chapter 22. Using, Configuring, and Monitoring XTP

This chapter describes the X.25 Transport Protocol (XTP) for transporting X.25 traffic over TCP/IP; it also provides information for configuring and monitoring XTP. Included are the following sections:

- "The X.25 Transport Protocol"
- "Configuring XTP" on page 22-5
- "Configuration Procedures" on page 22-5
- "XTP Configuration Commands" on page 22-13
- "XTP Monitoring Commands" on page 22-19

The X.25 Transport Protocol

X.25 Transport Protocol (XTP) provides you with the services of a "protocol forwarder." A protocol forwarder is the focal point for inbound and outbound protocol packet processing. Forwarders receive packets on one network interface and send them to another interface.

XTP is designed to work with X.25 devices that are situated at multiple remote sites. In such environments, XTP can eliminate the use of X.25 packet-switched networks for communicating with servers at one or more centralized locations.

To enable this, you use routers at the server and remote locations to encapsulate the data and deliver the X.25 packets between the clients and server via TCP/IP.

Figure 22-1 on page 22-2 illustrates a network configuration before and after using XTP.

Configuration Information

X.25 recognizes an incoming call for XTP based on the node addresses configured for XTP. Therefore, in order to transport X.25 traffic between the X.25 nodes, you must configure X.25 to map to the data terminal equipment (DTE) address and IP addresses of the routers to which the nodes are connected.

For example, in Figure 22-1 on page 22-2, you configure X.25 clients on remote routers and on the concentrator router. *Remote routers* in this example are the routers that connect the X.25 clients to the TCP/IP network that is used to access the X.25 server; the *concentrator router* connects the X.25 server to the TCP/IP network that is used to access the remote routers.

Note: When you configure XTP, if a router is connected to an X.25 switch, it is considered to be DTE. If it is not connected to a switch, it is considered to be DCE (Data Circuit-Terminating Equipment).

To configure a router for XTP, define the following information from the `XTP config>` prompt and then restart the router:

- Local DTEs
- Peer routers
- Remote DTEs
- PVCs

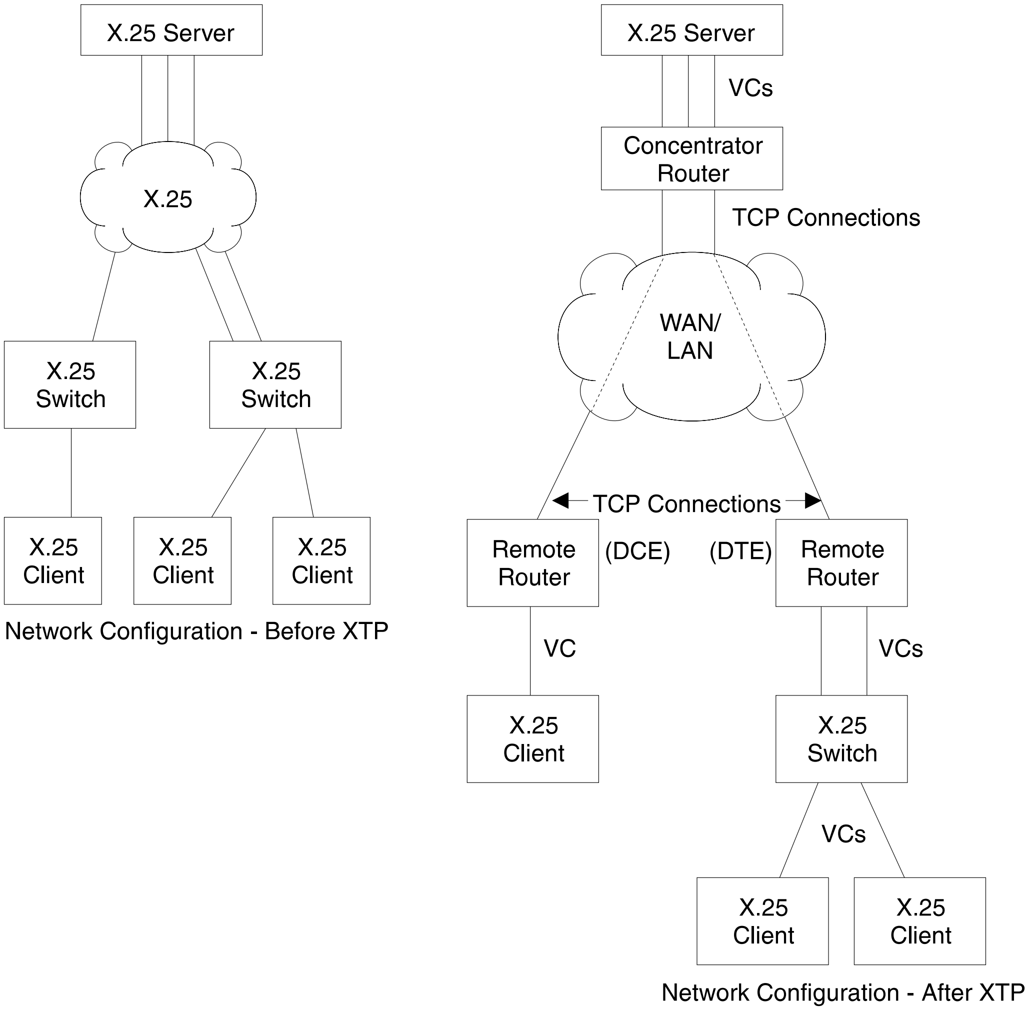


Figure 22-1. Configuration Before and After XTP

Local DTEs
X.25 nodes connected to the X.25 interfaces on the router

To configure local DTEs, use the X.121 address that is assigned to the local DTE. Multiple local DTEs can be configured on an interface.

Peer Routers
Routers with which you communicate over TCP/IP

Peer routers can differ depending on “point of view.” For example, in Figure 22-1, the *two remote routers* are the peer routers from the perspective of the concentrator router. However, the *concentrator router* is the peer router from the perspective of the two remote routers.

You designate the peer router by its internal IP address.

Remote DTEs
Remote X.25 nodes to which the local X.25 nodes open connections and exchange data. Use the X.121 address that is assigned to the remote DTE.

Configure a *unique* IP address for each peer router. For example, in Figure 22-1, the concentrator router must know the unique IP address of each remote router, and each remote router must know the IP address of the concentrator router.

PVC

A permanent channel that remains connected after X.25 restarts.

PVCs, because they are constant channels, are similar to leased telephone lines. A PVC, in the XTP context, is a PVC from a local X.25 DTE node to a remote X.25 DTE.

When you configure a router for PVCs, map the IP address of the peer router and the PVC number of the remote and local DTE. A PVC is identified by four pieces of information which are the:

- Logical channel number of the local PVC
- X.121 address of the local DTE
- Logical channel number of the PVC on the remote (peer) router
- X.121 address of the remote DTE

Additional configuration information can be found at “Configuring XTP” on page 22-5 and at “XTP Configuration Commands” on page 22-13.

DTE Address Wildcards

The “*” wildcard is available for DTE address configuration. This is in addition to the “?” character that can be specified in a DTE address to represent any one digit in that position in the address. For example, a specification of “1?2?3” can match address 18243 where the first, third, and fifth digits are 1, 2, and 3, respectively..

The “*” wildcard character can represent any string of zero or more digits. Its use is limited to the end of a DTE address specification. For example: “123*,” “5555*,” “9*” or “*.” The special case of a DTE address of “*” represents any DTE address, even a null address. The null address is useful for handling incoming calls with no calling address in the X.25 Call Request packet.

Use of the “*” wildcard increases the chances for adding a local or a remote DTE address that conflicts with an existing address. The **add local-dte** and **add remote-dte** commands are enhanced to provide the conflicting address when the user attempts to add a DTE address that conflicts with an existing address.

Example: `xtp config> add local-dte`

```
Interface number [0]? 1
DTE address [ ] 123456
DTE address [ ]?

XTP config>add local-dte
Interface number [0]?1
DTE address [ ]?1*
DTE address conflicts with existing DTE address 123456
```

XTP Backup Peer Function

The Backup Peer Function allows the association of multiple peer routers with a remote DTE. The user specifies a list of peer routers associated with a remote DTE.

Example:

```
XTP config>add rem
DTE address [ ]?123456
Peer router's internal IP Address [0.0.0.0]?10.0.0.2
Peer router's internal IP Address [0.0.0.0]?10.0.0.4
Peer router's internal IP Address [0.0.0.0]?11.0.0.1
Peer router's internal IP Address [0.0.0.0]?
```

When an incoming call for the remote DTE is received, a connection is attempted through each router in the list in the same order that they appear for the remote DTE.

Searching for a Remote DTE

When a DTE initiates a call for a remote DTE, both DTE addresses are inspected to determine if they are acceptable for X.25 transport. If they are acceptable, the X.25 Transport protocol forwarder determines through which peer router to attempt to complete the call. It starts with the first router in the remote DTE's list of peer routers in its search. The first condition that must be met is an active TCP connection to the peer router. If there is not an active TCP connection to the peer, the next router in the list is checked. When an active TCP connection is found, an attempt is made to complete the call. The Connection Request Timer is started to time the call connection process.

The remote DTE search is terminated by one of the following events:

- Successful completion of the call through the peer router

  This completes call setup processing and ends the search for the remote DTE.

- Rejection of the call by the peer router

  This causes the search for the remote DTE to proceed to the next router in the peer router list.

- Expiration of the Connection Request Timer

  This causes the search for the remote DTE to proceed to the next router in the peer router list.

If a pass through the list of peer routers is completed without a successful connection through any of the peer routers, the call to the local DTE is cleared.

Connection Request Timer

The Connection Request Timer is used to ensure that no call setup procedure hangs for an indeterminable time. There is a timer configured for each peer router.

Example:

```
XTP config>add peer-router
Router's internal IP Address [0.0.0.0]?10.0.0.2
Connection setup timeout [230]?60
```

The Connection Request Timer can be configured from 10 to 480 seconds. The default is 230 seconds. This default was determined based on the fact that the default setting for the X.25 Call Request Timer is 200 seconds.

The timer is started when an attempt is made to complete a call through a peer router. It is stopped when the call attempt is either accepted or rejected by the peer router.

Configuring XTP

XTP is a protocol forwarder used to transport X.25 traffic over TCP/IP. XTP allows existing X.25 devices to communicate over a TCP/IP backbone and migrate from an X.25 network to a network of your choice.

Configuration Procedures

This section defines the detail for configuring the network displayed in Figure 22-2.

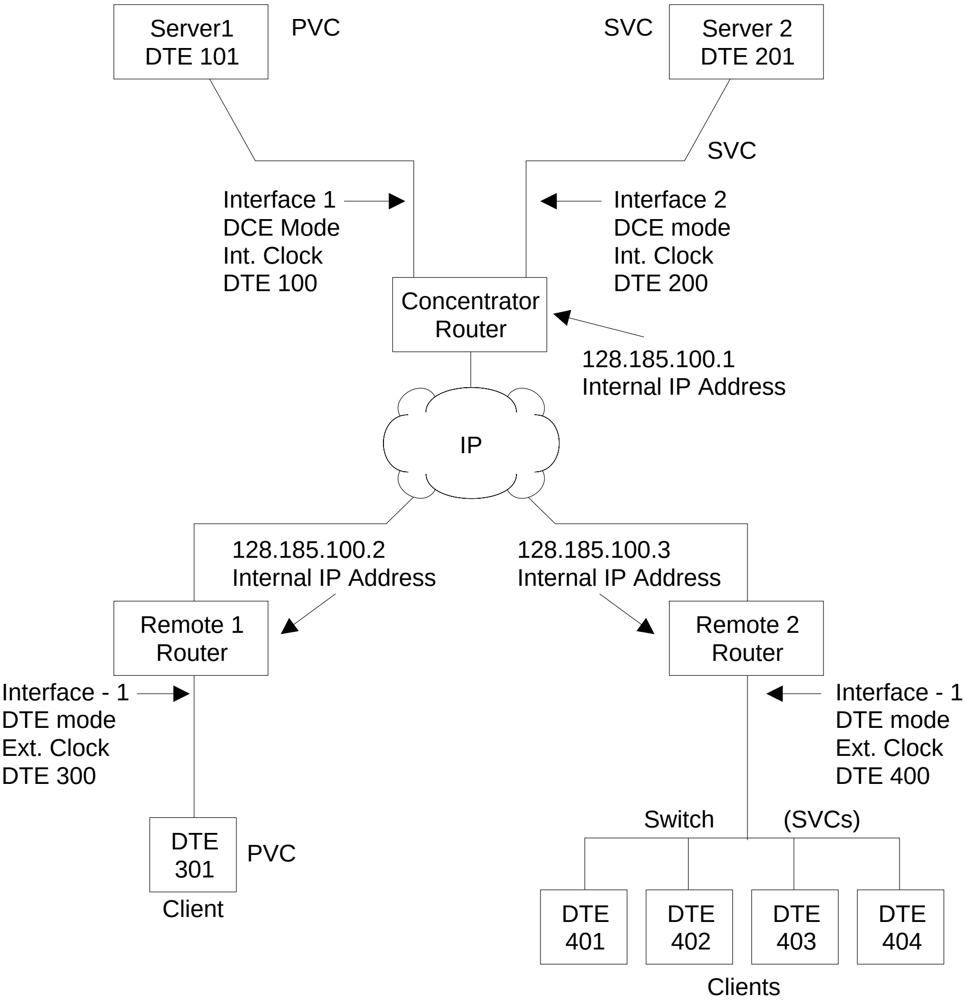


Figure 22-2. Sample XTP Configuration

This configuration shows three routers, the Concentrator router, Remote 1 router, and Remote 2 router. To make XTP operational on this network, perform the following steps for each of these routers:

- Set the data link
- Configure the IP interface
- Configure X.25
- Set the National Personality values
- Define the IP address
- Set the Internal IP address
- Configure XTP

Note: New configurations do not take effect until you restart the router.

Setting the Data Link

The data link defines the protocol you are using to send data packets over the network. Define the data link between the router you are configuring and each serial interface. The example in Figure 22-2 on page 22-5 configures a router with three serial interfaces, two for X.25 and one for PPP.

Set the data-link protocol for the serial interfaces:

```
Config>set data-link X25 1
Config>set data-link x25 2
Config>set data-link ppp 3
```

Configuring the IP Interface

In Figure 22-2 on page 22-5, the IP interface is PPP; enter **network 3** at the `Config>` prompt to configure this PPP interface:

```
Config>network 3
PPP interface configuration
```

Note: This procedure does not include details about the configuration of PPP. For details, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

Configuring X.25

Before configuring XTP, configure the X.25 parameters for each interface. The following example configures the basic parameters for X.25 and is based on the topology in Figure 22-2 on page 22-5.

The parameters you need to configure depend on your network topology. For details about all the X.25 parameters, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

Interface 1

Use the following instructions to configure *Interface 1* on the concentrator router as defined in Figure 22-2 on page 22-5.

1. At the `Config>` prompt, enter **`network`** followed by the number of the X.25 interface. In this example, it is interface 1.

```
Config>network 1
X.25 User Configuration
X.25 Config>
```

2. Add the XTP protocol to the X.25 interface and define general interface values. Enter **add protocol xtp** at the X.25 Config> prompt. This command needs to be entered *one time only*.

   ```
   X.25 Config>add protocol xtp
   Window Size [2]?
   Default Packet Size [128]?
   Maximum Packet Size [256]?
   ```

3. Specify the network address by entering **set address** X.25 node address. In Figure 22-2 on page 22-5, the node address (DTE address) is 100.

   ```
   X.25 Config>set address 100
   ```

4. Enter **set clocking** followed by **internal** or **external** based on your router type.

   ```
   X.25 Config>set clocking internal
   ```

5. Enter **set speed** followed by the access rate (line speed).

   ```
   X.25 Config>set speed
   Access rate in bps [9600]?19200
   ```

6. Enter **set equipment-type** and specify whether the frame and packet levels act as DCE or DTE.

   ```
   X.25 Config>set equipment-type dce
   ```

7. Enter **set pvc** and define the lowest and the highest PVC you are using.

   ```
   X.25 Config>set pvc low 1
   X.25 Config>set pvc high 1
   ```

8. Enter **add pvc** to define individual PVCs.

   ```
   X.25 Config>add pvc
   Protocol [IP]?xtp
   Packet Channel [1]?
   Destination X.25 Address [ ]?101
   Window Size [2]?
   Packet Size [128]
   ```

Interface 2

Use the following instructions to configure interface 2.

1. At the Config> prompt, enter **network** followed by the number of the X.25 interface. In Figure 22-2 on page 22-5, it is 2.

   ```
   Config>network 2
   X.25 User Configuration
   X.25 Config>
   ```

2. Use the same procedures as defined in “Interface 1” on page 22-6 to set the following parameters for interface 2:

   - address = 200
   - clocking = internal
   - speed = 19200
   - equipment = dce

3. Enter **set svc** and define the lowest and highest SVC you are using.

   There are three types of SVCs: two-way, inbound and outbound. The defaults are “svc low-two-way = 1” and “svc high-two-way = 64.” All other SVC types default to 0. For additional information on SVCs and PVCs, refer to *Software User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

```
X.25 Config>set svc ?
X.25 Config>set svc low-inbound 0
X.25 Config>set svc high-inbound 0
X.25 Config>set svc low outbound 0
X.25 Config>set svc high outbound 0
X.25 Config>set svc low-two-way 2
X.25 Config>set svc high-two-way 2
```

4. Exit the `X.25 Config>` prompt.

```
X25 Config>exit
Config>
```

Setting the National Personality

Each X.25 public network has its own standard configuration. The National Personality refers to a group of 28 variables that define the characteristics of the public data network. These variables provide the router with control information for packets transferred over the link and influence the X.25 facilities used between and XTP router and its local DTE.

All facilities contained in incoming call requests are passed on to the peer router, regardless of whether the local router was configured to support that facility. For example, when packet size negotiation is requested in the incoming call and flow control negotiation is not configured in the router.

The router will insure any packet size and window size being negotiated is within the range specified when defining the X.25 interface. For example, a packet window greater than 7 is negotiated down to 7 if packet-ext-seq-mode has not been defined for the X.25 interface.

To view the configuration values, enter **list detailed** at the `X.25 Config>` prompt. To set the default values for the national personality, enter **set national-personality** at the `X.25 Config>` prompt. For further information, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

Defining the IP Address

Before you configure the Concentrator router (as displayed in Figure 22-2 on page 22-5) for XTP, define the IP address for this router. Enter **protocol ip** at the `Config>` prompt and enter **add address** at the `IP config>` prompt.

```
Config>protocol ip
IP config>add address
Which net is this address for [0]?3
New address [0.0.0.0]?128.185.100.7
Address mask [255.255.0.0]?255.255.255.0
```

Setting the Internal IP Address

Each router identifies its peer routers by the internal IP address of the peer routers.

To set the internal IP address of the peer router, enter **set internal IP address** at the `IP Config>` prompt.

```
IP config>set internal-ip-address
Internal IP address [0.0.0.0]?128.185.100.1
```

Configuring XTP

After you have configured X.25 and defined the IP address, you are ready to configure XTP for the router.

If you need further configuration information when configuring XTP, see “XTP Configuration Commands” on page 22-13.

Note: When configuring your network for XTP, remember that the peer routers are always the routers you are communicating with over TCP/IP. Therefore, the peer router can differ depending on the point of view. When configuring the routers defined as Remote 1 router and Remote 2 router in Figure 22-2 on page 22-5, to them the peer router is the Concentrator router.

Implement the following steps to configure XTP for the router:

1. To access the `XTP config>` prompt, enter **protocol xtp** at the `Config>` prompt.
2. Add interface 1 to the XTP configuration. Enter **add local-dte** at the `XTP Config>` prompt.

   ```
   XTP config>add local-dte
   Interface number [0]?1
   DTE address [ ]?101
   DTE address [ ]?
   ```

   Entering a null DTE address ends the command input.
3. Add interface 2 to the XTP configuration. Enter **add local-dte** at the `XTP Config>` prompt.

   ```
   XTP config>add local-dte
   Interface number [0]?2
   DTE address [ ]?201
   DTE address [ ]?
   ```

   Entering a null DTE address ends the command input.
4. Add Remote 1 router as the peer router. Enter **add peer-router** and enter the IP address of this router.

   ```
   XTP config>add peer-router
   Router's internal IP Address [0.0.0.0]?128.185.100.2
   Connection setup timeout [230]?
   ```
5. Add the remote DTE for Remote 1 router. Enter **add remote-dte** and enter the IP and DTE address of this DTE.

   ```
   XTP config>add remote-dte
   DTE address [ ]?301
   Peer router's internal IP Address [0.0.0.0]?128.185.100.2
   Peer router's internal IP Address [0.0.0.0]?
   ```

   Note:

   A remote DTE is *required* only if one of the following applies:

   - The Concentrator Router will be initiating XTP connections to the remote DTE due to incoming calls from its local DTEs.
   - The DTE is part of an XTP PVC definition.
6. Add Remote 2 router (as the peer router). Enter **add peer-router** and enter the IP address of this router.

   ```
   XTP config>add peer-router
   Router's internal IP Address [0.0.0.0]?128.185.100.3
   Connection setup timeout [230]?
   ```

7. Add the remote DTEs for Remote 2 router. Enter add remote-dte and enter the IP and DTE addresses of this DTE.

```
XTP config>add remote-dte
DTE address [ ]?401
Peer router's internal IP Address [0.0.0.0]?128.185.100.3
Peer router's internal IP Address [0.0.0.0]?

XTP config>add remote-dte
DTE address [ ]?402
Peer router's internal IP Address [0.0.0.0]?128.185.100.3
Peer router's internal IP Address [0.0.0.0]?

XTP config>add remote-dte
DTE address [ ]?403
Peer router's internal IP Address [0.0.0.0]?128.185.100.3
Peer router's internal IP Address [0.0.0.0]?

XTP config>add remote-dte
DTE address [ ]?404
Peer router's internal IP Address [0.0.0.0]?128.185.100.3
Peer router's internal IP Address [0.0.0.0]?
```

8. Add an XTP PVC to logically associate the local PVC to Server 1 with the remote DTE 301.

```
XTP config>add pvc
Local PVC number [1]? 1
Local X.25 DTE address [ ]? 101
Remote PVC number [1]? 1
Remote X.25 DTE address [ ]?301
```

When entering DTE addresses, you can specify either of the following:

A '?' in place of any digit. The '?' means any single digit in this digit position.

An '*' as the last digit of an address to represent any combination of zero or more digits.

Sample Configuration of Remote Routers

The following is a sample configuration of Remote 1 router and Remote 2 router (see Figure 22-2 on page 22-5). The process is the same as that defined in the section at “Configuration Procedures” on page 22-5.

Remote 1 router

```
*talk 6

Config>set data-link x25 1
Config>set data-link ppp 2
Config>network 1

X.25 Config>set address 300
X.25 Config>set clocking internal
X.25 Config>set speed 19200
X.25 Config>set equipment-type dce
X.25 Config>set pvc low 1
X.25 Config>set pvc high 1
X.25 Config>add pvc
Protocol [IP]?xtp
Packet Channel [1]?1
Destination X.25 Address [ ]?301

Window Size [2]?
Packet Size [128]?
X.25 Config>exit
Config>

Config>protocol ip
IP config>add address
Which net is this address for [0]?2
New address [0.0.0.0]?128.185.100.8
Address mask [255.255.0.0]?255.255.255.0

IP config>set internal-ip-address
Internal IP address [0.0.0.0]?128.185.100.2
IP Config>exit
Config>

Config>protocol xtp
XTP config>add local-dte
Interface number [0]?1
DTE address [ ]?301
DTE address [ ]?

XTP config>add peer-router
Router's IP address?128.185.100.1

XTP config>add remote-dte
DTE address [ ]?101
Peer router's internal IP Address ]0.0.0.0]?128.185.100.1
Peer router's internal IP Address [0.0.0.0]?

XTP config>add pvc
Local PVC number [1]? 1
Local X.25 DTE address [ ]? 101
Remote PVC number [1]? 1
Remote X.25 DTE address [ ]?301
```

Remote 2 router

```
*talk 6

Config>set data-link x25 1
Config>set data-link ppp 2
Config>network 1

X.25 Config>set address 400
X.25 Config>set clocking external
X.25 Config>set speed 19200
X.25 Config>set equipment-type dte
X.25 Config>set svc low-inbound 0
X.25 Config>set svc high-inbound 0
X.25 Config>set svc low-outbound 0
X.25 Config>set svc high-outbound 0
X.25 Config>set svc low-two-way 1
X.25 Config>set svc high-two-way 64
X.25 Config>add protocol
Protocol [IP]?xtp
Window Size [2]?
Default Packet Size [128]?
Maximum Packet Size [256]?
X.25 Config>exit

Config>protocol ip
IP config>add address
Which net is this address for [0]?2
New address [0.0.0.0]?128.185.100.9
Address mask [255.255.0.0]?255.255.255.0

IP config>set internal-ip-address
Internal IP address [0.0.0.0]?128.185.100.3
IP Config>exit
Config>

Config>protocol xtp
XTP config>add local-dte
Interface number [0]?1
DTE address [ ]?401
DTE address [ ]?402
DTE address [ ]?403
DTE address [ ]?404
DTE address [ ]?

XTP Config>add peer-router
Router's IP address?128.185.100.1

XTP config>add remote-dte
DTE address [ ]?201
Peer router's internal IP Address [0.0.0.0]?128.185.100.1
Peer router's internal IP Address [0.0.0.0]?
XTP config>exit

Config>
```

XTP Configuration Commands

This section summarizes and explains the XTP configuration commands.

To access the XTP configuration environment, enter the **protocol xtp** command at the `Config>` prompt.

```
Config> xtp
XTP config>
```

Enter XTP configuration commands at the `XTP config>` prompt.

Table 22-1. XTP Configuration Commands Summary

Command	Function
? (Help)	Lists available commands or lists the options associated with specific commands.
Add	Adds an interface, peer router, remote DTE or PVC definitions.
Change	Changes a peer router, remote DTE or PVC definition.
Delete	Deletes a local DTE, peer router, remote DTE or PVC definition.
Enable-XTP	Activates the XTP forwarder.
Disable-XTP	Deactivates the XTP forwarder.
Set	Sets the value of the XTP Keepalive Timer.
List	Lists interfaces, peer routers, remote DTEs and PVC definitions.
Exit	Returns to the `Config>` prompt.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You also can enter a **?** after a command to list its options.

Syntax: ?

Example: **?**

```
ADD
CHANGE
DELETE
DISABLE-XTP
ENABLE-XTP
LIST
EXIT
```

Add

Adds a local X.25 node, a peer router, a remote X.25 node with corresponding routers, or a PVC from a local X.25 node to a remote X.25 node.

Wild card addressing is included in the XTP forwarder. When the local or remote DTE addresses are entered, they can contain a wild card character (? or *). For additional information on the use of wildcards, see “DTE Address Wildcards” on page 22-3.

Syntax: add local-dte
peer-router
remote-dte

pvc

local-dte

Adds the X.25 DTE addresses, or the X.25 nodes, that communicate with the router on the specified interface.

You can configure multiple local nodes.

Example: `add local-dte`

```
Interface number [0]?1
DTE address [ ]?101
DTE address [ ]?
```

peer-router

Adds peer routers. Enter the internal IP addresses of the routers to which the remote X.25 nodes are connected. You can use these IP addresses to open TCP connections and transport X.25 packets that contain connection requests and X.25 data.

Example: `add peer-router`

```
Router's internal IP Address [0.0.0.0]?128.185.100.2
Connection setup timeout [230]?
```

remote-dte

Adds remote X.25 nodes and corresponding routers. You can connect remote nodes with local X.25 nodes so they can exchange data. You must configure an IP address for each remote X.25 node you configure. Any request or data sent to this remote node goes to the router. The router then uses one of its local X.25 interfaces to forward the data to the X.25 node.

Define a remote DTE if this router is to initiate XTP connections to the remote DTE due to incoming calls from its local DTEs, or if the remote DTE is part of an XTP PVC definition.

Example: `add remote-dte`

```
DTE address [ ]?301
Peer router's internal IP Address [0.0.0.0]?128.185.100.2
Peer router's internal IP Address [0.0.0.0]?
```

pvc

Adds a PVC from a local X.25 node to a remote X.25 node.

Three things need to exist in order to activate a PVC configuration:

- An X.25 PVC from the router to the local X.25 node
- An X.25 PVC from the peer router to the remote X.25 node
- A TCP connection to the peer router where the remote node is resident

Example: `add pvc`

```
Local PVC Number [1]?1
Local X.25 DTE address [ ]?100
Remote PVC Number [1]?1
Remote X.25 DTE address [ ]?301
```

Note: When you add PVCs to the router configuration, you also must configure the PVC in X.25. For details on configuring X.25 interfaces, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

Change

Changes a peer router, remote DTE, or PVC from the XTP configuration.

```
Syntax:  change   peer-router
                  remote-dte
                  pvc
```

peer-router

Changes specific peer routers from the XTP configuration.

Example: `change peer-router`

```
Router IP Address [0.0.0.0]?128.185.100.2
```

remote-dte

Changes specific remote dtes from the XTP configuration.

Example: `change remote-dte`

```
DTE address [ ]?401
Peer router's internal IP Address [0.0.0.0]?128.185.100.2
Peer router's internal IP Address [0.0.0.0]?
```

pvc

Changes PVC definitions from the XTP configuration.

Example: `change pvc`

```
Local PVC number [1]?1
Local DTE address [ ]?301
```

Delete

Deletes a local DTE, peer router, remote DTE, or PVC from the XTP configuration.

```
Syntax:  delete   local-dte
                  peer-router
                  remote-dte
                  pvc
```

local-dte

Deletes specific local interfaces from the XTP configuration.

Example: `delete local-dte`

```
Interface number [0]?1
DTE address [ ]?101
Record deleted
```

peer-router

Deletes specific peer routers from the XTP configuration.

Example: `delete peer-router`

```
Router IP Address [0.0.0.0]?128.185.100.2
Record deleted
```

remote-dte

Deletes specific remote dtes from the XTP configuration.

Example: `delete remote-dte`

```
DTE address [ ]?401
```

pvc

Deletes PVC definitions from the XTP configuration.

Example: **delete pvc**

```
Local PVC number [1]?1
Local DTE address [ ]?301
Record deleted
```

Enable

Activates the XTP forwarder.

Syntax: enable-xtp

Example: **enable-xtp**

Disable

Deactivates the XTP forwarder.

Syntax: disable-xtp

Example: **disable-xtp**

Set

Sets the XTP Keepalive Timer.

Syntax: keep-alive-timer

Example: **set keep-alive-timer**

```
Keepalive timer in seconds [10]?60
```

List

Lists the interfaces, peer routers, remote DTEs, or PVCs.

Syntax: list all
xtp-status
local-dtes
peer-routers
remote-dtes
pvcs

all

Displays all the interfaces, peer routers, remote DTEs, and PVCs configured for XTP.

Example: **list all**

```
STATUS:  XTP-ENABLED
Local DTEs:
Interface        DTE Addr
   1             101
   2             201

Peer Routers:
128.185.100.2
128.185.100.3


Remote DTEs:
DTE Address      Peer Router
    301         128.185.100.2
    401         128.185.100.3
    402         128.185.100.3
    403         128.185.100.3
    404         128.185.100.3

PVCs:
Local PVC        Local DTE      Remote PVC      Remote DTE
 Number            Address        Number          Address
  1                  100              1             301
```

xtp-status

Displays the status of XTP indicating whether it is enabled or disabled.

Example: **`list xtp-status`**

```
STATUS:  XTP-ENABLED
```

peer-routers

Displays all the peer routers configured for XTP.

Example: **`list peer-routers`**

```
Peer Routers:
128.185.100.2
128.185.100.3
```

remote-dtes

Displays all the remote DTEs configured for XTP.

Example: **`list remote-dtes`**

```
Remote DTEs:
DTE Address      Peer Router
    301         128.185.100.2
    401         128.185.100.3
    402         128.185.100.3
    403         128.185.100.3
    404         128.185.100.3
```

pvcs

Displays all the PVCs configured for XTP.

Example- **`list pvcs`**

```
PVCs:

Local PVC      Local DTE      Remote PVC      Remote DTE
 Number          Address        Number          Address
  1                100              1             301
```

Exit

Returns the `Config>` prompt.

Syntax: exit

Example: **`exit`**

XTP Monitoring Commands

This section summarizes and then describes the XTP monitoring commands. These commands allow you to display the current active interfaces, peer routers, remote DTE, PVCs and SVCs. They also allow you to dynamically add or delete interfaces, DTEs, or peer routers.

To display the XTP> prompt, enter **protocol xtp** at the monitoring (+) prompt:

```
+protocol xtp
X.25 Transport Console
XTP>
```

Enter the XTP monitoring commands at the XTP> prompt.

Table 22-2. XTP Monitoring Commands Summary

Command	Function
? (Help)	Lists available commands or lists the options associated with specific commands.
Add	Dynamically adds local DTEs, remote DTEs, or peer routers
Delete	Dynamically deletes configurations for local DTEs, remote DTEs, or peer routers
List	Displays individual PVC or SVC statistics and general information
Exit	Returns to the monitoring +> prompt

? (Help)

Lists the commands that are available from the current prompt. You also can enter a **?** after a specific command to list its options.

Syntax: ?

Example: **?**

```
ADD
DELETE
LIST
EXIT
```

Add

Adds an interface, peer router, or remote DTE to the XTP configuration.

Syntax: add local-dtes
peer-router
remote-dtes

local-dtes

Adds a local interface to the XTP configuration.

Example: **add local-dtes**

```
Interface number [0]?1
DTE address [ ]?101
```

peer-router

Adds a peer router to the XTP configuration.

Example: **add peer-router**

```
Router's IP Address [0.0.0.0]?128.185.100.2
```

remote-dtes

Adds a remote DTE to the XTP configuration.

Example: **add remote-dtes**

```
Peer router's IP Address [0.0.0.0]?128.185.100.2
DTE address [ ]?301
DTE address [ ]?
```

Delete

Deletes a local DTE, peer router, or remote DTE from the router configuration.

Syntax: delete local-dtes
peer-router
remote-dtes

local-dtes

Deletes a local interface from the XTP configuration.

Example: **delete local-dtes**

```
Interface Number [0]?1
DTE address [ ]?101
DTE address [ ]?
```

peer-router

Deletes a peer router from the XTP configuration.

Example: **delete peer-router**

```
Router's IP Address [0.0.0.0]?123.185.100.2
```

remote-dtes

Deletes a remote DTE from the XTP configuration.

Example: **delete remote-dtes**

```
DTE address [ ]?401
DTE address [ ]?
```

List

Displays the current active interfaces, peer routers, remote DTEs, PVCs, and SVCs.

Syntax: list all
xtp-status
local-dtes
peer-routers
remote-dtes
pvcs
pvc-detailed
pvcs-all-detailed
svcs
svc-detailed
svc-all-detailed

all

Displays output of all list command options.

example: `list all`

```
STATUS:  XTP-ENABLED
KEEP-ALIVE TIMER = 20 seconds


LIST OF LOCAL DTES
------------------
Interface     Local
  No           DTE
  1            101
  2            201


LIST OF PEER ROUTERS
--------------------
 Router            CNN      Number       Received           Sent
                  State    of  Ckts     Pkts    Bytes     Pkts    Bytes
128.185.100.3     Active      15         60      1533      12      142
128.185.100.2     Active      12         63      1620      10      130


LIST OF REMOTE DTES
-------------------
      Remote         Router
       DTE             IP
       404           128.185.100.3
       403           128.185.100.3
       402           128.185.100.3
       401           128.185.100.3
       301           128.185.100.2


LIST OF PVCS
------------
Index     Int    PVC       Local      Remote
 No        No   State       DET         DTE
  1         1   Active      100         301


LIST OF SVCS   (list svcs)
------------
Index   Int   Logical    SVC           Local               Remote
  No     No   Channel   State           DTE                  DTE
  1       2       5      ACT        333333333333        4444444444444


SVC  1 IN DETAIL   (list svc-detailed)
----------------
Int    Log     SVC       Received           Sent             Dropped
 No    Chn    State      Pkts   Bytes      Pkts   Bytes      Pkts   Bytes

  2      5     ACT          2     116         2     106         0       0


LIST OF SVCS   (svcs-all-detailed)
------------
Int    Log     SVC       Received           Sent             Dropped
 No    Chn    State      Pkts   Bytes      Pkts   Bytes      Pkts   Bytes

  2      5     ACT          1       7         1       2         0       0
```

xmp-status

Displays whether XTP is enabled/disabled, and the time specified for the Keepalive Timer.

Example: `list xtp-status`

```
            STATUS:  XTP-ENABLED
                     KEEP-ALIVE-TIMER = 20 seconds
```

local-dtes

Displays all the interfaces configured for XTP.

Example: **`list local-dtes`**

```
LIST OF LOCAL DTES
------------------
Interface     Local
  No           DTE
  1            101
  2            201
```

peer-routers

Displays all the peer routers configured for XTP.

Example: **`list peer-routers`**

```
LIST OF PEER ROUTERS
--------------------
 Router             CNN      Number        Received           Sent
                   State    of  Ckts     Pkts    Bytes     Pkts    Bytes
128.185.100.3     Active      15          60      1533      12      142
128.185.100.2     Active      12          63      1620      10      130
```

remote-dtes

Displays all the remote interfaces configured for XTP.

Example: **`list remote-dtes`**

```
LIST OF REMOTE DTES
-------------------
      Remote         Router
       DTE             IP
       404          128.185.100.3
       403          128.185.100.3
       402          128.185.100.3
       401          128.185.100.3
       301          128.185.100.2
```

pvcs

Displays all the PVCs configured for XTP.

Example: **`list pvcs`**

```
LIST OF PVCS
------------
Index     Int    PVC       Local      Remote
 No        No   State       DET         DTE
  1         1   Active      100         301
```

pvc-detailed

Displays detailed information for a specific PVC definition. For a listing of Index numbers, enter **list all** at the `xtp>` prompt.

Example: **`list pvc-detailed`**

```
PVC Index Number [1]?1

PVC  1 IN DETAIL
----------------
Int     PVC        Received           Sent             Dropped
 No    State      Pkts     Bytes    Pkts     Bytes    Pkts      Bytes
 1     ACTIVE      55       3220     35       2350     15        1870
```

pvcs-all-detailed

Displays detailed information for all PVC definitions.

Example: **list pvcs-all-detailed**

```
LIST OF PVCS
------------
INT     PVC         Received            Sent            Dropped
 No    State      Pkts     Bytes    Pkts     Bytes   Pkts     Bytes
 1    ACTIVE      55       3220     35       2350    15       1870
```

svcs

Displays all the SVCs definitions.

Example: **list svcs**

```
LIST OF SVCS
------------
Index     Int    SVC       Local      Remote
 No        No   State       DTE         DTE
  1         1   Active      200         401
  2         1   Active      200         402
  3         2   Active      200         403
  4         2   Active      200         404
```

svc-detailed

Displays information for specific SVC definitions.

Example: **list svc-detailed**

```
SVC Index Number [1]?1
SVC  1 IN DETAIL
----------------
Int     SVC        Received            Sent            Dropped
 No    State     Pkts     Bytes    Pkts     Bytes   Pkts      Bytes
 1    ACTIVE     75       4220     55       3350    20        870
```

svcs-all-detailed

Displays information for all the SVC definitions.

Example: **list svcs-all-detailed**

```
LIST OF SVCS
------------
Index     Int    SVC         Received          Sent           Dropped
No        No    State      Pkts    Bytes   Pkts     Bytes   Pkts    Bytes
1          1    ACTIVE     4220     55          3550        20      870
2          1    ACTIVE     3220     40          2350        15      970
3          2    ACTIVE     4003     50          3892        20      870
4          2    ACTIVE     3967     58          4167        12      800
```

Exit

Returns the command line to the monitoring (+) prompt.

Syntax: exit

Example: **exit**

Chapter 23. Using and Configuring OSPF

This chapter describes how to use the Open Shortest Path First (OSPF) Protocol, which is an Interior Gateway Protocol (IGP). The router supports the following IGPs for building the IP routing table, Open Shortest Path First (OSPF) Protocol and RIP Protocol. OSPF is based on link-state technology or the shortest-path first (SPF) algorithm. RIP is based on the Bellman-Ford or the distance-vector algorithm.

Included in this chapter are the following sections:

- “The OSPF Routing Protocol”
- “Configuring OSPF” on page 23-4
- “Accessing the OSPF Configuration Environment” on page 23-17
- “OSPF Configuration Commands” on page 23-17.
- “Multicast Forwarding” on page 23-11

Routers that use a common routing protocol form an *autonomous system* (AS). This common routing protocol is called an Interior Gateway Protocol (IGP). IGPs dynamically detect network reachability and routing information within an AS and use this information to build the IP routing table. IGPs can also import external routing information into the AS. The router can simultaneously run OSPF and RIP. When it does, OSPF routes are preferred. In general, use of the OSPF protocol is recommended due to its robustness, responsiveness, and decreased bandwidth requirements.

The OSPF Routing Protocol

The router supports a complete implementation of the OSPF routing protocol, as specified in RFC 1583 (Version 2). This version is incompatible with bridging routers running OSPF Version 1. OSPF information will not be exchanged between routers running Version 1 and Version 2.

OSPF is a link-state dynamic routing protocol that detects and learns the best routes to (reachable) destinations. OSPF can quickly perceive changes in the topology of an AS, and after a short convergence period, calculate new routes. The OSPF protocol does not encapsulate IP packets, but forwards them based on destination address only.

OSPF Routing Summary

When a router is initialized, it uses the Hello Protocol to send hello packets to its neighbors, and they in turn send their packets to the router. On broadcast and point-to-point networks, the router dynamically detects its neighboring routers by sending the Hello packets to the multicast address *ALLSPFRouters*; on non-broadcast networks you must configure information to help the router discover its *neighbors*. On all multi-access networks (broadcast and non-broadcast), the Hello Protocol also elects a *designated router* for the network.

The router then attempts to form adjacencies with its neighbors to synchronize their topological databases. Adjacencies control the distribution (sending and receiving) of the routing protocol packets as well as the distribution of the topological database updates. On a multi-access network, the designated router determines which routers become adjacent.

A router periodically advertises its status or link state to its adjacencies. *Link state advertisements* (LSAs) flood throughout an area ensuring that all routers have exactly the same topological database. This database is a collection of the link state advertisements received from each router belonging to an area. From the information in this database, each router can calculate a shortest path tree with itself designated as the root. Then the shortest path tree generates the routing table.

OSPF is designed to provide services not available with RIP. OSPF includes the following features:

- *Least-Cost Routing.* Allows you to configure path costs based on any combination of network parameters. For example, bandwidth, delay, and dollar cost.
- *No limitations to the routing metric.* While RIP restricts the routing metric to 16 hops, OSPF has no restriction.
- *Multipath Routing.* Allows you to use multiple paths of equal cost that connect the same points. You can then use these paths for load distribution that results in more efficient use of network bandwidth.
- *Area Routing.* Decreases the resources (memory and network bandwidth) consumed by the protocol and provides an additional level of routing protection.
- *Variable-Length Subnet Masks.* Allow you to break an IP address into variable-size subnets, conserving IP address space.
- *Routing Authentication.* Provides additional routing security.

OSPF supports the following physical network types:

- *Point-to-Point.* Networks that use a communication line to join a single pair of routers. A 56-Kbps serial line that connects two routers is an example of a point-to-point network.
- *Broadcast.* Networks that support more than two attached routers and are capable of addressing a single physical message to all attached routers. A token-ring network is an example of a broadcast network.
- *Non-Broadcast.* Networks that support more than two attached routers but have no broadcast capabilities. An X.25 Public Data Network is an example of a non-broadcast network. For OSPF to function properly, this network requires extra configuration information about other OSPF routers attached to the non-broadcast network.

Designated Router

Every multi-access network has a designated router that performs two main functions for the routing protocol: it originates network link advertisements and it becomes adjacent to all other routers on the network.

When a designated router originates network link advertisements, it lists all the routers, including itself, currently attached to the network. The link ID for this advertisement is the IP interface address of the designated router. By using the subnet/network mask, the designated router obtains the IP network number.

The designated router becomes adjacent to all other routers and is tasked with synchronizing the link state databases on the broadcast network.

The Hello Protocol elects the designated router after determining the router's priority from the *Rtr Pri* field of the hello packet. When a router's interface first becomes functional, it checks to see if the network currently has a designated router. If it does, it accepts that designated router regardless of that router's priority, otherwise, it declares itself the designated router. If the router declares itself the designated router at the same time that another router does, the router with higher router priority (*Rtr Pri*) becomes the designated router. If both *Rtr Pris* are equal, the one with the higher router ID is elected.

Once the designated router is elected, it becomes the endpoint for many adjacencies. On a broadcast network, this optimizes the flooding procedure by allowing the designated route to multicast its Link State Update packets to the address ALLSPFRouters rather than sending separate packets over each adjacency.

Multicast OSPF

Multicasting is a LAN technique that allows copies of a single packet to pass to a selected subset of all possible destinations. Some hardware (Ethernet, for example) supports multicast by allowing a network interface to belong to one or more multicast groups. Refer to "IP Multicast Support" on page 16-13 for details about the router's support of IP multicasting.

The OSPF protocol supports IP multicast routing through multicast extensions to OSPF (MOSPF).

An MOSPF router distributes group location information throughout the routing domain by flooding a new type (type 6) of link state advertisement, the group-membership-LSA. This enables the MOSPF routers to efficiently forward a multicast datagram to its multiple destinations. Each router does this by calculating the path of the multicast datagram as a tree whose root is the datagram source and whose terminal branches are LANs containing group members.

While running MOSPF, multicast datagram forwarding works in the following ways:

- Although forwarding IP multicasts is not reliable, IP multicast datagrams are delivered with the same best effort as with the delivery of IP unicasts.
- Multicast datagrams travel the shortest path between the datagram source and any particular destination (OSPF link state cost). This occurs because a separate tree is built for each datagram source and destination group pair.
- A multicast datagram is forwarded as a datalink multicast at each hop. The ARP protocol is not used. For some network technologies, mapping between Class D addresses and datalink multicast occurs while for others, Class D addresses are mapped to the datalink broadcast address.
- When paths from the datagram source to two separate group members share an initial common segment, only a single datagram is forwarded until the paths go in separate directions. The path can split at either a router or at a network. If the path splits at a router, the router replicates the packet before it is sent. If the path splits at a network, it replicates through a datalink multicast.

- A network configuration could include both MOSPF routers and routers without multicast extensions. In this configuration, all routers interoperate in the routing of unicasts. This allows you to slowly introduce multicast capability into an internetwork.

  Some configurations of MOSPF and non-MOSPF routers may produce unexpected failures in multicast routing.

- Separate multicast paths are constructed in MOSPF for each TOS. However, some routers do not support TOS-based routing. You can mix non-TOS routers with TOS-based routers but this causes TOS to be ignored in the forwarding of multicasts.
- The router can be configured to send SNMP traps to a multicast group address by adding a group address to a particular SNMP community name.

Configuring OSPF over ATM

The options for configuring OSPF over an ATM subnetwork depend on whether LAN Emulation or Classical IP over ATM is being used for the IP layer. In the case of LAN Emulation, OSPF is configured in the same way as for a real LAN. For Classical IP over ATM the OSPF configuration options are the same as for Wide Area Subnetworks. See “Configuring Wide Area Subnetworks” on page 23-11. Both NBMA and P-2-MP configurations are supported.

Configuring OSPF

The following sections present information on how to initially configure the OSPF protocol. This information outlines the tasks required to get the OSPF protocol up and running. Information on how to make further configuration changes is explained under “OSPF Configuration Commands” on page 23-17.

The following steps outline the tasks required to get the OSPF protocol up and running. The sections that follow explain each step in detail, including examples.

Before your router can run the OSPF protocol, you must do the following:

1. Enable the OSPF protocol. In doing so, you must estimate the final size of the OSPF routing domain. (See “Enabling the OSPF Protocol” on page 23-5.)
2. Set the OSPF router ID. (See “Setting OSPF Router IDs” on page 23-5.)
3. Define OSPF areas attached to the router. If no OSPF areas are defined, a single backbone area is assumed. (See “Defining Backbone and Attached OSPF Areas” on page 23-5.)
4. Define the router’s OSPF network interfaces. Set the cost of sending a packet out on each interface, along with a collection of the OSPF operating parameters. (See “Setting OSPF Interfaces” on page 23-9.)
5. If you want to forward IP multicasts (IP Class D addresses), enable IP multicast routing capability. (See “Multicast Forwarding” on page 23-11.)
6. If the router interfaces to non-broadcast networks (X.25 and Frame-Relay), you must set additional interface parameters. (See “Setting Non-Broadcast Network Interface Parameters” on page 23-11.) (See “Configuring Wide Area Subnetworks” on page 23-11.)

7. If you want the router to import routes learned from other routing protocols running on this router (BGP, RIP or statically configured routes), you have to enable AS boundary routing. In addition, you must define whether routes are imported as Type 2 or Type 1 externals. (See “Enabling AS Boundary Routing” on page 23-13.)
8. If you want to boot a neighboring router over an attached point-to-point interface, you must configure the neighbor’s IP address. Do this by adding the neighbor for the point-to-point interface.

Enabling the OSPF Protocol

When enabling the OSPF routing protocol, you must supply the following two values to estimate the final size of the OSPF routing domain:

- Total number of AS external routes that will be imported into the OSPF routing domain. A single destination may lead to multiple external routes when it is imported by separate AS boundary routers. For example, if the OSPF routing domain has two AS boundary routers, both importing routes to the same 100 destinations, set the number of AS external routes to 200.
- Total number of OSPF routers in the routing domain.

Configure these two values identically in all of your OSPF routers. Each router running the OSPF protocol has a database describing a map of the routing domain. This database is identical in all participating routers. From this database the IP routing table is built through the construction of a shortest-path tree, with the router itself as root. The routing domain refers to an AS running the OSPF protocol.

To enable the OSPF routing protocol, use the **enable** command as shown in the following example.

```
OSPF Config> enable ospf
Estimated # external routes[0]? 200
Estimated # OSPF routers [0]? 60
```

Setting OSPF Router IDs

Every router in an OSPF routing domain must be assigned a unique 32-bit router ID. The value used for the OSPF router ID is chosen as follows:

If the IP configuration **set router ID** command is used, the value configured is used as an OSPF router ID.

If the IP configuration **set internal address** command is used, the address configured is used as the OSPF router ID. It is recommended that the same value be used for the router ID and internal address, if defined.

If neither the router ID nor the internal address are configured during IP configuration, the first OSPF interface address will be used as the OSPF router ID.

Defining Backbone and Attached OSPF Areas

Figure 23-1 on page 23-7 shows a sample diagram of the structure of an OSPF routing domain. One division is between IP subnetworks within the OSPF domain and IP subnetworks external to the OSPF domain. The subnetworks included within the OSPF domain are subdivided into regions called *areas*. OSPF areas are collections of contiguous IP subnetworks. The function of areas is to reduce the OSPF overhead required to find routes to destinations in a different area. Overhead is reduced both because less information is exchanged between routers and because fewer CPU cycles are required for a less complex route table calculation.

Every OSPF routing domain must have at least a *backbone area*. The backbone is always identified by area number 0.0.0.0.. For small OSPF networks, the backbone is the only area required. For larger networks with multiple areas, the backbone provides a core that connects the areas. Unlike other areas, the backbone's subnets can be physically separate. In this case, logical connectivity of the backbone is maintained by configuring *virtual links* between backbone routers across intervening non-backbone transit areas.

Routers that attach to more than one area function as area *border routers*. All area border routers are part of the backbone, so a border router must either attach directly to a backbone IP subnet or be connected to another backbone router over a virtual link. In addition, there must be a collection of backbone subnetworks and virtual links that connects all of the backbone routers.

The information and algorithms used by OSPF to calculate routes vary according to whether the destination IP subnetwork is within the same area, in a different area within the same domain, or external to the OSPF domain. Every router maintains a complete map of all links within its area. All router to multi-access network, network to multi-access router, and router to router links are included in the map. A shortest path first algorithm is used to calculate the best routes to destinations within the area from this map. Routes between areas are calculated from summary advertisements originated by area border routers for IP subnetworks, IP subnetwork ranges, and autonomous system external (ASE) boundary routers located in other areas of the OSPF domain. External routes are calculated from ASE advertisements that are originated by ASE boundary routers and flooded throughout the OSPF routing domain.

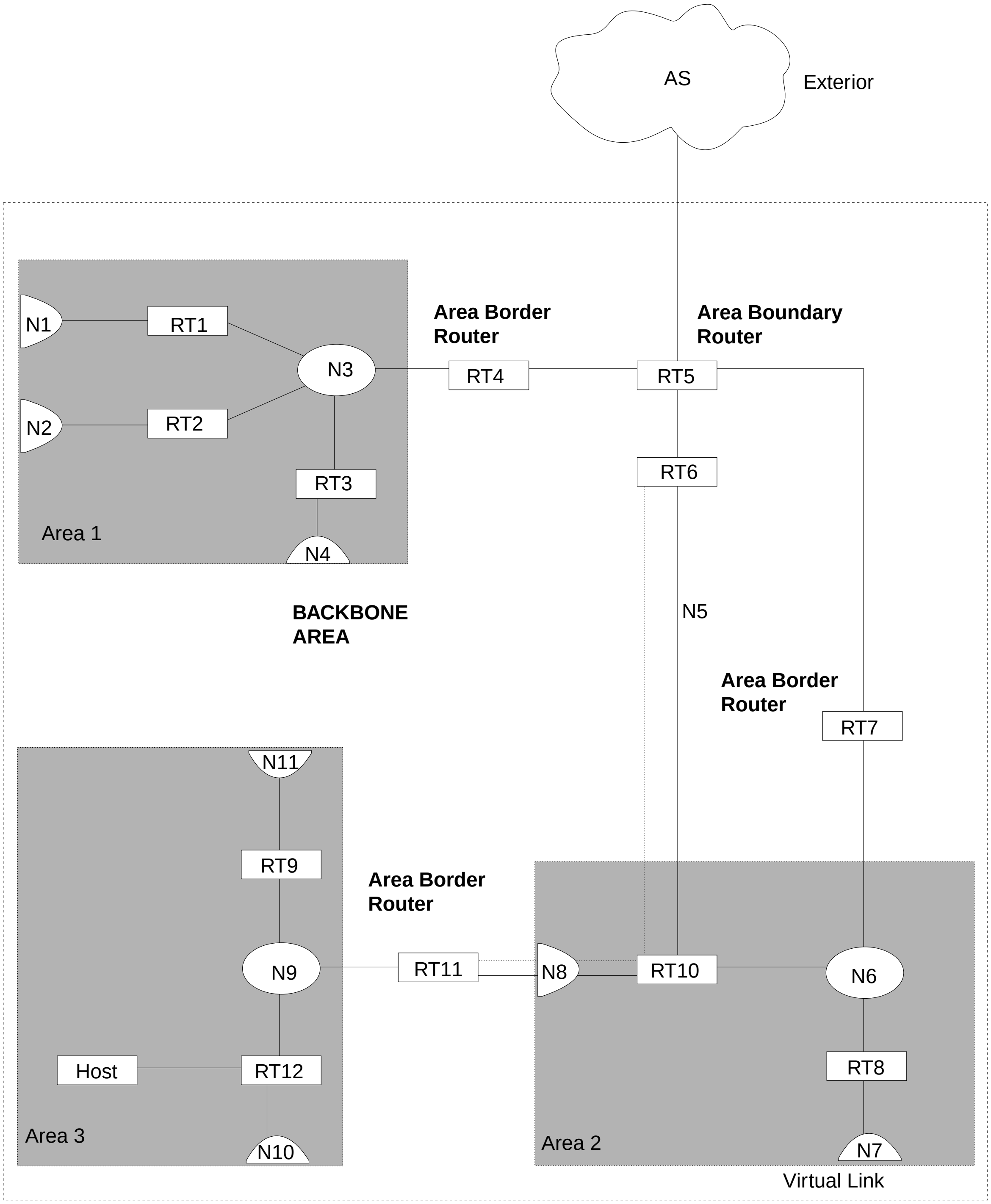


Figure 23-1. OSPF Areas

The backbone is responsible for distributing inter-area routing information. The backbone area consists of any of the following:

- Networks belonging to Area 0.0.0.0
- Routers attached to those networks
- Routers belonging to multiple areas
- Configured virtual links

The **set area** command is used to define areas that a router attaches to. If no set area command is used, the default is that all interfaces of the router attach to the backbone.

When area border routers are configured, options on the **set area** and **add range** commands can be used to control what OSPF route information crosses the area boundary.

One option is to use the set area command to define an area as a *stub*. ASE advertisements are never flooded into stub areas. In addition, the set area command has an option to suppress origination into the stub of summary advertisements for inter-area routes. Area border routers advertise default routes into stub areas. Traffic within the stub destined for unknown IP subnets is forwarded to the area border router. The border router uses its more complete routing information to forward the traffic on an appropriate path toward its destination. An area cannot be configured as a stub if it is used as a transit area for virtual links.

The other option is to use IP subnet address ranges to limit the number of summary advertisements that are used for inter-area advertisements of an area's subnets. A range is defined by an IP address and an address mask. Subnets are considered to fall within the range if the subnet IP address and the range IP address match after the range mask has been applied to both addresses. When a range is added for an area at an area border router, the border router suppresses summary advertisements for subnets in the areas that are included in the range. The suppressed advertisements would have been originated into the other areas that the border router attaches to. Instead, the area border router may originate a single summary advertisement for the range or no advertisement at all, depending on the option chosen with the add range command. Note that if the range is not advertised, there will be no inter-area routes for any destination that falls within the range. Also note that ranges cannot be used for areas that are used as transit areas by virtual links.

To set the parameters for an OSPF area, use the **set area** command and respond to the following prompts:

```
OSPF Config> set area
Area number [0.0.0.0]? 0.0.0.1
Authentication type [1]? 1
Is this a stub area?  [No]:
```

- Stub area designation

  Define an area as a stub when:

  1. There is no requirement for the area to handle transit backbone traffic.
  2. It is acceptable for area routers to use an area-border-router-generated default for traffic destined outside the AS.

  In this case, only the area border routers will have to store AS external routes.

Setting OSPF Interfaces

OSPF interfaces are a subset of the IP interfaces defined during IP configuration. The parameters configured for OSPF interfaces determine the topology of the OSPF domain, the routes that will be chosen through the domain, and the characteristics of the interaction between directly connected OSPF routers. The **set interface** command is used to define an OSPF interface and to specify some of its characteristics. Other characteristics of the interface were specified in response to the **add address** prompt during IP configuration.

OSPF Domain Topology

The definition of the topology of an OSPF domain depends on a definition of which routers are directly connected across some physical media or subnetwork technology and the area that those connections are part of. The basic case is for all routers attached to a physical subnetwork to be directly connected, but it is possible to define multiple IP subnetworks over a single subnetwork. In that case, OSPF will consider routers to be directly connected only when they have OSPF interfaces attached to the same IP subnetwork. It is also possible to have cases where routers attached to the same subnetwork do not have a direct link layer connection.

For LAN media, directly connected OSPF routers are determined from the IP subnetwork and physical media associated with an OSPF interface. The IP address of the OSPF interface is specified in response to the **Interface IP address** prompt. This address must match the address of an IP interface that was defined with the **add address** command during IP configuration. The IP address, along with the subnetwork mask defined with the **add address** command determine the IP subnetwork that the OSPF interface attaches to. The *net index* associated with the IP interface by the add address command determines the physical subnetwork to which the OSPF interface attaches. The broadcast capability of LANs allows OSPF to use multicast hello messages to discover other routers that have interfaces attached to the same IP subnetwork. Consequently, the interface parameters are all that are required for OSPF to determine which routers are directly connected across a LAN.

LANs may be used to connect an OSPF router with IP hosts. In this case, it is still necessary to define an OSPF interface to any IP subnetwork that is defined for the LAN. Otherwise, OSPF will not generate routes with those IP subnetworks as destinations.

The requirements for configuring OSPF interfaces that attach to serial lines vary with the lower layer technology.

For point-to-point lines, there is only one other router that is accessible over the interface, so the directly connected router can be determined without additional configuration. In fact, because there is no requirement to configure an IP subnetwork at all, unnumbered OSPF interfaces can be used for point-to-point lines. In this case, the same net index used as the IP address for the IP add address command is used as the IP address for the OSPF set interface command.

For subnetwork technologies like Frame Relay, ATM, and X.25 that support connections to multiple routers over a single serial line, the configuration of the OSPF interfaces is similar to that for a LAN, but because directly connected routers are not discovered dynamically for these subnetwork technologies, additional configuration is required to specify directly connected neighbors. For more information on the required configuration, see “Configuring Wide Area Subnetworks” on page 23-11.

Costs for OSPF Links

OSPF calculates routes by finding the least-cost path to a destination. The cost of each path is the sum of the costs for the different links in the path. The cost of a link to a directly connected router is specified at the **set interface** command for **Type of Service 0 cost**.

Correctly configuring the costs according to the desirability of using interfaces for data traffic is critical for obtaining the desired routes through an OSPF domain. The factors that make individual links more or less desirable may vary in different networks, but the most common goal is to choose routes with the least delay and the most capacity. In general, this policy can be achieved by making the cost of a link inversely proportional to the bandwidth of the media used for the physical subnetwork.

A recommended approach is to use a cost of one for the highest bandwidth technology. For example, use the value 1 as the cost for an interface running 100 Mbps ATM. With this approach, Ethernet interfaces would be configured with a cost of 10, 16 Mbps token-ring interfaces would be configured with a cost of 6, 4 Mbps token-ring interfaces would be configured with a cost of 25, and serial line interfaces would be configured with a cost that depends on their bandwidth.

The cost of an OSPF interface can be dynamically changed from the router’s monitoring environment. This new cost is flooded quickly throughout the OSPF routing domain, and modifies the routing immediately.

When the router restarts/reloads, the cost of the interface reverts to the value that has been configured in SRAM.

Interactions Between Neighbor Routers

A number of the values configured with the **set interface** command are used to specify parameters that control the interaction of directly connected routers. They include:

- Retransmission interval
- Transmission delay
- Router priority
- Hello interval
- Dead router interval
- Authentication key

In most cases, the default values can be used.

Note: The hello interval, the dead router interval, and the authentication key must have the same value for all OSPF routers that attach to the same IP subnetwork. If the values are not the same, routers will fail to form direct connections (adjacencies).

Multicast Forwarding

To enable the routing of IP multicast (class D) datagrams, use the **enable multicast-routing** command. When enabling multicast routing, you will also be prompted as to whether you want the router to forward multicasts between OSPF areas.

```
OSPF Config>enable multicast forwarding
Inter-area multicasting enabled?  [No]: yes
```

When the **enable multicast forwarding** command is first invoked, multicast is enabled on all OSPF interfaces with default parameters.

If you want to change the MOSPF parameters, use the **set interface** command. You will be queried for multicast parameters only if you have first enabled multicast forwarding.

Setting Non-Broadcast Network Interface Parameters

If the router is connected to a non-broadcast, multi-access network, such as an X.25 PDN, you have to configure the following parameters to help the router discover its OSPF neighbors. This configuration is necessary only if the router will be eligible to become designated router of the non-broadcast network.

First configure the OSPF poll interval with the following command:

```
OSPF Config> set non-broadcast
Interface IP address [0.0.0.0]? 128.185.138.19
Poll Interval [120]?
```

Then configure the IP addresses of all other OSPF routers that will be attached to the non-broadcast network. For each router configured, you must also specify its eligibility to become the designated router.

```
OSPF Config> add neighbor
Interface IP address [0.0.0.0]? 128.185.138.19
IP Address of Neighbor [0.0.0.0]? 128.185.138.21
Can that router become Designated Router [Yes]?
```

Configuring Wide Area Subnetworks

Frame Relay, Classical IP over ATM, and X.25 allow direct connections between multiple routers over a single serial line. Additional configuration beyond that achieved with the **set interface** command is required for OSPF interfaces that attach to this kind of network. Because OSPF protocol messages are sent directly to specific neighbors on these networks, configuration is used instead of dynamic discovery to determine neighbor relationships and router roles.

Note: The configurations described in this section do not apply to point-to-point networks.

OSPF can assume either of two patterns for the direct connections between routers across these subnetworks:

- Point-to-Multipoint
- Non-broadcast multiaccess (NBMA)

The key factor that distinguishes these two patterns is whether or not there is a direct connection between all pairs of routers that attach to the subnetwork (*full mesh connectivity*) or whether some of the routers are only connected through multihop paths with other routers as intermediates (*partial mesh connectivity*).

Non-broadcast multiaccess (NBMA) requires *full mesh connectivity* while point-to-multipoint requires only *partial mesh connectivity*.

Point-to-multipoint is the default choice because it works for both full mesh connectivity and partial mesh connectivity. But when full mesh connectivity is available, NBMA is a more efficient solution.

Configuring Point-to-Multipoint Subnetworks

Point-to-multipoint can be configured more easily than NBMA because there are no DRs, but neighbor relationships must be configured for all pairs of routers that will exchange data traffic directly across the point-to-multipoint subnet. Each pair of directly connected routers will exchange hello messages, so one side can discover the other through these messages. The router configured to send the first hello message, however, must have the IP address of its neighbor configured using the **add neighbor** command.

It is important to remember that OSPF will not calculate the correct routes if some of the routers attached to a subnetwork represent it as NBMA and others represent it as point-to-multipoint. Therefore, it is important that the **set non-broadcast** command is never used for any interface to a point-to-multipoint network.

Configuring NBMA Subnetworks

For NBMA IP subnetworks, some subset of the attached OSPF routers are configured to be eligible to be the designated router (DR). Each router eligible to be the DR periodically sends hello messages to all other routers eligible to be the DR. These messages are used in the protocol to elect a DR and a backup DR. Both the DR and the backup DR periodically exchange hello messages with all other OSPF routers that are attached to the NBMA IP subnetwork. Also, the flow of OSPF route information across the NBMA IP subnetwork is only between each of the attached routers and the DR or backup DR.

NBMA is selected by using the **set non-broadcast** command for interfaces that attach to an NBMA subnetwork. This command must be used for all interfaces that attach to the NBMA network.

The configuration required for an OSPF router that attaches to an NBMA subnetwork depends on whether or not that router is eligible to become the DR.

- For a router not eligible to become a DR, the **set interface** command must be used to set the router priority to 0.
- For a router eligible to become a DR, the **set interface** command must be used to set the router priority to a nonzero value and the **add neighbor** command must be used to identify all of the OSPF routers with interfaces attached to the NBMA subnetwork and to indicate which them are eligible to become DR.

Note: In a star configuration, use the **add neighbor** command at the hub (neighbors at the remote site do not need to be configured). The **add neighbor** command takes effect immediately without restarting the router.

Enabling AS Boundary Routing

To import routes learned from other protocols (RIP and statically configured information) into the OSPF domain, enable AS boundary routing. You must do this even if the only route you want to import is the default route (destination 0.0.0.0).

When enabling AS boundary routing, you are asked which external routes you want to import. You can choose to import, or not to import, routes belonging to several categories. The categories are as follows:

- BGP routes
- RIP routes
- Static routes
- Direct routes

For example, you can choose to import BGP and direct routes, but not RIP or static routes.

Independently of the above external categories, you can also configure whether or not to import subnet routes into the OSPF domain. This configuration item defaults to OFF (subnets not imported).

The metric type used in importing routes determines how the imported cost is viewed by the OSPF domain. When comparing two type 2 metrics, only the external cost is considered in picking the best route. When comparing two type 1 metrics, the external and internal costs of the route are combined before making the comparison. For example, you can set the router so that its default is originated only if a route to 10.0.0.0 is received from AS number 12. Setting the AS number to 0 means “from any AS.” Setting the network number to 0.0.0.0 means “any routes received.”

The syntax of the **enable** command is as follows:

```
OSPF Config>enable as boundary
Import BGP routes? [No]: yes
Import RIP routes? [No]:
Import static routes? [No]:
Import direct routes? [No]: yes
Import subnet routes? [No]:
Always originate default route? [No]: yes
Originate as type 1 or 2 [2]? 2
Default route cost [1]?
Default forwarding address [0.0.0.0]? 10.0.0.0
```

Other Configuration Tasks

Setting Virtual Links

To maintain backbone connectivity, you must have all of your backbone routers interconnected either by permanent or virtual links. You can configure virtual links between any two area border routers that share a common non-backbone and non-stub area. Virtual links are considered to be separate router interfaces connecting to the backbone area. Therefore, you are asked to also specify many of the interface parameters when configuring a virtual link.

The following example illustrates the configuration of a virtual link. Virtual links must be configured in each of the link’s two endpoints. Note that you must enter OSPF router IDs in the same form as IP addresses.

```
OSPF Config> set virtual
Virtual endpt.  (Router ID) [0.0.0.0]? 128.185.138.21
Link's transit area [0.0.0.1]?
Retransmission Interval (in seconds) [10]?
Transmission Delay (in seconds) [5]?
Hello Interval (in seconds) [30]?
Dead Router Interval (in seconds) [180]?
Authentication Key []? 3-14159
```

Configuring for Routing Protocol Comparisons

If you use a routing protocol in addition to OSPF, or when you change your routing protocol to OSPF, you must set the Routing Protocol Comparison.

OSPF routing in an AS occurs on these three levels: intra-area, inter-area, and exterior.

Intra-area routing occurs when a packet's source and destination address reside in the same area. Information that is about other areas does not affect this type of routing.

Inter-area routing occurs when the packet's source and destination addresses reside in different areas of the same AS. OSPF does inter-area routing by dividing the path into three contiguous pieces: an intra-area path from source to an area border router; a backbone path between the source and destination areas; and then another intra-area path to the destination. You can visualize this high-level of routing as a star topology with the backbone as hub and each of the areas as a spoke.

Exterior routes are paths to networks that lie outside the AS. These routes originate either from routing protocols, such as Border Gateway Protocol (BGP), or from static routes entered by the network administrator. The exterior routing information provided by BGP does not interfere with the internal routing information provided by the OSPF protocol.

AS boundary routers can import exterior routes into the OSPF routing domain. OSPF represents these routes as AS external link advertisements.

OSPF imports external routes in separate levels. The first level, called type 1 routes, is used when the external metric is comparable to the OSPF metric (for example, they might both use delay in milliseconds). The second level, called external type 2 routes, assumes that the external cost is greater than the cost of any internal OSPF (link-state) path.

Imported external routes are tagged with 32 bits of information. In a router, this 32-bit field indicates the AS number from which the route was received. This enables more intelligent behavior when determining whether to re-advertise the external information to other Autonomous systems.

OSPF has a 4-level routing hierarchy (see Figure 23-2 on page 23-15). The **set comparison** command tells the router where the BGP/RIP/static routes fit in the OSPF hierarchy. The two lower levels consist of the OSPF internal routes. OSPF intra-area and inter-area routes take precedence over information obtained from any other sources, all of which are located on a single level.

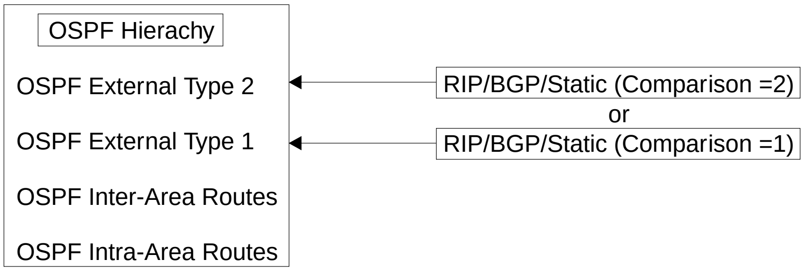


Figure 23-2. OSPF Routing Hierarchy

To put the BGP/RIP/static routes on the same level as OSPF external type 1 routes, set the comparison to 1. To put the BGP/RIP/static routes on the same level as OSPF external type 2 routes, set the comparison to 2. The default setting is 2.

For example, suppose the comparison is set to 2. In this case, when RIP routes are imported into the OSPF domain, they will be imported as type 2 externals. All OSPF external type 1 routes override received RIP routes, regardless of metric. However, if the RIP routes have a smaller cost, the RIP routes override OSPF external type 2 routes. The comparison values for all of your OSPF routers must match. If the comparison values set for the routers are inconsistent, your routing will not function correctly.

The syntax of the **set comparison** command is as follows:

```
OSPF Config> set comparison
Compare to type 1 or 2 externals [2]?
```

Converting from RIP to OSPF

To convert your Autonomous System from RIP to OSPF, install OSPF one router at a time, leaving RIP running. Gradually, all your internal routes will shift from being learned via RIP to being learned by OSPF (OSPF routes have precedence over RIP routes). If you want to have your routes look exactly as they did under RIP (in order to check that the conversion is working correctly) use hop count as your OSPF metric. This is done by assigning the cost of each OSPF interface to 1.

Remember that the size of your OSPF system must be estimated when the protocol is enabled. This size estimate should reflect the final size of the OSPF routing domain.

After installing OSPF on your routers, turn on AS boundary routing in all those routers that still need to learn routes via other protocols (BGP, RIP, and statically configured routes). The number of these AS boundary routers should be kept to a minimum.

Finally, you can disable the receiving of RIP information on all those routers that are not AS boundary routers.

Dynamically Changing Interface Costs

The cost of an OSPF interface can be dynamically changed from the router’s console interface. This new cost is flooded quickly throughout the OSPF routing domain, and modifies the routing immediately.

When the router restarts/reloads, the cost of the interface reverts to the value that has been configured in SRAM.

Accessing the OSPF Configuration Environment

To access the OSPF configuration environment, enter the following command at the Config> prompt:

```
Config> protocol ospf
Open SPF-based Routing Protocol configuration console
OSPF Config>
```

OSPF Configuration Commands

Before you can use OSPF, you must configure it using the OSPF configuration commands. The following section summarizes and then explains the OSPF commands. Enter these commands at the OSPF config> prompt. Table 23-1 shows the commands.

Table 23-1. OSPF Configuration Command Summary

Command	Function
? (Help)	Lists the OSPF configuration commands or lists the options associated with specific commands.
Add	Adds to already existent OSPF information. You can add ranges to areas, and neighbors to non-broadcast networks.
Delete	Deletes OSPF information from SRAM.
Disable	Disables the entire OSPF protocol, AS boundary routing capability, or IP multicast routing.
Enable	Enables the entire OSPF protocol, AS boundary routing capability, or IP multicast routing.
Join	Configures the router to belong to one or more multicast groups.
Leave	Removes the router from membership in multicast groups.
List	Displays OSPF configuration.
Set	Establishes or changes the configuration information concerning OSPF areas, interfaces, non-broadcast networks, or virtual links. This command also allows you to set the way in which OSPF routes are compared with information gained from other routing protocols.
Exit	Exits the OSPF configuration process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
add
delete
disable
enable
exit
join
leave
list
set
```

Add

Use the **add** command to add more information to already existing OSPF information. With this command you can add ranges to areas as well as neighbors to non-broadcast networks.

Syntax: add range . . .
neighbor . .

range *area# IP-address IP-address-mask*

Adds ranges to OSPF areas. OSPF areas can be defined in terms of address ranges. External to the area, a single route is advertised for each address range. For example, if an OSPF area were to consist of all subnets of the class B network 128.185.0.0, it would be defined as consisting of a single address range. The address range would be specified as an address of 128.185.0.0 together with a mask of 255.255.0.0. Outside of the area, the entire subnetted network would be advertised as a single route to network 128.185.0.0.

Ranges can be defined to control which routes are advertised externally to an area. There are two choices:

- When OSPF is configured to advertise the range, a single inter-area route is advertised for the range if at least one component route of the range is active within the area.
- When OSPF is configured not to advertise the range, no inter-area routes are advertised for routes that fall within the range.

Ranges cannot be used for areas that serve as transit areas for virtual links. Also, when ranges are defined for an area, OSPF will not function correctly if the area is partitioned but is connected by the backbone.

Example:

```
add range 0.0.0.2 128.185.0.0 255.255.0.0

inhibit advertisement ? [No]
```

1. The *area number* has:

   Valid Values: Any valid area number

   Default Value: none

2. The *IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

3. The *IP address mask* has:

   Valid Values: Any valid IP address mask.

   Default Value: none

neighbor

Configures neighbors adjacent to the router over this interface. Non-broadcast multi-access networks, neighbors need configured only to those routers that are eligible to become the designated router. In point-to-multipoint networks, at least one end of every logical connection must have a configured neighbor.

Example: **`add neighbor`**

```
Interface IP address [0.0.0.0]? 128.185.138.19
IP Address of Neighbor [0.0.0.0]? 128.185.138.21
Can that router become Designated Router on this net [Yes]?
```

1. The *Interface IP address* has:

   Valid Values: Any valid IP address. The last octet must be a zero.

   Default Value: none

2. The *IP Address of Neighbor* has:

   Valid Values: Any valid IP address.

   Default Value: none

3. Answer the question, *Can that router become designated router on this area?*.

Delete

Use the delete command to delete OSPF information from SRAM.

Syntax: delete range . . .
area . . .
interface . . .
neighbor . . .
non-broadcast . . .
virtual-link

range *area# IP-address*

Deletes ranges from OSPF areas.

Example: **`delete range 0.0.0.2 128.185.0.0 255.255.0.0`**

1. The *area number* of the range has:

   Valid Values: Any valid area address

   Default Value: none

2. The *IP Address of Range* has:

   Valid Values: Any valid IP address.

   Default Value: none

3. The *IP Address Mask of Range* has:

   Valid Values: Any valid IP address mask.

   Default Value: none

area *area#*

Deletes OSPF areas from the current OSPF configuration.

Example: **`delete area 0.0.0.1`**

interface *interface-IP-address*

Deletes an interface from the current OSPF configuration.

Example: **`delete interface 128.185.138.19`**

The *interface IP address* has:

Valid Values: Any valid IP address.

Default Value: none

neighbor

Deletes configured neighbors from the current OSPF configuration.

Example: **`delete neighbor`**

```
      Interface IP address [0.0.0.0]? 128.185.138.19
      IP Address of Neighbor [0.0.0.0]? 128.185.138.21
```

1. The *interface IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

2. The *neighbor IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

non-broadcast *interface-IP-address*

Deletes non-broadcast network information from the current OSPF configuration.

Example: **`delete non-broadcast 128.185.133.21`**

1. The *interface IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

virtual-link

Deletes a virtual link that you have set using the **set virtual-link** command.

Example: **`delete virtual-link`**

```
      Virtual endpoint (Router ID) [0.0.0.0]?
      Link's transit area [0.0.0.1]?
```

1. The *virtual endpoint (router ID)* that defines the id of the virtual neighbor has:

   Valid Values: Any valid IP address.

   Default Value: none

2. The *link's transit area* has:

   Valid Values: Any valid area address.

   Default Value: 0.0.0.1

Disable

Use the **disable** command to disable either the entire OSPF protocol or just the AS boundary routing capability.

Syntax: disable as boundary routing
multicast forwarding
OSPF routing protocol

as boundary routing

Disables the AS boundary routing capability. When disabled, the router will not import external information into the OSPF domain.

Example: `disable as boundary routing`

multicast forwarding

Disables IP multicast routing on all interfaces. When disabled, the router will not forward IP multicast (Class D) datagrams.

Example: `disable multicast forwarding`

OSPF routing protocol

Disables the entire OSPF protocol.

Example: `disable OSPF routing protocol`

Enable

Use the **enable** command to enable either the entire OSPF protocol, the advertisement of a stub to route to a subnet, or just the AS boundary routing capability.

Syntax: enable as boundary routing
multicast forwarding
OSPF routing protocol
subnet

as boundary routing

Enables the AS boundary routing capability which allows you to import routes learned from other protocols (BGP, RIP, and statically configured information) into the OSPF domain. For additional information on the use of the **enable** command, see “Configuring OSPF” on page 23-4.

Example: `enable as boundary routing`

```
Import BGP routes? [No]:  yes
Import RIP routes? [No]:
Import static routes? [No]:
Import direct routes? [No]: yes
Import subnet routes? [No]:
Always originate default route? [No]: yes
Originate as type 1 or 2 [2]? 2
Default route cost [1]?
Default forwarding address [0.0.0.0]? 10.0.0.0
```

1. The *Default route cost* is the parameter that specifies the cost that OSPF associates with the default route to its area border router. The cost is used to determine the shortest path for the default route to its area border router.

   Valid Values: 0 to 16777215

   Default Value: 1

2. The *Default forwarding address* is the parameter that specifies the forwarding address that will be used in the imported default route.

   Valid Values: a valid IP address

   Default Value: none

multicast forwarding

Enables the forwarding of IP multicast (Class D) datagrams. When enabling multicast routing, you are also prompted whether you want to forward IP multicast datagrams between OSPF areas. To run MOSPF (OSPF with multicast extensions), a router currently running OSPF needs only to use this command. You do not need to reenter its configuration information.

Example: **enable multicast forwarding**

```
        Inter-area multicasting enabled (Yes or No): yes
```

OSPF routing protocol

Enables the entire OSPF protocol. When enabling the OSPF routing protocol, you must supply the following two values that will be used to estimate the size of the OSPF link state database:

- Total number of AS external routes that will be imported into the OSPF routing domain. A single destination may lead to multiple external routes when it is imported by separate AS boundary routers. For example, if the OSPF routing domain has two AS boundary routers, both importing routes to the same 100 destinations, the number of AS external routes should be set to 200.

  Valid Values: 0 to 65535

  Default Value: 100

- Total number of OSPF routers in the routing domain.

  Valid Values: 0 to 65535

  Default Value: 50

Example: **enable OSPF routing protocol**

```
        Estimated # external routes[0]? 200
        Estimated # OSPF routers [0]? 60
```

subnet

For an interface to a point-to-point serial line, this option enables the advertisement of a stub route to the subnet that represents the serial line rather than the host route for the other router’s address. You must supply this router’s address for the interface to identify it.

Example:

```
OSPF Config> enable subnet
Interface IP address [0.0.0.0]? 8.24.3.1
```

The *interface IP address* has:

Valid Values: Any valid IP address.

Default Value: none

Join

Use the **join** command to configure the router as a member of a multicast group. When the router is the member of a multicast group, it responds to PINGS and SNMP queries sent to the group address.

To request group membership in a more temporary and more immediate way (a restart/reload is not required), issue the **join** command from the OSPF monitoring console. Also, from the OSPF monitoring console, the join command keeps track of the number of times a particular group is joined.

Syntax: join *multicast-group-address*

Example: `join 224.185.0.0`

The *group address* parameter specifies the 6-byte (12-digit hexadecimal) group/multicast address.

Valid Values: class D IP address from 224.0.0.1 to 239.255.255.255

Default Value: none

Leave

Use the **leave** command to remove a router's membership from a multicast group. This will prevent the router from responding to PINGS and SNMP queries sent to the group address.

To delete group membership in a more immediate way (a restart/reload is not required), issue the **leave** command from the OSPF monitoring console. Also, from the OSPF monitoring console, group membership is not deleted until the number of leaves executed equals the number of joins previously executed.

Syntax: leave *multicast-group-address*

Example: `leave 224.185.0.0`

The *address to be deleted* has:

Valid Values: Any valid IP address.

Default Value: none

List

Use the **list** command to display OSPF configuration information.

Syntax: list all
areas
interfaces
neighbors
non-broadcast
virtual-links

all

Lists all OSPF-related configuration information.

Example: `list all`

```
          --Global configuration--
    OSPF Protocol:          Enabled
    # AS ext. routes:       300
    Estimated # routers:    100
    External comparison:    Type 2
    AS boundary capability: Enabled
    Import external routes: BGP RIP STA DIR SUB
    Orig. default route:    No (0,0.0.0.0)
    Default route cost:     (1, Type 2)
    Default forward. addr.: 0.0.0.0
    Multicast forwarding:   Enabled
    Inter-area multicast:   Enabled

            --Area configuration--
Area ID          AuType          Stub? Default-cost Import-summaries?
0.0.0.0          0=None          No         N/A           N/A


       --Interface configuration--
    IP address        Area    Cost  Rtrns  TrnsDly  Pri  Hello Dead
    128.185.184.11    0.0.0.1   1      5       1     1    10    60
    128.185.177.11    0.0.0.1   1      5       1     1    10    60
    128.185.142.11    0.0.0.0   1      5       1     1    10    60
```

OSPF protocol Displays whether OSPF is enabled or disabled.

# AS ext. routes Displays the estimated number of Autonomous System external routes. The router cannot accept more than this number of AS external routes.

Estimated # routers Displays the estimated number of routers found in the OSPF configuration.

External comparison Displays the external route type used by OSPF when importing external information into the OSPF domain and when comparing OSPF external routes to RIP/BGP routes.

AS boundary capability Displays whether the router will import external routes into the OSPF domain.

Import external Displays which routes will be imported.

Orig default route Displays whether the router will import a default into the OSPF domain. When the value is “YES,” a nonzero network number is displayed in parentheses. This indicates that the default route will originated only if a route to that network is available.

Default route cost Displays the cost and type that will be used in the imported default route.

Default forward addr Displays the forwarding address that will be used in the imported default route.

Multicast forwarding Displays whether IP multicast datagrams will be forwarded.

Inter-area multicast Displays whether IP multicast datagrams will be forwarded between areas.

Area-ID Displays the attached area ID (area summary information)

AuType Displays the method used for area authentication. “Simple-pass” means a simple password scheme is being used for the area’s authentication.

Stub area
Displays whether or not the area being summarized is a stub area. Stub areas do not carry external routes, resulting in a smaller routing database. However, stub areas cannot contain AS boundary routers, nor can they support configured virtual links.

OSPF interfaces
For each interface, its IP address is printed, together with configured parameters. “Area” is the OSPF area to which the interface attaches. “Cost” indicates the TOS 0 cost (or metric) associated with the interface. “Rtrns” is the retransmission interval, which is the number of seconds between retransmissions of unacknowledged routing information. “TrnsDly” is the transmission delay, which is an estimate of the number of seconds it takes to transmit routing information over the interface (it must be greater than 0). “Pri” is the interface’s Router Priority, which is used when selecting the designated router. “Hello” is the number of seconds between Hello Packets sent out the interface. “Dead” is the number of seconds after Hellos cease to be heard that the router is declared down.

Virtual links
Lists all virtual links that have been configured with this router as endpoint. “Virtual endpoint” indicated the OSPF Router ID of the other endpoint. “Transit area” indicates the non-backbone area through which the virtual link is configured. Virtual links are considered treated by the OSPF protocol similarly to point-to-point networks. The other parameters listed in the command (“Rtrns,” “TrnsDly,” “Hello,” and “Dead”) are maintained for all interfaces. See the OSPF list interfaces command for more information.

areas

Lists all information concerning configured OSPF areas.

Example: **`list areas`**

```
                        --Area configuration--
Area ID          AuType         Stub? Default-cost Import-summaries?
0.0.0.0          0=None          No         N/A           N/A
0.0.0.1          1=Simp-Pass     No         N/A           N/A
```

Area-ID
Displays the attached area ID (area summary information).

AuType
Displays the method used for area authentication. “Simple-pass” means a simple password scheme is being used for the area’s authentication.

Stub area
Displays whether or not the area being summarized is a stub area.

interfaces

For each interface, its IP address is printed, together with configured parameters. “Area” is the OSPF area to which the interface attaches. “Cost” indicates the TOS 0 cost (or metric) associated with the interface. “Rtrns” is the retransmission interval, which is the number of seconds between retransmissions of unacknowledged routing information. “TrnsDly” is the transmission delay, which is an estimate of the number of seconds it takes to transmit routing information over the interface (it must be greater than 0). “Pri” is the interface’s router priority, which is used when selecting the designated router. “Hello” is the number of seconds between Hello Packets sent out the interface. “Dead” is the number of seconds after Hellos cease to be heard that the router is declared down.

Example: **`list interfaces`**

```
               --Area configuration--
IP address       Area     Cost  Rtrns  TrnsDly  Pri Hello Dead
128.185.208.43   1.1.1.1   1      5       1       1    10   40
10.1.155.43      0.0.0.0   1      5       1       1    10   40
10.1.152.43      0.0.0.0   1      5       1       1    10   40


               --Multicast Parameters--
IP address       MCForward   DLUnicast  IGMPoll  IGMPtimeout
128.185.208.43   On          Off        60       180
10.1.155.43      On          Off        60       180
10.1.152.43      On          Off        60       180
```

Note: Multicast parameters are not displayed if multicast is disabled.

neighbors

Lists neighbors to non-broadcast networks. It displays IP address of the neighbor and the IP address of the interface to that neighbor. It also indicates whether the neighbor is eligible to become the “Designated Router” on the net.

Example: **`list neighbors`**

```
        --Neighbor configuration--
Neighbor Addr     Interface Address   DR eligible?
2.3.4.5           1.2.3.4             yes
2.5.6.7           5.6.7.8             no
```

non-broadcast

Lists all information related to interfaces connected to non-broadcast multi-access networks. For each non-broadcast interface, as long as the router is eligible to become designated router on the attached network, the polling interval is displayed together with a list of the router’s neighbors on the non-broadcast network.

Example: **`list non-broadcast`**

```
        --NBMA configuration--
Interface Addr      Poll Interval
128.185.235.34      120
```

virtual-links

Lists all virtual links that have been configured with this router as endpoint. “Virtual endpoint” indicated the OSPF router ID of the other endpoint. “Transit area” indicates the non-backbone area through which the virtual link is configured. Virtual links are considered treated by the OSPF protocol similarly to point-to-point networks. The other parameters listed in the command (“Rtrns,” “TrnsDly,” “Hello,” and “Dead”) are maintained for all interfaces. See the OSPF **list interfaces** command for more information.

Example: **`list virtual-links`**

```
         --Virtual link configuration--
Virtual endpoint   Transit area   Rtrns  TrnsDly Hello Dead
0.0.0.0            0.0.0.1         10      5     30    180
```

Set

Use the **set** command to display or change the configuration information concerning OSPF areas, interfaces, non-broadcast networks, or virtual links. This command also allows you to set the way in which OSPF routes are compared to information obtained from other routing protocols.

Syntax: set area
comparison
interface
non-broadcast
virtual-link

area

Sets the parameters for an OSPF area. If no areas are defined, the router software assumes that all the router's directly attached networks belong to the backbone area (area ID 0.0.0.0).

Example: **set area**

```
Area number [0.0.0.0]? 0.0.0.1
Authentication type [1]? 1
Is this a stub area? [No]:  yes
Stub default cost? [0]:
Import summaries? [Yes]:
```

- *Area number* - is the OSPF area address.
- *Authentication type* - (security scheme) to be used in the area. The choices for authentication types are 1, which indicates a simple password; or 0, which indicates that no authentication is necessary to pass packets.

  All OSPF routers attached to the same subnet must have the same Authentication Key. For example, suppose the address mask for this network interface is 255.255.255.0, the IP address is 128.185.138.19 and the authentication key is *xyz123*. According to the subnet mask and IP address combination, the interface attaches to the subnet 128.185.138.0 of network 128.185.0.0. All other OSPF routers attached to subnet 128.185.138.0 must have their authentication key set to *xyz123*.

  Valid Values: any 8 characters

  Default Value: 0
- *Stub area designation.* If you designate YES:
  - The area does not receive any AS external link advertisements, reducing the size of your database and decreasing memory usage for routers in the stub area.
  - You cannot configure virtual links through a stub area.
  - You cannot configure a router within the stub area as an AS boundary router.

  External Routing in Stub Areas. You cannot configure the backbone as a stub area. External routing in stub areas is based on a default route. Each border area router attaching to a stub area originates a default route for this purpose. The cost of this default route is also configurable with the **set area** command.

comparison

Tells the router where the BGP/RIP/static routes fit in the OSPF hierarchy. The two lower levels consist of the OSPF internal routes. OSPF internal routes take precedence over information gained from any other sources, all of which are located on a single level.

Example: **`set comparison`**

```
OSPF Config> set comparison
Compare to type 1 or 2 externals [2]?
```

interface

Sets the OSPF parameters for the router's network interfaces.

1. The *interface IP address* is for each interface in the router.
2. *attaches to area* is the area to which the interface attaches.
3. The timer values are the same values for all routers attached to a common network segment.
   1. The *retransmission interval* is the interval after which a Link Request for one or more link state advertisements will be re-sent.

      Valid values: 1 to 65535 seconds

      Default Value: 5

   2. The *Transmission delay* is an estimate of the number of seconds that it takes to transmit link-state information over the interface.

      Each link-state advertisement has a finite lifetime that is equal to the constant MaxAge (1 hour). As each link-state advertisement is sent to the particular interfaces, it is aged by this configured transmission delay. The minimum delay is 1 second.

      Valid Values: 1 to 65535 seconds

      Default Value: 1

   3. The *Router Priority* value is used for broadcast and non-broadcast multiaccess networks to elect the designated router. For point-to-point links, this value should be **0**, which means that this router must not be elected the designated router for its network.

      Valid Values: 0 to 255

      Default Value: 1

   4. The *Hello Interval* is the interval between hello packets sent on the interface.

      Valid Values: 1 to 65535 seconds

      Default Value: 10

   5. The *Dead Router Interval*

      Dead Router Interval is the interval after which a router that has not sent a hello will be considered dead. The Dead Router Interval defaults to four times the configured Hello Interval. The value for this parameter must be greater than the Hello Interval.

      Valid Values: 2 to ≥ 65535 seconds

      Default Value: 40 (or four times the configured hello interval)

4. The *Type of service 0 cost*.

   Valid Values: 1 to 65535

Default Value: 1

5. The *Authentication key* is the parameter that defines the password used for this OSPF area. When password authentication is used, only packets with the correct authentication key are accepted.

   Valid Values: any 8 characters

   Default Value: 0

Example: **set interface**

```
Interface IP address [0.0.0.0]? 128.185.138.19
Attaches to area [0.0.0.0]? 0.0.0.1
Retransmission Interval (in seconds) [5]?
Transmission Delay (in seconds) [1]? 1
Router Priority [1]? 1
Hello Interval (in seconds) [10]? 10
Dead Router Interval (in seconds) [60]? 40
Type Of Service 0 cost [1]? 5
Authentication Key []? xyz_q
Retype Auth.  Key []? xyz_q
Forward multicast datagrams (Yes or No)? Yes
Forward as datalink unicasts (Yes or No)? No
IGMP polling interval (in seconds) [60]? 60
IGMP timeout (in seconds) [180]? 180
```

When responding to the prompts, supply the IP address for each interface in the router and answer the questions that follow. For the following parameters, you must enter the same value for all routers attached to a common network:

- Hello interval
- Dead router interval
- Authentication key (if an authentication of 1 is used)

The first prompt asks for the OSPF area to which the interface attaches. For example, suppose that the interface address mask is 255.255.255.0, indicating that the interface attaches to a subnet (128.185.138.0) of network 128.185.0.0. All other OSPF routers attached to subnet 128.185.138.0 must also have their *hello interval* set to 10, *dead router interval* set to 40, and their interface *authentication key* set to xyz_q.

Note that IP interfaces to point-to-point lines may be unnumbered. In this case a net index is configured instead of an IP address. This implementation of OSPF will work with these unnumbered interfaces, but to work correctly, both ends of the point-to-point line must use an unnumbered interface.

In a multicast routing configuration (multicast has been enabled), the MOSPF parameters for each OSPF interface are set to their default values. This means that:

- Multicast forwarding is enabled.
- Multicast datagrams are forwarded as data-link multicasts.
- IGMP Host Membership is sent out on the interface every 60 seconds.
- Local group database entries are removed 180 seconds after IGMP Host Membership reports for the group cease to be received by the interface.

If you want to change the MOSPF parameters, use the **set interface** command. You will be queried for multicast parameters (the last five parameters shown in the output display above) only if you have first enabled multicast forwarding.

On networks that lie on the edge of an Autonomous System, where multiple multicast routing protocols (or multiple instances of a single multicast routing protocol) may exist, you may need to configure forwarding as data-link unicasts to avoid unwanted datagram replication. In any case, for all routers attached to a common network, the interface parameters “forward multicast datagrams” and “forward as data-link unicasts” should be configured identically.

non-broadcast

Overrides the point-to-multipoint default to select NBMA for X.25, Frame Relay or ATM networks. You must set non-broadcast consistently across all interfaces that attach to the same subnetwork for OSPF to function correctly.

The *interface IP address* has:

Valid Values: Any valid IP address.

Default Value: none

The NBMA Poll Interval is used to send Hello packets to inactive neighbors. (Inactive neighbors are those neighbors that the router has not heard from for a period greater than the Dead Router interval.) The router still polls these neighbors at a reduced rate. Set the NBMA Poll Interval much higher than the configured Hello Interval for the router.

Valid Values: 1 to 65535 seconds

Default Value: 120 seconds

Example: **set non-broadcast**

```
Interface IP address [0.0.0.0]? 128.185.138.19
Poll Interval [120]?
```

virtual-link

Configures virtual links between any two area border routers. To maintain backbone connectivity you must have all of your backbone routers interconnected either by permanent or virtual links. Virtual links are considered to be separate router interfaces connecting to the backbone area. Therefore, you are asked to also specify many of the interface parameters when configuring a virtual link.

Virtual links can be configured between any two backbone routers that have an interface to a common non-backbone area. Virtual links are used to maintain backbone connectivity and must be configured at both endpoints.

Note: This OSPF implementation supports the use of virtual links when one end of the virtual link may be an unnumbered point to point line. For this configuration to work, the router id must be used as the source address in OSPF protocol messages sent over the virtual link. Use of the router id can be insured by configuring the internal IP address with the address used as the router id. Another requirement for this configuration to work is that the OSPF implementations at both ends of the virtual link support it.

1. The *virtual endpoint (router ID)* defines the ID of the virtual neighbor.

   Valid Values: Any valid IP address.

   Default Value: none

2. The *link’s transit area.* is the non-backbone, non-stub area through which the virtual link is configured. Virtual links can be configured between any two area border routers that have an interface to a common

non-backbone and non-stub area. Virtual links must be configured in each of the link's two endpoints.

Valid Values: 0.0.0.1 to 255.255.255.255

Default Value: 0.0.0.1

3. The timer values are the same values for all routers attached to a common network segment.

   a. The *retransmission interval* is the interval after which a Link Request for one or more link state advertisements will be re-sent.

      Valid Values: 1 to 65535 seconds

      Default Value: 10

   b. The *Transmission delay* parameter is an estimate of the number of seconds that it takes to transmit link-state information over the interface.

      Each link-state advertisement has a finite lifetime that is equal to the constant MaxAge (1 hour). As each link-state advertisement is sent to the particular interfaces, it is aged by this configured transmission delay. The minimum delay is 1 second.

      Valid Values: 1 to 65535 seconds

      Default Value: 5

      The *Hello Interval* is the interval between hello packets sent on the interface.

      Valid Values: 1 to 255 seconds

      Default Value: 30

   c. The *Dead Router Interval* is the interval after which a router that has not sent a hello will be considered dead. This parameter defaults to six times the configured Hello Interval and must be set to a value greater than the Hello Interval.

      Valid Values: 2 to 65535 seconds

      Default Value: 180

4. The *Authentication key*. defines the password used for this OSPF area. When password authentication is used, only packets with the correct authentication key are accepted.

Example: **set virtual-link**

```
Virtual endpt.  (Router ID) [0.0.0.0]? 128.185.138.21
Link's transit area [0.0.0.1]? 0.0.0.1
Retransmission Interval (in seconds) [10]?
Transmission Delay (in seconds) [5]?
Hello Interval (in seconds) [30]?
Dead Router Interval (in seconds) [180]?
Authentication Key []? 314159
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example:

```
exit
```

Chapter 24. Monitoring OSPF

This chapter describes the OSPF console commands and contains the following sections:

- "Accessing the OSPF Console Environment"
- "OSPF Console Commands."

Accessing the OSPF Console Environment

For information on how to access the OSPF console environment, refer to *Getting Started (Introduction to the User Interface)* in the *Software User's Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

OSPF Console Commands

This section summarizes and then explains all the OSPF console monitoring commands. These commands enable you to monitor the OSPF routing protocol. Table 24-1 lists the OSPF console commands.

Enter the OSPF console commands at the `OSPF>` prompt.

Table 24-1 (Page 1 of 2). OSPF Console Command Summary

Command	Function
? (Help)	Lists the OSPF console commands or lists the options associated with specific commands.
Advertisement	Displays a link state advertisement belonging to the OSPF database.
Area summary	Displays OSPF area statistics and parameters.
AS external	Lists the AS external advertisements belonging to the OSPF link state database.
Database summary	Displays the advertisements belonging to an OSPF area's link state database.
Dump routing tables	Displays the OSPF routes contained in the routing table.
Interface summary	Displays OSPF interface statistics and parameters.
Join	Configures the router to belong to one or more multicast groups.
Leave	Removes the router from membership in multicast groups.
Mcache	Displays a list of currently active multicast forwarding cache entries.
Mgroups	Displays the group membership of the router's attached interfaces.
Mstats	Displays various multicast routing statistics.
Neighbor summary	Displays OSPF neighbor statistics and parameters.
Ping	Continuously sends ICMP Echo Requests (or pings) a given destination, printing a line for each response received.

Table 24-1 (Page 2 of 2). OSPF Console Command Summary

Command	Function
Routers	Displays the reachable OSPF area-border routers and AS-boundary routers.
Size	Displays the number of LSAs currently in the link state database, categorized by type.
Statistics	Displays OSPF statistics detailing memory and network usage.
Traceroute	Displays the complete route (hop-by-hop) to a given destination.
Weight	Dynamically changes the cost of an OSPF interface.
Exit	Exits the OSPF console process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADVERTISEMENT expansion
AREA summary
AS-EXTERNAL advertisement
DATABASE summary
DUMP routing tables
EXIT
INTERFACE summary
JOIN
LEAVE
MCACHE
MGROUPS
MSTATS
NEIGHBOR summary
PING address
ROUTERS
SIZE
STATISTICS
TRACEROUTE
WEIGHT
```

Advertisement Expansion

Use the **advertisement expansion** command to print the contents of a link state advertisement contained in the OSPF database. For a summary of the router's advertisements use the **database** command.

A link state advertisement is defined by its link state type, link state ID and its advertising router. There is a separate link state database for each OSPF area. Providing an area-id on the command line tells the software which database you want to search. The different kinds of advertisements, which depend on the value given for link-state-type, are:

- Router links - Contain descriptions of a single router's interface.
- Network links - Contain the list of routers attached to a particular interface.

- Summary nets - Contain descriptions of a single inter-area route.
- Summary AS boundary routers - Contain descriptions of the route to an AS boundary router in another area.
- AS external nets - Contain descriptions of a single route.
- Multicast group memberships - Contain descriptions of a particular group's membership in the neighborhood of the advertising router.

  Note: Link State IDs, advertising routers (specified by their router IDs), and area IDs take the same format as IP addresses. For example, the backbone area can be entered as 0.0.0.0.

Example 1 shows an expansion of a router links advertisement. The router's ID is 128.185.184.11. It is an AS boundary router and has three interfaces to the backbone area (all of cost 1). Multicast routing has been enabled. Detailed field descriptions are provided with the example.

This command has also been enhanced in two ways. First of all, when displaying router-LSAs and network-LSAs, the reverse cost of each router-to-router link and router-to-transit-network link is displayed, as well as the previously displayed forward cost. This is done because routing of multicast datagrams whose source lies in different areas/Autonomous systems is based on reverse cost instead of forward cost. In those cases where there is no reverse link (which means that the link will never be used by the Dijkstra), the reverse cost is shown as “1-way.”

In addition, the LSA's OSPF options are displayed in the same manner as they were displayed in the detailed OSPF **neighbor** command.

New group-membership-LSAs can also be displayed. The “LS destination” of each group-membership-LSA is a group address. A router originates a group-membership-LSA for each group that has members on one or more of the router's attached networks. The group-membership-LSA for the group lists those attached transit networks having group members (the type “2” vertices), and when there are members belonging to one or more attached stub networks, or if the router itself is a member of the multicast group, a type “1” vertex whose ID is the router's OSPF router ID is included.

Syntax: advertisement *ls-type link-state-id [advertising-router] [area-id]*

Example 1: `advertisement 1 128.185.184.11 0.0.0.0`

```
LS age:     173
LS options:  E,MC
LS type:     1
LS destination (ID): 128.185.184.11
LS originator:   128.185.184.11
LS sequence no:  0x80000047
LS checksum:     0x122
LS length:       60
Router type:  ASBR,W
# router ifcs:   3
         Link ID:          128.185.177.31
         Link Data:        128.185.177.11
         Interface type:   2
                  No. of metrics: 0
                  TOS 0 metric:   3 (0)
         Link ID:          128.185.142.40
         Link Data:        128.185.142.11
         Interface type:   2
                  No. of metrics: 0
```

```
        TOS 0 metric:   4 (0)
Link ID:         128.185.184.0
Link Data:       255.255.255.0
Interface type:   3
        No. of metrics: 0
        TOS 0 metric:   1
```

LS age Indicates the age of the advertisement in seconds.

LS options Indicates the optional OSPF capabilities supported by the piece of the routing domain described by the advertisement. These capabilities are denoted by E (processes type 5 externals; when this is not set to the area to which the advertisement belongs has been configured as a stub), T (can route based on TOS) and MC (can forward IP multicast datagrams).

LS type Classifies the advertisement and dictates its contents: 1 (router links advertisement), 2 (network link advertisement), 3 (summary link advertisement), 4 (summary ASBR advertisement), 5 (AS external link) and 6 (group-membership advertisement).

LS destination Identifies what is being described by the advertisement. Depends on the advertisement type. For router links and ASBR summaries, it is the OSPF router ID. For network links, it is the IP address of the network's designated router. For summary links and AS external links, it is a network/subnet number. For group-membership advertisements, it is a particular multicast group.

LS originator OSPF router ID of the originating router.

LS sequence number Used to distinguish separate instances of the same advertisement. Should be looked at as a signed 32-bit integer. Starts at 0x80000001, and increments by one each time the advertisement is updated.

LS checksum A checksum of advertisement contents, used to detect data corruption.

LS length The size of the advertisement in bytes.

Router type Indicates the level of function of the router. ASBR means that the router is an AS boundary router, ABR that the router is an area border router, and W that the router is a wildcard multicast receiver.

# Router ifcs The number of router interfaces described in the advertisement.

Link ID Indicates what the interface connects to. Depends on Interface type. For interfaces to routers (i.e., point-to-point links), the Link ID is the neighbor's router ID. For interfaces to transit networks, it is the IP address of the network designated router. For interfaces to stub networks, it is the network's network/subnet number.

Link Data 4 bytes of extra information concerning the link, it is either the IP address of the interface (for interfaces to point-to-point networks and transit networks), or the subnet mask (for interfaces to stub networks).

Interface type One of the following: 1 (point-to-point connection to another router, 2 (connection to transit network, 3 (connection to stub network) or 4 (virtual link).

No. of metrics The number of non-zero TOS values for which metrics are provided for this interface.

TOS 0 metric The cost of the interface. In parenthesis the reverse cost of the link is given (derived from another advertisement). If there is no reverse link, “1-way” is displayed.

The LS age, LS options, LS type, LS destination, LS originator, LS sequence no, LS checksum and LS length fields are common to all advertisements. The Router type and # router ifcs are seen only in router links advertisements. Each link in the router advertisement is described by the Link ID, Link Data, and Interface type fields. Each link can also be assigned a separate cost for each IP Type of Service (TOS); this is described by the No. of metrics and TOS 0 metric fields (the router currently does not route based on TOS, and looks at the TOS 0 cost only).

Example 2 shows an expansion of a group-membership advertisement. A group-membership advertisement for a given group/advertising router combination lists those networks directly attached to the advertising router which have group members. It also lists whether the router itself is a member of the specified group. The example below shows that network 128.185.184.0 has members of group 224.0.1.1.

Example 2:

```
adv  6  224.0.1.1 128.185.184.114

           For which area [0.0.0.0]?

           LS age:     168
           LS options:  E
           LS type:     6
           LS destination (ID): 224.0.1.1
           LS originator:   128.185.184.114
           LS sequence no:  0x80000001
           LS checksum:     0x7A3
           LS length:       28
           Vertex type: 2
           Vertex ID:   128.185.184.114
```

Vertex type
Describes the object having group members, one of: 1 (the router itself, or stub networks attached to the router) or 2 (a transit network).

Vertex ID
When the vertex type is 1, always the advertising router's ID. When the vertex type is 2, the IP address of the transit network's designated router.

Area Summary

Use the **area summary** command to display the statistics and parameters for all OSPF areas attached to the router.

In the example below, the router attaches to a single area (the backbone area). A simple password scheme is being used for the area's authentication. The router has three interfaces attaching to the area, and has found 4 transit networks, 7 routers and no area border routers when doing the SPF tree calculation for the backbone.

Syntax: area

Example:

```
area

           Area ID      Authentication   #ifcs  #nets  #rtrs  #brdrs
           0.0.0.0       Simple-pass       3      4      7      0
```

# ifcs
Indicates the number of router interfaces attached to the particular area. These interfaces are not necessarily functional.

# nets
Indicates the number of transit networks that have been found while doing the SPF tree calculation for this area.

# rtrs Indicates the number of routers that have been found when doing the SPF tree calculation for this area.

# brdrs Indicates the number of area border routers that have been found when doing the SPF tree calculation for this area.

AS-external advertisements

Use the **AS-external advertisements** command to list the AS external advertisements belonging to the OSPF routing domain. One line is printed for each advertisement. Each advertisement is defined by the following three parameters: its link state type (always 5 for AS external advertisements), its link state ID (called the LS destination), and the advertising router (called the LS originator).

Syntax: as-external

Example: **as-external**

```
Type LS destination LS originator   Seqno      Age   Xsum
  5  0.0.0.0       128.185.123.22  0x80000084 430   0x41C7
  5  128.185.131.0 128.185.123.22  0x80000080 450  0x71DC
  5  128.185.132.0 128.185.123.22  0x80000080 450  0x66E6
  5  128.185.144.0 128.185.123.22  0x80000002 329  0xF2CA
  5  128.185.178.0 128.185.123.22  0x80000081 450  0x72AA
  5  128.185.178.0 128.185.129.40  0x80000080 382  0xDD28
  5  129.9.0.0     128.185.123.22  0x80000082 451  0x4F30
  5  129.9.0.0     128.185.126.24  0x80000080 676  0x324A
  5  134.216.0.0   128.185.123.22  0x80000082 451  0x505A
  5  134.216.0.0   128.185.126.24  0x80000080 676  0x3374
  5  192.9.3.0     128.185.123.22  0x80000082 451  0xF745
  5  192.9.3.0     128.185.126.24  0x80000080 677  0xDA5F
  5  192.9.12.0    128.185.123.22  0x80000082 452  0x949F
  5  192.9.12.0    128.185.128.41  0x80000080 679  0x31B2
  5  192.26.100.0  128.185.123.22  0x80000081 452  0xFDCD
  5  192.26.100.0  128.185.126.24  0x80000080  21  0xDEE8
                              etc.
               # advertisements:             133
               Checksum total:               0x43CC41
```

Type Always 5 for AS external advertisements.

LS destination Indicates an IP network/subnet number. These network numbers belong to other Autonomous Systems.

LS originator Advertising router.

Seqno, Age, Xsum It is possible for several instances of an advertisement to be present in the OSPF routing domain at any one time. However, only the most recent instance is kept in the OSPF link state database (and printed by this command). The LS sequence number (Seqno), LS age (Age) and LS checksum fields (Xsum) are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600.

At the end of the display, the total number of AS external advertisements is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement's LS checksum fields. This information can be used to quickly determine whether two OSPF routers have synchronized databases.

Database Summary

Use the **database summary** command to display a description of the contents of a particular OSPF area's link state database. AS external advertisements are omitted from the display. A single line is printed for each advertisement. Each advertisement is defined by the following three parameters: its link state type (called Type), its link state ID (called the LS destination) and the advertising router (called the LS originator).

Syntax: database *area-id*

Example: `database 0.0.0.0`

```
Type LS destination LS originator    Seqno     Age  Xsum
  1  128.185.123.22 128.185.123.22 0x80000084  442 0xCE2D
  1  128.185.125.38 128.185.125.38 0x80000082  470 0x344D
  1  128.185.126.24 128.185.126.24 0x80000088 1394 0xCC47
  1  128.185.128.41 128.185.128.41 0x80000082  471 0x16A2
  1  128.185.129.25 128.185.129.25 0x8000008D 1624 0x8B64
  1  128.185.129.40 128.185.129.40 0x8000008A 1623 0xABBE
  1  128.185.136.39 128.185.136.39 0x80000082  469 0x5045
  2  128.185.125.40 128.185.129.40 0x80000049  457 0xA31
  2  128.185.126.25 128.185.129.25 0x80000002 1394 0x56B8
  2  128.185.127.24 128.185.126.24 0x8000007F 1031 0x592D
  2  128.185.129.25 128.185.129.25 0x8000005F 2295 0x8219
  2  128.185.129.40 128.185.129.40 0x80000001 1623 0x12C9
  6  224.0.2.6      128.185.142.9  0x8000003D  232 0x513F
  6  224.0.2.6      128.185.184.11 0x80000003  376 0x2250

               # advertisements:            14
               Checksum total:              0x4BBC2
```

Type
Separate LS types are numerically displayed: type 1 (router links advertisements), type 2 (network links advertisements), type 3 (network summaries), type 4 (AS boundary router summaries), and type 6 (group-membership-LSAs).

LS destination
Indicates what is being described by the advertisement.

LS originator
Advertising router.

Seqno, Age, Xsum
It is possible for several instances of an advertisement to be presenting the OSPF routing domain at any one time. However, only the most recent instance is kept in the OSPF link state database (and printed by this command). The LS sequence number (Seqno), LS age (Age) and LS checksum fields (Xsum) are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600.

At the end of the display, the total number of advertisements in the area database is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement's LS checksum fields. This information can be used to quickly determine whether two OSPF routers have synchronized databases.

Note: When comparing multicast-capable to non-multicast routers, the above database checksum (and also # advertisements) will not necessarily match, because non-multicast routers do not handle or store group-membership-LSAs.

Dump Routing Tables

Use the **dump routing tables** command to display all the routes that have been calculated by OSPF and are now present in the routing table. Its output is similar in format to the IP console's dump routing tables command.

Syntax: dump

Example: **dump**

```
Type   Dest net         Mask     Cost Age   Next hop(s)
SPE1   0.0.0.0          00000000  4    3    128.185.138.39
SPF*   128.185.138.0    FFFFFF00  1    1    Eth/0
Sbnt   128.185.0.0      FFFF0000  1    0    None
SPF    128.185.123.0    FFFFFF00  3    3    128.185.138.39
SPF    128.185.124.0    FFFFFF00  3    3    128.185.138.39
SPF    192.26.100.0     FFFFFF00  3    3    128.185.131.10
RIP    197.3.2.0        FFFFFF00  10   30   128.185.131.10
RIP    192.9.3.0        FFFFFF00  4    30   128.185.138.21
Del    128.185.195.0    FFFFFF00  16   270  None

Default gateway in use.

Type Cost Age  Next hop
SPE1 4    3    128.185.138.39

Routing table size: 768 nets (36864 bytes), 36 nets known
```

Type (route type) Indicates how the route was derived.

Sbnt - indicates that the network is subnetted; such an entry is a placeholder only.

Dir - Indicates a directly connected network or subnet.

RIP - Indicates the route was learned through the RIP protocol.

Del - Indicates the route has been deleted.

Stat - Indicates a statically configured route.

BGP - Indicates routes learned through the BGP protocol.

BGPR - Indicates routes learned through the BGP protocol that are readvertised by OSPF and RIP.

Fltr - Indicates a routing filter.

SPF - Indicates that the route is an OSPF intra-area route.

SPIA - Indicates that it is an OSPF inter-area routes.

SPE1, SPE2 - Indicates OSPF external routes (type 1 and 2 respectively).

Rnge - Indicates a route type that is an active OSPF area address range and is not used in forwarding packets.

Dest net IP destination network/subnet.

Mask IP address mask.

Cost Route Cost.

Age For RIP and BGP routes, the time that has elapsed since the routing table entry was last refreshed.

Next Hop IP address of the next router on the path toward the destination host. Also displayed is the interface type used by the sending router to forward the packet.

An asterisk (*) after the route type indicates the route has a static or directly connected backup. A percent sign (%) after the route type indicates that RIP updates will always be accepted for this network/subnet.

A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination. The first hops belonging to these routes can be displayed with the IP console's **route** command.

Interface Summary

Use the **interface summary** command to display statistics and parameters related to OSPF interfaces. If no arguments are given (see Example 1), a single line is printed summarizing each interface. If an interface's IP address is given (see Example 2), detailed statistics for that interface will be displayed.

Syntax: interface *interface-ip-address*

Example 1: **`interface`**

```
Ifc Address      Phys    assoc. Area     Type   State   #nbrs #adjs
9.67.217.66      TKR/0   2.2.2.2        Brdcst   64        0    0
128.185.123.22   PPP/0   0.0.0.0        Brdcst   64        0    0
```

Ifc Address Interface IP address.

Phys Displays the physical interface.

Assoc Area Attached area ID.

Type Can be either Brdcst (broadcast, e.g., an Ethernet interface), P-P (a point-to-point network, e.g., a synchronous serial line), P-2-MP (point-to-multipoint, e.g., a Frame-Relay network), Multi (non-broadcast, multi-access, e.g., an X.25 connection) or VLink (an OSPF virtual link).

State Can be one of the following: 1 (down), 2 (looped back), 4 (waiting), 8 (point-to-point), 16 (DR other), 32 (backup DR) or 64 (designated router).

#nbrs Number of neighbors. This is the number of routers whose hellos have been received, plus those that have been configured.

#adjs Number of adjacencies. This is the number of neighbors in state Exchange or greater. These are the neighbors with whom the router has synchronized or is in the process of synchronization.

Example 2: `interface 128.185.125.22`

```
Interface address:      128.185.125.22
Attached area:          0.0.0.1
Physical interface:     Eth/1
Interface mask:         255.255.255.0
Interface type:         Brdcst
State:                  32
Designated Router:      128.185.184.34
Backup DR:              128.185.184.11

DR Priority:       1  Hello interval:   10  Rxmt interval:     5
Dead interval:    40  TX delay:          1  Poll interval:     0
Max pkt size:   2044  TOS 0 cost:        1

# Neighbors:       0  # Adjacencies:     0  # Full adjs.:      0
# Mcast floods:    0  # Mcast acks:      0

MC forwarding:    on  DL unicast:      off  IGMP monitor:     on
# MC data in:      0  # MC data acc:     0  # MC data out:     0
IGMP polls snt:   75  IGMP polls rcv:    0  Unexp polls:       0
IGMP reports:      0
```

Interface Address Interface IP address.

Attached Area Attached area ID.

Physical interface Displays physical interface type and number.

Interface Mask Displays interface subnet mask.

Interface type Can be either Brdcst (broadcast, e.g., an Ethernet interface), PP (a point-to-point network, e.g., a synchronous serial line), P-2-MP (point-to-multipoint, e.g., a Frame-Relay network), Multi (non-broadcast, multi-access, e.g., an X.25 connection) and VLink (an OSPF virtual link).

State Can be one of the following: 1 (Down), 2 (Attempt), 4 (Init), 8 (2-Way), 16 (ExStart), 32 (Exchange), 64 (Loading) or 128 (Full).

Designated Router IP address of the designated router.

Backup DR IP address of the backup designated router.

DR Priority Displays priority assigned to designated router.

Hello interval Displays the current hello interval value.

Rxmt interval Displays the current retransmission interval value.

Dead interval Displays the current dead interval value.

TX delay Displays the current transmission delay value.

Poll interval Displays the current poll interval value.

Max pkt size Displays the maximum size for an OSPF packet sent out this interface.

TOS 0 cost Displays the interface's TOS 0 cost.

# Neighbors Number of neighbors. This is the number of routers whose hellos have been received, plus those that have been configured.

# Adjacencies Number of adjacencies. This is the number of neighbors in state Exchange or greater.

# Full adj Number of full adjacencies. The number of full adjacencies is the number of neighbors whose state is Full (and therefore, with which the router has synchronized databases).

# Mcast Floods Number of link state updates flooded out the interface (not counting retransmissions).

# Mcast acks Number of link state acknowledgements flooded out the interface (not counting retransmissions).

MC forwarding Displays whether multicast forwarding has been enabled for the interface.

DL unicast Displays whether multicast datagrams are to be forwarded as data-link multicasts or as data-link unicasts.

IGMP monitor Displays whether IGMP is enabled on the interface.

# MC data in Displays the number of multicast datagrams that have been received on this interface and then successfully forwarded.

# MC data acc Displays the number of multicast datagrams that have been successfully forwarded.

# MC data out Displays the number of datagrams that have been forwarded out the interface (either as data-link multicasts or data-link unicasts).

IGMP polls sent Displays the number of IGMP Host Membership Queries that have been sent out the interface.

IGMP polls rcv Displays the number of IGMP Host Membership Queries that have been received on the interface.

Unexp polls Displays the number of IGMP Host Membership Queries that have been received on the interface that were unexpected (i.e., received when the router itself was sending them).

IGMP reports Displays the number of IGMP Host Membership Reports received on the interface.

Nbr node: type and ID Displays the identity of the upstream node if the router were supposed to receive datagrams on this interface. Type here is an integer from 1 to 3, with 1 indicating router, 2 indicating transit net and 3 indicating stub net.

Join

Use the **join** command to establish the router as a member of a multicast group.

This command is similar to the join command in the OSPF configuration console with two differences:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (i.e., a restart/reload is not required).
- The command keeps track of the number of times a particular group is “joined.”

When the router is the member of a multicast group, it responds to pings and SNMP queries sent to the group address.

Syntax: join *multicast-group-address*

Example: `join 224.185.0.0`

Leave

Use the **leave** command to remove a router's membership in a multicast group. This will keep the router from responding to pings and SNMP queries sent to the group address.

This command is similar to the leave command in the OSPF configuration console with two differences:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (i.e., a restart/reload is not required).
- The command will not delete group membership until the “leaves” executed equals the number of “joins” previously executed.

Syntax: leave *multicast-group-address*

Example: `leave 224.185.0.0`

Mcache

Use the **mcache** command to display a list of currently active multicast cache entries. Multicast cache entries are built on demand, whenever the first matching multicast datagram is received. There is a separate cache entry (and therefore a separate route) for each datagram source network and destination group combination.

Cache entries are cleared on topology changes (e.g., a point-to-point line in the MOSPF system going up or down), and on group membership changes.

Syntax: mcache

Example 1: `mcache`

```
          0: TKR/0          1: SDLC/0         2: FR/0
          3: Internal

   Source          Destination     Count   Upst    Downstream
   133.1.169.2     225.0.1.10      8       Local   2 (4),3
   133.1.169.2     225.0.1.20      8       Local   2 (4),3
   3.3.3.3         225.0.1.10      8       2       3
```

Source Source network/subnet of matching datagrams.

Destination Destination group of matching datagrams.

Count Displays the number of received datagrams that have matched the cache entry.

Upst Displays the neighboring network/router from which the datagram must be received in order to be forwarded. When this reads as “none,” the datagram will never be forwarded.

Downstream Displays the total number of downstream interfaces/neighbors to which the datagram will be forwarded. When this is 0, the datagram will not be forwarded.

Mgroups

Use the **mgroups** command to display the group membership of the router's attached interfaces. Only the group membership for those interfaces on which the router is either designated router or backup designated router are displayed.

Syntax: mgroups

Example: **`mgroups`**

```
                 Local Group Database
   Group             Interface                 Lifetime (secs)

  224.0.1.1        128.185.184.11 (Eth/1)       176
  224.0.1.2        128.185.184.11 (Eth/1)       170
  224.1.1.1        Internal                     1
```

Group Displays the group address as it has been reported (via IGMP) on a particular interface.

Interface Displays the interface address to which the group address has been reported (via IGMP).

The router's internal group membership is indicated by a value of “internal.” For these entries, the lifetime field (see below) indicates the number of applications that have requested membership in the particular group.

Lifetime Displays the number of seconds that the entry will persist if Membership Reports cease to be heard on the interface for the given group.

Mstats

Use the **mstats** command to display various multicast routing statistics. The command indicates whether multicast routing is enabled and whether the router is an inter-area and/or inter-AS multicast forwarder. inter-area multicast forwarder.

Syntax: mstats

Example: **`mstats`**

```
                    MOSPF forwarding:       Enabled
                    Inter-area forwarding:  Enabled
                    DVMRP forwarding:       Disabled

      Datagrams received:           2496  Datagrams (ext source):    0
      Datagrams fwd (multicast):       0  Datagrams fwd (unicast):   0
      Locally delivered:               0  No matching rcv interface: 0
      Unreachable source:              3  Unallocated cache entries: 0
      Off multicast tree:              0  Unexpected DL multicast:   0
      Buffer alloc failure:            0  TTL scoping:               0

      # DVMRP routing entries:        0  # DVMRP entries freed:     0
      # fwd cache alloc:              1  # fwd cache freed:         0
      # fwd cache GC:                 0  # local group DB alloc:    0
      # local group DB free:          1
```

MOSPF forwarding Displays whether the router will forward IP multicast datagrams.

Inter-area forwarding Displays whether the router will forward IP multicast datagrams between areas.

DVMRP forwarding Displays whether the router is configured to use DVMRP for multicast routing.

Datagrams received Displays the number of multicast datagrams received by the router (datagrams whose destination group lies in the range 224.0.0.1 - 224.0.0.255 are not included in this total).

Datagrams (ext source) Displays the number of datagrams that have been received whose source is outside the AS.

Datagrams fwd (multicast) Displays the number of datagrams that have been forwarded as data-link multicasts (this includes packet replications, when necessary, so this count could very well be greater than the number received).

Datagrams fwd (unicast) Displays the number of datagrams that have been forwarded as data-link unicasts.

Locally delivered Displays the number of datagrams that have been forwarded to internal applications.

No matching rcv interface Displays the count of those datagrams that were received by a non-inter-AS multicast forwarder on a non-MOSPF interface.

Unreachable source Displays a count of those datagrams whose source address was unreachable.

Unallocated cache entries Displays a count of those datagrams whose cache entries could not be created due to resource shortages.

Off multicast tree Displays a count of those datagrams that were not forwarded either because there was no upstream neighbor or no downstream interfaces/neighbors in the matching cache entry.

Unexpected DL multicast Displays a count of those datagrams that were received as data-link multicasts on those interfaces that have been configured for data-link unicast.

Buffer alloc failure Displays a count of those datagrams that could not be replicated because of buffer shortages.

TTL scoping Indicates those datagrams that were not forwarded because their TTL indicated that they could never reach a group member.

DVMRP routing entries Displays the number of DVMRP routing entries

DVMRP entries freed Indicates the number of DVMRP entries that have been freed.The size will be the number of routing entries minus the number of entries freed.

# fwd cache alloc Indicates the number of cache entries allocated. The current forwarding cache size is the number of entries allocated (“# fwd cache alloc”) minus the number of cache entries freed (“# fwd cache freed”).

# fwd cache freed Indicates the number of cache entries freed. The current forwarding cache size is the number of entries allocated (“# fwd cache alloc”) minus the number of cache entries freed (“# fwd cache freed”).

# fwd cache GC Indicates the number of cache entries were cleared because they were not recently used and the cache overflowed.

# local group DB alloc Indicates the number of local group database entires allocated. The number allocated (“# local group DB alloc”) minus the number freed (“# local group DB free”) equals the current size of the local group database.

# local group DB free Indicates the number of local group database entires freed. The number allocated (“# local group DB alloc”) minus the number freed (“# local group DB free”) equals the current size of the local group database.

The number of cache hits can be calculated as the number of datagrams received (“Datagrams received”) minus the total of datagrams discarded due to “No matching rcv interface,” “Unreachable source” and “Unallocated cache entries,” and minus “# local group DB alloc.” The number of cache misses is simply “# local group DB alloc.”

Neighbor Summary

Use the **neighbor summary** command to display statistics and parameters related to OSPF neighbors. If no arguments are given (see Example 1), a single line is printed summarizing each neighbor. If a neighbor's IP address is given (see Example 2), detailed statistics for that neighbor will be displayed.

Syntax: neighbor *neighbor-ip-address*

Example 1: `neighbor`

```
Neighbor addr   Neighbor ID     State  LSrxl  DBsum  LSreq  Ifc
128.185.125.39  128.185.136.39  128      0     0       0   PPP/1
128.185.125.41  128.185.128.41    8      0     0       0   PPP/1
128.185.125.38  128.185.125.38    8      0     0       0   PPP/1
128.185.125.25  128.185.129.25    8      0     0       0   PPP/1
128.185.125.40  128.185.129.40  128      0     0       0   PPP/1
128.185.125.24  128.185.126.24    8      0     0       0   PPP/1
```

Neighbor addr Displays the neighbor address.

Neighbor ID Displays the neighbor's OSPF router ID.

Neighbor State Can be one of the following: 1 (Down), 2 (Attempt), 4 (Init), 8 (2-Way), 16 (ExStart), 32 (Exchange), 64 (Loading) or 128 (Full).

LSrxl Displays the size of the current link state retransmission list for this neighbor.

DBsum Displays the size of the database summary list waiting to be sent to the neighbor.

LSreq Displays the number of more recent advertisements that are being requested from the neighbor.

Ifc Displays the interface shared by the router and the neighbor.

Example 2: `neighbor 128.185.138.39`

The meaning of most of the displayed fields is given in section 10 of the OSPF specification (RFC 1131).

```
            Neighbor IP address:    128.185.184.34
            OSPF Router ID:         128.185.207.34
            Neighbor State:         128
            Physical interface:     Eth/1
            DR choice:              128.185.184.34
            Backup choice:          128.185.184.11
            DR Priority:            1
            Nbr options:            E,MC


            DB summ qlen:      0  LS rxmt qlen:      0  LS req qlen:
0
            Last hello:        7


            # LS rxmits:     108  # Direct acks:    13  # Dup LS rcvd:
572
            # Old LS rcvd:     2  # Dup acks rcv:  111  # Nbr losses:
29
            # Adj. resets:    30
```

Neighbor IP addr Neighbor IP address.

OSPF router ID Neighbor's OSPF router ID.

Neighbor State Can be one of the following: 1 (Down), 2 (Attempt), 4 (Init), 8 (2-Way), 16 (ExStart), 32 (Exchange), 64 (Loading) or 128 (Full).

Physical interface Displays physical interface type and number of the router and neighbor's common network.

DR choice, backup choice, DR priority Indicate the values seen in the last hello received from the neighbor.

Nbr options Indicates the optional OSPF capabilities supported by the neighbor. These capabilities are denoted by E (processes type 5 externals; when this is not set the area to which the common network belongs has been configured as a stub), T (can route based on TOS) and MC (can forward IP multicast datagrams). This field is valid only for those neighbors in state Exchng or greater.

DBsumm qlen Indicates the number of advertisements waiting to be summarized in Database Description packets. It should be zero except when the neighbor is in state Exchange.

LS rxmt qlen Indicates the number of advertisements that have been flooded to the neighbor, but not yet acknowledged.

LS req qlen Indicates the number of advertisements that are being requested from the neighbor in state Loading.

Last hello Indicates the number of seconds since a hello has been received from the neighbor.

# LS rxmits Indicates the number of retransmissions that have occurred during flooding.

# direct acks Indicates responses to duplicate link state advertisements.

# Dup LS rcvd Indicates the number of duplicate retransmissions that have occurred during flooding.

# Old LS rcvd Indicates the number of old advertisements received during flooding.

# Dup acks rcvd Indicates the number of duplicate acknowledgements received.

# Nbr losses Indicates the number of times the neighbor has transitioned to Down state.

# Adj. resets Counts entries to state ExStart.

Ping

See “Ping” on page 17-7 for an explanation of the **Ping** command.

Traceroute

See “Traceroute” on page 17-9 for an explanation of the **Traceroute** command.

Routers

Use the **routers** command to display all router routes that have been calculated by OSPF and are now present in the routing table. With the **dump routing tables** command, the Net field indicates that the destination is a network. The routers command covers all other destinations.

Syntax: routers

Example: **routers**

```
DType RType Destination       AREA          Cost   Next hop(s)

 ASBR  SPF   128.185.142.9    0.0.0.1         1   128.185.142.9
 Fadd  SPF   128.185.142.98   0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.7    0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.48   0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.111  0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.38   0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.11   0.0.0.1         1    0.0.0.0
   BR  SPF   128.185.142.9    0.0.0.2         1   128.185.142.9
   BR  SPF   128.185.142.9    0.0.0.2         2   128.185.184.114
 Fadd  SPF   128.185.142.47   0.0.0.2         1    0.0.0.0
```

DType Indicates destination type:

Net indicates that the destination is a network
ASBR indicates that the destination is an AS boundary router
ABR indicates that the destination is an area border router
Fadd indicates a forwarding address (for external routes)

RType Indicates route type and how the route was derived:

SPF indicates that the route is an intra-area route (comes from the Dijkstra calculation)

SPIA indicates that it is an inter-area route (comes from considering summary link advertisements).

Destination Destination router's OSPF ID. For Type D entries, one of the router's IP addresses is displayed (which corresponds to a router in another AS).

Area Displays the AS area to which it belongs.

Cost Displays the route cost.

Next hop Address of the next router on the path toward the destination host. A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination.

Size

Use the **size** command to display the number of LSAs currently in the link state database, categorized by type.

Syntax: size

Example: **size**

```
# Router-LSAs:           6
# Network-LSAs:          2
# Summary-LSAs:          45
# Summary Router-LSAs:   6
# AS External-LSAs:      2
# Group-membership-LSAs: 11

# Intra-area routes:     11
# Inter-area routes:     15
# Type 1 external routes: 0
# Type 2 external routes: 2
```

Statistics

Use the **statistics** command to display statistics generated by the OSPF routing protocol. The statistics indicate how well the implementation is performing, including its memory and network utilization. Many of the fields displayed are confirmation of the OSPF configuration.

Syntax: statistics

Example: **statistics**

```
S/W version:            2.1
OSPF Router ID:         128.185.184.11
External comparison:    Type 2
AS boundary capability: Yes
Import external routes: BGP RIP STA DIR SUB
Orig. default route:    No (0,0.0.0.0)
Default route cost:     (1, Type 2)
Default forward. addr:  0.0.0.0

Attached areas:              1  Estimated # external routes: 10
Estimated # OSPF routers:   30  Estimated heap usage:      2368
OSPF packets rcvd:           0  OSPF packets rcvd w/ errs:   0
Transit nodes allocated:     6  Transit nodes freed:         0
LS adv. allocated:          24  LS adv. freed:               2
Queue headers alloc:        32  Queue headers avail:         3
```

```
 # Dijkstra runs:             1  Incremental summ. updates:   0
Incremental VL updates:       0  Buffer alloc failures:       0
Multicast pkts sent:        625  Unicast pkts sent:           0
LS adv. aged out:             0  LS adv. flushed:             0

External LSA database:
Current state:           Normal
Number of LSAs:               9
Number of overflows           0
```

S/W version Displays the current OSPF software revision level.

OSPF Router ID Displays the router's OSPF ID.

External comparison Displays the external route type used by the router when importing external routes.

AS boundary capability Displays whether external routes will be imported.

Import external routes Displays which external routes will be imported.

Orig default route Displays whether the router will advertise an OSPF default route. If the value is “Yes” and a nonzero number is displayed in parentheses, then a default route will be advertised only when a route to the network exists.

Default route cost Displays the cost and type of the default route (if advertised).

Default forward addr Displays the forwarding address specified in the default route (if advertised).

Attached areas Indicates the number of areas that the router has active interfaces to.

Estimated heap usage Rough indication of the size of the OSPF link state database (in bytes).

Transit nodes Allocated to store router links and network links advertisements.

LS adv. Allocated to store summary link and AS external link advertisements.

Queue headers Form lists of link state advertisements. These lists are used in the flooding and database exchange processes; if the number of queue headers allocated is not equal to the number freed, database synchronization with some neighbor is in progress.

# Dijkstra runs Indicates how many times the OSPF routing table has been calculated from scratch.

Incremental summ updates, incremental VL updates Indicate that new summary link advertisements have caused the routing table to be partially rebuilt.

Buffer alloc failures. Indicate buffer allocation failures. The OSPF system will recover from temporary lack of packet buffers.

Multicast pkts sent Covers OSPF hello packets and packets sent during the flooding procedure.

Unicast pkts sent Covers OSPF packet retransmissions and the Database Exchange procedure.

LS adv. aged out Counts the number of advertisements that have hit 60 minutes. Link state advertisements are aged out after 60 minutes. Usually they will be refreshed before this time.

LS adv. flushed Indicates number of advertisements removed (and not replaced) from the link state database.

Incremental ext. updates. Displays number of changes to external destinations that are incrementally installed in the routing table.

External LSA database: Provides information about the LSA database:

Current state Whether the database of current AS external LSAs is in normal or overload state.

Number of LSA The number of external LSAs currently in the database

Number of overflows Number of times the external AS LSA database has entered overload state.

Weight

Use the **weight** command to change the cost of one of the routers OSPF interfaces. This new cost is immediately flooded throughout the OSPF routing domain, causing routes to be updated accordingly.

The cost of the interface will revert to its configured cost whenever the router is restarted or reloaded. To make the cost change permanent, you must reconfigure the appropriate OSPF interface after invoking the weight command. This command will cause a new router links advertisement to be originated, unless the cost of the interface does not change.

Syntax: weight *ip-interface-address new-cost*

Example: `weight 128.185.124.22 2`

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 25. Configuring SNMP

This chapter describes the SNMP configuration commands and includes the following sections:

- "Accessing the SNMP Configuration Environment"
- "SNMP Configuration Commands"

Accessing the SNMP Configuration Environment

To access the SNMP configuration environment, enter the following command at the `Config>` prompt:

```
Config> protocol snmp
SNMP user configuration
SNMP Config>
```

SNMP Configuration Commands

This section summarizes and then explains all the SNMP configuration commands.

Table 25-1 lists the SNMP configuration commands. The SNMP configuration commands allow you to specify parameters that define the relationship between the SNMP agent and the network management station. The information you specify takes effect immediately after a restart or reload of the IBM 2216.

Enter the SNMP configuration commands at the `SNMP Config>` prompt.

Table 25-1. SNMP Configuration Commands Summary

Command	Function
? (Help)	Lists all the SNMP configuration commands or lists the options associated with specific commands.
Add	Adds a community to the list of SNMP communities, an IP address with mask to a community, or a subtree to a MIB view.
Delete	Removes a community from the list of SNMP communities, an IP address with mask from a community, or a subtree from a MIB view.
Enable/Disable	Enables/disables SNMP protocol and traps associated with named communities.
List	Displays the current communities with their associated access modes, enabled traps, IP addresses, and views. Also displays all views and their associated MIB subtrees.
Set	Sets a community's access mode or view. A community's access mode is one of the following: Read and trap generation Read, write and trap generation Trap generation only Also allows setting of trap UDP port.
Exit	Exits the SNMP configuration process and returns to the CONFIG environment.

Table 25-2. SNMP Configuration Commands Options Summary

COMMAND	PARAM 1	PARAM 2	PARAM 3	PARAM 4	DEFAULT
add	community	<comm_name>			None
	address	<comm_name>	<ipAddress>	<ipMask>	
	sub_tree	<view_text_name>	<oid>		
delete	community	<comm_name>			
	address	<comm_name>	<ipAddress>	<ipMask>	
	sub_tree	<view_text_name>	<oid>		
disable	snmp				
	trap	all	<comm_name>		
		cold_start	<comm_name>		
		link_down	<comm_name>		
		link_up	<comm_name>		
		auth_fail	<comm_name>		
		enterprise	<comm_name>		
enable	snmp				
	trap	all	<comm_name>		
		cold_start	<comm_name>		
		link_down	<comm_name>		
		link_up	<comm_name>		
		auth_fail	<comm_name>		
		enterprise	<comm_name>		
list	all				
	community	access			access
		traps			
		address			255.255.255.255
		view			all
	views				
set	community	access	read_trap	<comm_name>	
			write_read_trap	<comm_name>	
			trap_only	<comm_name>	
		view	<community>	all	all
				<view_text_name>	
	trap_port	<udpPort#>			
exit					

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADD
DELETE
SET
ENABLE
DISABLE
LIST
EXIT
```

Add

Use the **add** command to add a community name to the list of SNMP communities, add an address to a community, or assign a portion of the MIB (subtree) to a view.

Syntax: add community
address
sub_tree

community

Use the **add community** command to create a community. It will be created with a default access of read_trap, a view of all, all traps disabled and all IP addresses allowed.

Note: The **add community** command no longer allows you to select access type or trap control. Use the set community access command to assign access types to existing SNMP communities and use the **enable trap** or the **disable trap** command for trap control.

The *community name*. parameter provides the community name used by the SNMP client. This community name is used when accessing the management information base (MIB) in the device from the host specified by the Community IP address parameter.

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

Example: **add community <community_name>**

```
Community Name  []?
```

Community Name Specifies the name of community (32 visual characters maximum). Characters such as spaces, tabs, or <esc> key sequences are not accepted.

address

Use the **add address** command to add to the community definition an address of a network management station in the network that should be allowed to communicate with this box. You must supply the name of the community and the network address (in standard a.b.c.d notation). You also may supply a net mask to restrict access to either an individual host (mask = 255.255.255.255) or to a network of hosts. More than one address can be added to a community; enter the command each time you want to add another address.

If you do not specify an address for a community, requests are handled from any host. Addresses specify hosts that receive the traps. If no address is specified, no trap is generated.

Also, a trap is sent to a specified host only if the associated net mask is defined as 255.255.255.255.

1. The *community name* has:

   Valid Values: A string of 1 to 32 alphanumeric characters.

   Characters such as spaces, tabs, or <ESC> key sequences are not supported.

   Default Value: none

2. The *IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

3. You also may supply a *net mask* to restrict access to either an individual host (mask = 255.255.255.255) or to a network of hosts.

   Valid Values: 0.0.0.0 - 255.255.255.255

   Default Value: none

Example: **add address <community_name> <ipAddress> <ipMask>**

```
Community Name []?
New Address [0.0.0.0]?
```

sub_tree

Use the **add sub_tree** command to add a portion of the MIB to a view or to create a new view. The default is the entire MIB. The **add sub_tree** command is used to manage MIB views. More than one subtree can be added to a view defined by <view_text_name>. To create a new MIB view, issue the **add sub_tree** command with the new view name.

Note: You must assign a view to one or more communities using the **set community view** command to have it take effect. The subtree definitions are inclusive; that is, the subtree OID specified and any OID that is lexicographically greater than the specified OID is considered part of the MIB view.

Valid Values:

- All - Assigns all supported MIB views to the named community.
- View - Assigns a specified MIB view to the named community.

Default Value: All

The *MIB OID name* is the parameter that specifies the MIB Object ID for the sub_tree. This must be entered as a numeric value, not a symbolic value.

This parameter contains a MIB subtree name included in the view defined with the View name parameter. All children of a specified MIB subtree are also included in the view.

For example, to provide a view that would give access to the system group in MIB-II, specify **1.3.6.1.2.1.1**.

Valid Values:

An object identifier in the form of <element1>.<element2>.<element3>..., where:

- You need a minimum of 3 elements.
- You can define a maximum of 49 elements.
- element1 is 0, 1, or 2.
- element2 is an integer between 1 and 40.
- element3 and subsequent elements are integers between 1 and the size of an unsigned byte integer.

Default Value: None

Example: `add sub_tree`

```
View Name  []?
MIB OID name []?
```

View Name Specify the name of the view (32 visual characters maximum). Characters such as spaces, tabs, or <Esc> key sequences are not accepted.

MIB OID Specifies the MIB Object ID for the sub_tree. This must be entered as a numeric value in dotted notation, *not* a symbolic value.

Delete

Use the **delete** command to delete:

- a specific address.
- a community and all of its addresses.
- a subtree from a view.

Syntax: delete community
address
sub_tree

community

Removes a community and its IP addresses. You must supply the community name.

The *community name*.

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

This parameter provides a community name used by the SNMP client. This community name is used when accessing the management information base

(MIB) in the device from the host specified by the Community IP address parameter.

Example: `delete community <community_name>`

address

Removes an address from a community. You must supply the name.

1. The *community name* has:

   Valid Values: A string of 1 to 31 alphanumeric characters.

   Characters such as spaces, tabs, or <ESC> key sequences are not supported.

   Default Value: public

   This parameter provides a community name used by the SNMP client. This community name is used when accessing the management information base (MIB) in the device from the host specified by the Community IP address parameter.

2. The *IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

3. You also may supply a *net mask* to restrict access to either an individual host (mask = 255.255.255.255) or to a network of hosts.

   Valid Values: 0.0.0.0 - 255.255.255.255

   Default Value: none

Example: `delete address <comm_name> <ipAddress> <ipMask>`

sub_tree

Removes a MIB or a portion of the MIB from a view. You must supply the name of the subtree. If all subtrees are deleted, the MIB view is also deleted and all references to it from any associated SNMP communities are removed.

1. The *view name* to be removed is the parameter that allows you to select the view used by the community defined in the Community name parameter. This view determines which MIB objects this community may access. If no view is specified, the community may access all objects known to the router's SNMP agent.

   This parameter should be answered if you decide to restrict a community from accessing the entire MIB managed by the router's SNMP agent.

   You must configure the View name parameter and the MIB Subtree parameter before you can configure this parameter.

   Valid Values:

   - All - Assigns all supported MIB views to the named community.
   - View - Assigns a specified MIB view to the named community.

   Default Value: All

2. The *MIB OID name* is the parameter that specifies the MIB Object ID for the sub_tree. This must be entered as a numeric value, not a symbolic value.

This parameter contains a MIB subtree name included in the view defined with the View name parameter. All children of a specified MIB subtree are also included in the view.

For example, to provide a view that would give access to the system group in MIB-II, specify **1.3.6.1.2.1.1**.

Valid Values:

An object identifier in the form of <element1>.<element2>.<element3>..., where:

- You need a minimum of 3 elements.
- You can define a maximum of 49 elements.
- element1 is 0, 1, or 2.
- element2 is an integer between 1 and 40.
- element3 and subsequent elements are integers between 1 and the size of an unsigned byte integer.

Default Value: None

Example: `delete sub_tree <view_text_name> <oid>`

Disable

Use the **disable** command to disable the SNMP protocol or specified traps on the router.

Syntax: disable snmp
trap

snmp

Disables SNMP

The *community name* has:

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

Example: `disable snmp`

trap

Disables specified traps or all traps. You must specify the trap type from the following options.

Example: `disable trap <trap_type> <community_name>`

Trap Type	Description
all	Disables all traps in a specified community. Specify the community name as part of the command line.
cold_start	Disables cold start traps in a specified community. A cold start trap (0) means that the transmitting router is reinitializing and that the agent's configuration or the protocol entity implementation may be altered. Specify the community name as part of the command line.
link_down	Disables link_down traps in a specified community. A link_down trap (2) recognizes a failure in one of the communication links represented in the agent's configuration. The link_down trap-PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable-bindings.
link_up	Disables link_up traps in a specified community. A link_up trap recognizes that a previously inactive link in the network has come up. The link_up trap-PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable-bindings.
auth_fail	Disables authentication failure traps for a specified community. Authentication failure traps indicate that the sender of the SNMP request does not have the proper permission to talk to this box's SNMP agent.
enterprise	Disables enterprise specific traps in a specified community. Enterprise specific traps indicate that some enterprise specific event has occurred. The specific-trap field identifies the particular trap that occurred. For example, when configured to do so, ELS event messages are sent in enterprise-specific traps.

Enable

Use the **enable** command to enable the SNMP protocol or specified traps on the router.

Syntax: enable snmp
trap

snmp

Enables SNMP

Example: `enable snmp`

trap

Enables specified traps or all traps. You must specify the trap type from the options shown below.

The *community name* has:

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

Example: `enable trap <trap_type> <community_name>`

Trap Type	Description
all	Enables all traps in a specified community. Specify the community name as part of the command line.
cold_start	Enables cold start traps in a specified community. A cold start trap (0) means that the transmitting router is reinitializing and that the agent's configuration or the protocol entity implementation may be altered. Specify the community name as part of the command line.
link_down	Enables link_down traps in a specified community. A link_down trap (2) recognizes a failure in one of the communication links represented in the agent's configuration. The link_down trap-PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable-bindings.
link_up	Enables link_up traps in a specified community. A link_up trap recognizes that a previously inactive link in the network has come up. The link_up trap-PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable-bindings.
auth_fail	Enables authentication failure traps for a specified community. Authentication failure traps indicate that the sender of the SNMP request does not have the proper permission to talk to this box's SNMP agent.
enterprise	Enables enterprise specific traps in a specified community. Enterprise specific traps indicate that some enterprise specific event has occurred. The specific-trap field identifies the particular trap that occurred. For example, when configured to do so, ELS event messages are sent in enterprise-specific traps.

List

Use the **list** command to display the current configuration of SNMP communities, access modes, traps, network addresses, and views.

Syntax: list

all
community
views

list all

Displays the current configuration of SNMP communities for Access, Traps, Address, and View. See the description for the list community command on the next page for details on the options.

Example: `list all`

```
SNMP is enabled.
Trap UDP port: 162

Community Name   Access
public           Read, Write, Trap
oxnard           Read, Trap

Community Name   Enabled Traps
public           Link Down, Cold Restart
oxnard           NONE

Community Name   IP Address  IP Mask
public           All         N/A
oxnard           1.1.1.2     255.255.255.255

Community Name   View
public           All
oxnard           mib2
```

list community *option*

Displays the current attributes of an SNMP community. Options are access, traps, address, view.

Option	**Description**
Access	Displays the access modes for the community.
Address	Displays the network address for the community.
Traps	Displays the types of traps generated for the community.
View	Displays the MIB view for the community.

list community access

Example: **`list community access`**

```
Community Name   Access
public           Read, Write, Trap
oxnard           Read, Trap
```

list community traps

Example: **`list community traps`**

```
Community Name   Enabled Traps
public           Link Down, Cold Restart
oxnard           NONE
```

list community address

Example: **`list community address`**

```
Community Name   IP Address   IP Mask
public           All          N/A
oxnard           1.1.1.2      255.255.255.255
```

list community view

Example: **`list community view`**

```
Community Name   View
public           All
oxnard           mib2
```

list views

Displays the current views for a specified SNMP community.

Example: **`list views`**

```
View Name   Sub-Tree
mib2        1.3.6.1.2.1
```

Set

Use the **set** command to assign a MIB view to a community, to set the SNMP UDP trap port number, or set the access mode of the community.

Syntax: set community access
community view
trap_port

community access

Use the **set community access** command to assign one of three access types to a community. You must supply the name of the community and the access type.

The *community name* has:

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

Example: `set community access <options> <comm_name>`

Options	Description
read_trap	Sets read access and trap generation to the named community.
write_read_trap	Sets write and read and trap generation access to the community specified.
trap_only	Indicates the community is used only when sending an SNMP trap.

community view

Use the **set community view** command to assign a MIB view to a community.

Example: `set community view <comm_name> <options>`

Options	Description
all	Allows access to all MIB objects for the named community. All is the default.
view_text_name	Assigns a specified MIB view to the named community.

trap_port

Use the **set trap_port** command to specify a UDP port number, other than the default standard port 162, to send traps to. The default is the standard port.

Example: `set trap_port <udpport#>`

UDP Port Number	Specifies a User Datagram Protocol port other than the standard UDP port (default # 162).

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 26. Monitoring SNMP

This chapter describes the SNMP console commands and includes the following sections:

- “Accessing the SNMP Console Environment”
- “SNMP Console Commands”

Accessing the SNMP Console Environment

To access the SNMP console environment, enter the following command at the **+** (GWCON) prompt:

```
+ protocol snmp
SNMP>
```

SNMP Console Commands

This section summarizes and then explains all of the SNMP console commands.

Table 26-1 on page 26-2 lists the SNMP console commands. The SNMP console commands allow you to view the parameters of the SNMP configuration and display some statistics relating to the SNMP agent.

Temporary changes to the runtime SNMP parameters can be made through the console. They will immediately affect the operation of the SNMP agent. If you want to make the temporary changes permanent, then use the SAVE command. If the original SNMP configuration needs to be restored, use the REVERT command. This feature allows you to temporarily alter the behavior of the SNMP agent, without permanently changing the configuration. For the temporary changes to take affect, you must EXIT the SNMP console process.

Enter the SNMP console commands at the `SNMP>` prompt.

Table 26-1. SNMP Console Command Summary

Command	Function
? (Help)	Lists all the SNMP console commands or lists the options associated with specific commands.
Add	Adds a community to the list of SNMP communities, an IP address with mask to a community, or a subtree to a MIB view.
Delete	Removes a community from the list of SNMP communities, an IP address with mask from a community, or a subtree from a MIB view.
Enable/Disable	Enables/disables SNMP protocol and traps associated with named communities. These actions are only allowed in the SNMP Configuration environment.
List	Displays the current configuration of SNMP communities, views, access modes, traps, and network addresses.
Revert	Erases the specified changes and restores the settings to the values in the permanent SNMP configuration.
Save	Takes the specified changes and saves then permanently in the SNMP configuration.
Set	Sets a community's access mode or view. A community's access mode is one of the following: • Read and trap generation • Read, write and trap generation • Trap generation only Also allows setting of trap UDP port.
Statistics	Displays statistics about the SNMP agent.
Exit	Exits the SNMP console process and returns to the GWCON environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
list
statistics
exit
```

Add

Use the **add** command to add a community name to the list of SNMP communities, add an address to a community, or assign a portion of the MIB (subtree) to a view.

For information on using the **add** command, see “add” on “Add” on page 25-3.

Delete

Use the **delete** command to delete:

- A specific address.
- A community and all of its addresses.
- A subtree from a view.

For information on using the **delete** command, see "Delete" on page 25-5.

Disable

Use the **disable** command to disable the SNMP protocol or specified traps on the router. This command is available only in the SNMP Configuration environment.

For information on using the **disable** command, see "Disable" on page 25-7.

Enable

Use the **enable** command to enable the SNMP protocol or specified traps on the router. This command is available only in the SNMP Configuration environment.

For information on using the **enable** command, see "Enable" on page 25-8.

List

Use the **list** command to display the current configuration of SNMP communities, views, access modes, traps, and network addresses.

Syntax: list all
community
views

list all

Displays the current configuration of SNMP communities for Access, Traps, Address, and View. See the description for the list community command on the next page for details on the options.

Example: `list all`

```
SNMP is enabled.
Trap UDP port: 162

Community Name   Access
public           Read, Write, Trap
oxnard           Read, Trap


Community Name   Enabled Traps
public           Link Down, Cold Restart
oxnard           None


Community Name   IP Address  IP Mask
public           All         N/A
oxnard           1.1.1.2     255.255.255.255


Community Name   View
public           All
oxnard           mib2


View Name        Sub-Tree
mib2             1.3.6.1.2.1
```

list community option

Displays the current attributes of a specified SNMP community. Options are access, traps, address, view.

Example: **list community** option

Option	Description
Access	Displays the access modes for the community.
Address	Displays the network address for the community.
Traps	Displays the types of traps generated for the community.
View	Displays the MIB view for the community.

list community access

Example: **list community access**

```
Community Name   Access
public           Read, Write, Trap
oxnard           Read, Trap
```

list community traps

Example: **list community traps**

```
Community Name   Enabled Traps
public           Link Down, Cold Restart
oxnard           None
```

list community address

Example: **list community address**

```
Community Name   IP Address   IP Mask
public           All          N/A
oxnard           1.1.1.2      255.255.255.255
```

list community view

Example: **list community view**

```
Community Name   View
public           All
oxnard           mib2
```

list views

Displays the current views for a specified SNMP community.

Example: **list views**

```
View Name   Sub-Tree
mib2        1.3.6.1.2.1
```

Revert

Use the **revert** command to erase the specified changes and restore the settings to the values in the permanent SNMP configuration.

Save

Use the **save** command to save the specified changes permanently.

Set

For information on using the **set** command, see “Set” on page 25-11.

Statistics

Use the **statistics** command to display statistics about the SNMP agent.

Syntax: statistics

Example:

```
statistics
        SNMP memory in use = 9416
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example:

```
exit
```

Chapter 27. Using and Configuring DLSw

This chapter describes the Data Link Switching (DLSw) configuration commands, and describes the implementation of the Data Link Switching (DLSw) protocol. Changes made at the `Config>` prompt do not take effect immediately, but become part of the SRAM configuration used for subsequent restarts of the router. For a description of temporary, but immediate, configuration changes, see page 28-1.

The 2216 offers a wide range of function that enables you to integrate Systems Network Architecture (SNA) and Network Basic Input/Output System (NetBIOS) traffic into heterogeneous, wide area networks.

The following sections explain how to configure your router for DLSw:

- “About DLSw”
- “Using DLSw Features” on page 27-3
- “Setting up DLSw” on page 27-18
- “Sample DLSw Configuration” on page 27-23
- “Accessing the DLSw Configuration Environment” on page 27-35
- “Pre-Configuration Requirements” on page 27-36
- “DLSw Configuration Commands” on page 27-37

About DLSw

DLSw is a forwarding mechanism for the LLC2, SDLC, and QLLC (SNA over X.25) protocols. It relies on the bridging function of the router, the Switch-to-Switch protocol (SSP), and TCP/IP to provide a reliable transport of SNA traffic over an internet. DLSw does not provide full routing capabilities, but it provides switching at the data link layer. Rather than bridging LLC2 frames, DLSw encapsulates their data in TCP frames and forwards the resulting messages over the WAN link to a peer DLSw router for delivery to their intended end-station addresses.

How DLSw Works

LLC2, SDLC, and QLLC are connection-oriented protocols. DLSw provides them with dynamic characteristics of routable protocols ***and*** preserves both end-to-end reliability and control features for effective communication.

Problems in the Bridging Solution

Figure 27-1 on page 27-2 illustrates the traditional approach to bridging LLC2 frames across WAN links. With this approach, network delays occur much more frequently in the WAN than on a LAN. These delays can result from simple network congestion, slower line speeds, or other factors. Whatever the cause, these delays increase the possibilities of session timeouts and of data not reaching intended destinations.

Also, LAN protocols, like LLC2, use significantly shorter retransmit/response times than the WAN’s. Thus, end-to-end connections across a WAN link are extremely difficult to maintain, and session timeouts are much more probable.

In addition to session timeouts, there is a significant problem when data is delayed while crossing the WAN. A sending station can resend data that is delayed (but not lost); this can result in LLC2 end stations receiving duplicate data. Duplicate data can cause confusion for LLC2 procedures on the receiving side which can, in turn, result in inefficient use of the WAN link.

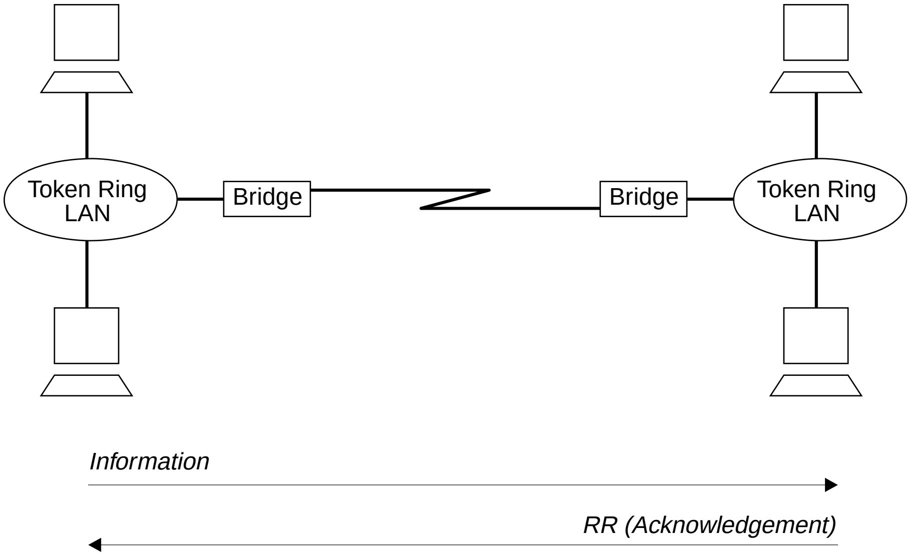


Figure 27-1. Traditional Approach to Bridging Across WAN Links

The preceding example shows traditional bridging, involving end-to-end data-link control. As a connectionless protocol, bridging does nothing to ensure the integrity of LLC traffic on the WAN.

Protocol Spoofing

To reduce the chance of session timeouts, and to maintain the appearance of end-to-end connectivity for sending stations, DLSw works by terminating or “spoofing” LLC2 connections at the local router. Upon receiving an LLC2 frame, the router sends an acknowledgement to the sending station. This acknowledgment tells the sender that data that was previously transmitted has been received.

The acknowledgment prevents the station from retransmitting. From this point forward, assuring that data gets through is the responsibility of the DLSw software. The software accomplishes this by encapsulating the data in routable IP frames, then transports them (via TCP) to a DLSw peer. The peer DLSw router strips away the TCP headers, determines the address of the data’s intended recipient, and establishes a new LLC2 connection with that end station.

Figure 27-2 on page 27-3 illustrates this relationship between two DLSw peer routers, each attached to a Token-Ring Network.

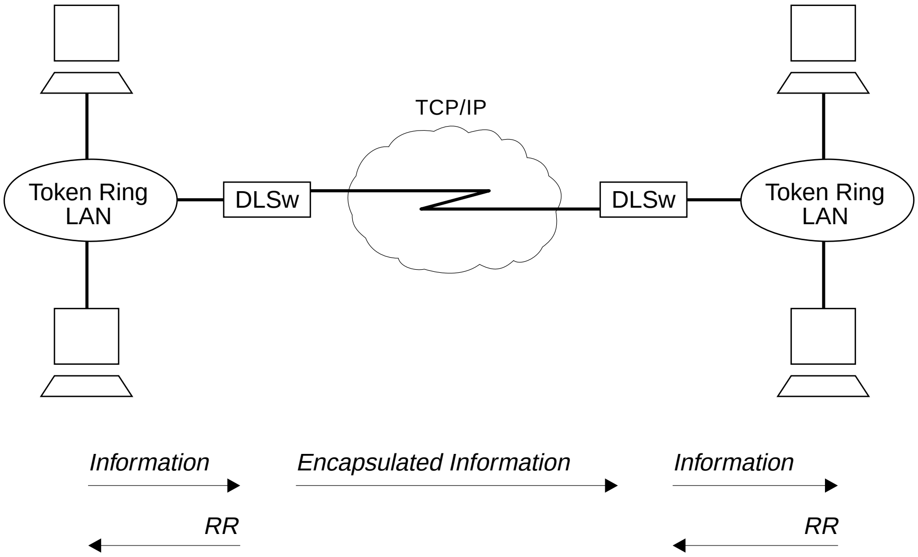


Figure 27-2. Data Link Switching over the WAN

DLSw terminates the LLC2 connection at the router. This means that LLC2 connections do not cross the wide area network. This reduces session timeouts and the acknowledgments (RRs) that would otherwise traverse the wide area's area links.

Benefits of DLSw

Because DLSw terminates the DLC connection at the local router (see Figure 27-2), it is especially effective at eliminating SNA session timeouts and reducing WAN overhead on shared circuits. The protocol has these main benefits:

- Reduces the possibility of session timeouts by terminating LLC2, SDLC, QLLC, and NetBIOS control traffic at the local router.
- Reduces WAN network overhead by eliminating the need to transmit acknowledgments (RRs) over the wide area. The RRs are confined to the LANs local to each DLSw router.
- Provides flow and congestion control, and broadcast control of search packets, between DLSw routers and their attached end stations.
- Increases Source Route Bridging hop-count limits.
- Allows protocol conversion among LLC2, SDLC, and QLLC.
- Supports NetBIOS traffic.

Using DLSw Features

The following sections address the use of various DLSw features:

- “TCP Connections, Neighbor Discovery, and Multicast Exploration” on page 27-4
- “LLC Device Support” on page 27-6
- “SDLC Device Support” on page 27-6
- “QLLC Device Support” on page 27-10
- “APPN Interface Support” on page 27-15
- “Using the Neighbor Priority Feature” on page 27-16

- “Balancing SNA and NetBIOS Traffic” on page 27-17

TCP Connections, Neighbor Discovery, and Multicast Exploration

DLSw uses TCP to provide reliable, sequenced delivery of end-user information across an IP network. DLSw message formats allow multiple end-station sessions, or circuits, to be carried across a single TCP transport connection. There are two ways to configure which DLSw-capable routers should have TCP transport connections between them to allow the desired end-station connectivity:

- Configure the IP addresses of the neighbor router at one or both of each pair of routers. This is the most basic method and is supported by all DLSw router vendors.
- Configure multicast group membership at each router, allowing the routers to discover each others' IP addresses dynamically. This is a special feature of this product's DLSw, to ease the burden of configuring neighbor IP addresses.

Configuring TCP Neighbors

To configure a neighbor IP address at a router, use the **add tcp** command once for each of that router's neighbors. It is not required for each of the two routers in a neighbor relationship to configure the other's IP address. Only one router needs to have the other's address, and the other router can be configured to accept dynamic TCP connections from non-configured neighbors. Use the **enable dynamic-neighbors** command to configure this behavior, and use the **set dynamic-tcp** command to configure the parameters used for these dynamic connections. Enabling dynamic TCP connections can be particularly useful for “hub” routers that you do not want to reconfigure when you set up new remote branch office routers that connect to the hub.

In addition to the IP address, the **add tcp** command allows you to configure a number of parameters for the neighbor and the TCP connection itself. The *keepalive* parameter controls whether the TCP layer occasionally polls its peer TCP layer in the absence of any user data traffic. Enabling keepalive messages results in more timely notification of TCP connection failure, but can increase WAN overhead and cause the reporting of failures that could have been successfully re-routed.

The *connectivity setup type* parameter controls when DLSw brings up and takes down the TCP connection. When one or both neighbors have the CST set to *active*, DLSw attempts to bring up the connection at system startup and at regular intervals until it is up. Once the TCP connection is established, DLSw attempts to keep it up at all times by trying to bring it back when it fails. If both neighbors have set the CST to *passive*, DLSw brings up the TCP connection only when it is actually needed to establish a DLSw end-station session. When the last DLSw session ends and no new session is started in a configurable period of time (the *neighbor inactivity timer*), DLSw disconnects the TCP connection and frees the associated internal resources.

Configuring Groups for Neighbor Discovery

To avoid configuring neighbor IP addresses in one or both of every pair of neighbor routers, set up DLSw to use multicast IP to discover the IP address of the neighbors to which it should connect. Use the **join-group** command at each router to make it a member of one or more DLSw groups and to assign a role within the group. The role may be “client,” “server,” or “peer.” DLSw uses multicast IP to discover the IP addresses of all DLSw routers that are members of the same groups and that have the complementary role (that is, clients discover servers within a group and vice versa, and peers discover other peers).

When DLSw learns the IP addresses of its neighbors in each group, it uses the “connectivity setup type” of its membership in the group and that of each group neighbor to determine when a TCP connection to that neighbor should be brought up. As with configured individual neighbors, when either CST is *active*, DLSw brings up the TCP connection to the discovered neighbor as soon as possible and attempts to keep the connection up at all times. When both CSTs are *passive*, DLSw brings up the TCP connection only when it is required to carry DLSw sessions, and uses the *neighbor inactivity timer* to disconnect the TCP connection when it is not being used.

Multicast Exploration and Frame Forwarding

DLSw uses multicast IP services for more than discovering the IP addresses of neighbor routers. It uses these same services to forward DLSw messages searching for end-station resources (for example, MAC addresses or NetBIOS names), and to forward NetBIOS datagram traffic. This feature can dramatically increase the scalability of DLSw networks because there is no need for static TCP connections to all neighbors to carry search and datagram messages. Also, DLSw does not need to send a different copy of each broadcast message on every TCP connection, but can send a single copy that is replicated within the multicast IP infrastructure.

To use multicast IP for exploration and frame forwarding, issue the **join-group** command and set the *connectivity setup type* to *passive*. DLSw automatically determines which other group members are multicast-capable, and which are using their group membership simply to discover neighbor IP addresses and bring up static TCP connections. DLSw simultaneously works with both types of neighbors when searching for end-station resources, forwarding NetBIOS datagrams, and establishing DLSw sessions.

When you issue the **join-group** command, you select one of two addressing methods to describe the group you are joining. When you provide a group ID and the client/server/peer role as previously described, the router constructs the corresponding multicast IP addresses and can communicate with other IBM routers that use this method. You may also choose to directly specify the multicast IP addresses to be used and whether each address should be read from, written to, or both. This method was introduced to support the APPN Implementers Workshop DLSw Version 2 Standard, and allows multicast interoperability with other DLSw Version 2 compliant products.

A given router may be a member of traditional groups and concurrently read from and write to DLSw Version 2 multicast addresses. The new multicast addresses may also be used for neighbor discovery, but you must ensure that for every pair of routers intended to form a TCP connection, one router has a *connectivity setup type* of *active* on a write-capable address on which the other router is reading. Whether you are doing neighbor discovery or not, specifying multicast addresses requires more careful configuration planning to ensure reachability than using group IDs and the client/server/peer model.

LLC Device Support

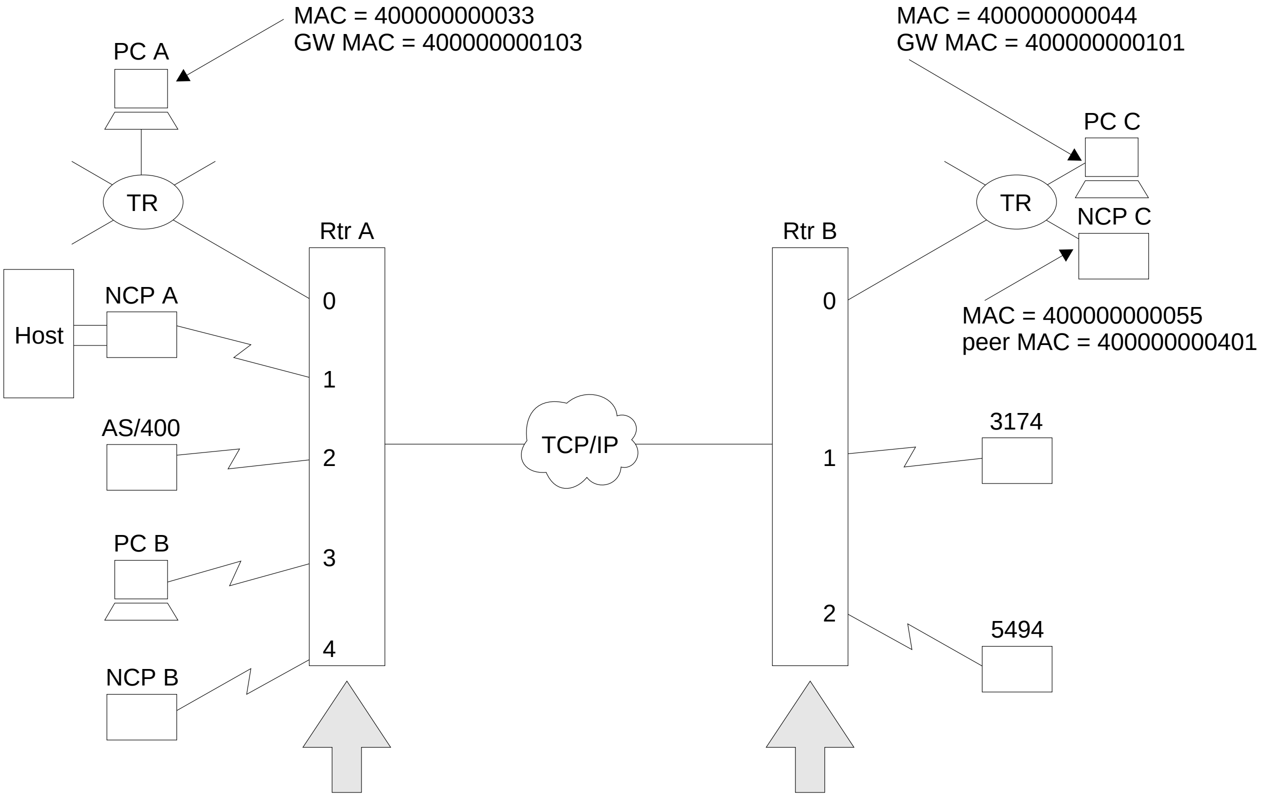
DLSw supports SNA and NetBIOS end stations attached to the router via LAN and remote-bridging WAN interfaces. These end stations and the router are both running ISO 8802-2 (IEEE 802.2) standard Logical Link Control (LLC) to provide data sequencing and reliable delivery. The router currently supports bridged LLC traffic over the following interface types, and all can be used for traffic flowing between DLSw and LLC end stations:

- Token-ring
- Ethernet/802.3
- FDDI
- Frame Relay (using RFC 1490 bridged frame formats)
- PPP
- Dial circuits that use PPP or FR framing (for example, ISDN)
- ATM (as a LAN Emulation client)

Because DLSw uses the MAC and SAP addresses available in bridged frames, there is no need to configure in DLSw any information about individual LLC end stations. DLSw receives broadcast traffic sent by these end stations, and uses normal LAN/bridge broadcast methods to make initial contact with them. You must, however, configure the bridging support for any interface that DLSw is to use, and configure within DLSw the SAPs that it is to use on each interface.

SDLC Device Support

DLSw supports SDLC end stations that may be SNA PU types 2.0, 2.1, 4 (for NCP-NCP traffic), or 4/5 (a host or NCP performing the SNA boundary function). The router can serve in either a primary or secondary SDLC link station role, based on the role configured for the SDLC interface, or based on SNA XID negotiation. In the primary role, the router can support multiple SDLC devices of differing PU types on the same physical multipoint SDLC line. In the secondary role, the router can represent multiple SDLC secondary stations on a single physical SDLC interface. It also supports the IBM 3174 Group Poll function in the secondary role.



Interface	SDLC Addr	PU Type	Source MAC	Dest MAC
1 (sec)	01	2	400000000101	400000000044
	02	5	400000000102	4011111101C1
	03	5	400000000103	400000000033
	04	2	400000000104	400000000301
2 (neg)	01	2	400000000201	4011111102C1
3 (neg)	01	2	400000000301	000000000000
4 (neg)	01	4	400000000401	400000000055

Interface	SDLC Addr	PU Type	Source MAC	Dest MAC
1 (pri)	C1	2	4011111101C1	000000000000
2 (neg)	C1	2	4011111102C1	400000000201

Figure 27-3. Example DLSw SDLC Configurations

Figure 27-3 illustrates some of the SDLC configurations supported by DLSw, and shows a subset of the DLSw configuration required to map between SDLC and DLSw (MAC and SAP) addresses. The diagram shows both *local* (within a single router) and *remote* (across two routers and an IP network) DLSw sessions.

The following DLSw sessions are configured:

- NCP A to PCs A, B, and C, and to the 3174

  For NCP A to be able to communicate with these 4 PUs, Router A must have a secondary link station configured on Interface 1 for each PU. This interface should be configured in SDLC as secondary, full-duplex, and point-to-point. Group poll is recommended whenever there are several secondary stations on the same interface, to reduce non-productive polling.

  In this example, NCP A communicates to PC C via SDLC station address 01, to the 3174 via address 02, to PC A via address 03, and to PC B via address 04. Note that the PC A and C sessions both involve SDLC-to-LLC conversion, in a local and remote configuration, respectively. The session to PC B is a local SDLC-to-SDLC session, which may be unusual.

  For the secondary link stations defined in Router A, a PU type of 5 indicates that the SDLC device is a host (here front-ended by a controller) performing the

SNA BNN function to a downstream PU2.0 device. A PU type of 2 here indicates that the SDLC host/FEP is acting as a T2.1 node communicating with another T2.1 node in the DLSw network.

- AS/400 to the 5494

  Here, these devices are to function as T2.1 nodes and the SDLC links on their respective routers are configured as negotiable (T2.1 nodes are also supported on fixed-role links, and DLSw restricts role negotiation accordingly). The stations will perform full XID negotiation, including role determination and SDLC address resolution (if the router and the end station on the same link is each configured with different SDLC station addresses). Note that there is no relationship in remote SDLC-SDLC configurations between the SDLC station addresses used on the two different SDLC links. Remote SDLC-to-LLC sessions are also supported between T2.1 devices.

- NCP B to NCP C

  NCP B is configured as PU Type 4, indicating that this DLSw session is to carry INN subarea traffic between NCPs, and not BNN traffic from an NCP to a PU 2 device. The example shows a remote SDLC-to-LLC session, but like-to-like sessions are also supported. DLSw INN function does not support multilink TGs or the NCP remote load/dump functions.

Address Mapping

DLSw configuration provides a mapping between single-byte SDLC station addresses and the MAC addresses and SAPs by which DLSw identifies end stations. The Source MAC address for an SDLC station represent the SDLC device to the rest of the DLSw network. It is the source address for frames coming from the device, and the destination address for frames going to the device. A Source MAC address is required for the SDLC device to be able to communicate through DLSw.

The Destination MAC address specifies the end station in the DLSw network to which this SDLC device should be connected when it starts to communicate. SDLC devices that are always to be the target of new sessions and never the initiator should have a zero destination MAC address. When the router is configured as a secondary link station, it is important to define a destination MAC address so that a host connect-out will be successful. This is because a secondary link station cannot initiate a contact to the host on behalf of a remote DLSw end station connecting in, but must wait to be polled. Note that when the remote DLSw end station is itself SDLC (for example, the 3174 on Router B in Figure 27-3 on page 27-7) and is paired with a local secondary station, the remote station may have a zero destination MAC address to reflect this dependency on a host connect-out.

DLSw Configuration and SDLC Configuration

To use DLSw over an SDLC interface, you configure the address mapping as part of DLSw configuration, and you also configure some information as part of SDLC configuration. As a minimum in SDLC, you must set the interface to be SDLC and configure other interface-level parameters such as the link role. SDLC interface parameters provide default values for all SDLC link stations on that interface, but if you wish to have unique values for a station, you can configure individual SDLC station information.

The address pair *interface number, SDLC station address* is the common key that links DLSw address-mapping information to the station-level configuration in SDLC. Router software make this association at initialization time. If DLSw attempts to initialize a link station whose SDLC station address is not configured in SDLC on the interface that DLSw specifies, SDLC creates a link station definition dynamically and uses the parameter defaults defined in SDLC for that interface.

Relationship to the SDLC Relay Function

SDLC Relay is a router function that encapsulates whole SDLC frames in IP packets, which are then routed to another router that also supports SDLC Relay. The destination router strips off the IP header and delivers the SDLC frames unmodified onto a destination SDLC link.

This function differs from DLSw SDLC support in the following ways:

- With SDLC Relay, there is no SDLC link station operating within the router. Control frames (for example, RR) flow across the IP network. With DLSw, the router's SDLC support terminates the SDLC connection. Only the data from the SDLC frames flows across the IP network. As a result, DLSw may provide better WAN bandwidth utilization, and is less sensitive to link timeouts due to WAN delays.
- SDLC data and control frames pass transparently through SDLC Relay, while DLSw needs to interpret and modify some of them. Along with the fact that DLSw terminates the SDLC connection, this means that certain product configurations and functions (for example, multilink TGs between NCPs) are not supported by DLSw.
- SDLC Relay requires that the data type of both communicating end stations be SDLC. DLSw provides a protocol conversion function, so the data type of the other end station might be LLC, SDLC, QLLC, or any other data type supported by a DLSw product.
- DLSw is a standard developed by the APPN Implementers Workshop and documented in an IETF RFC. As such, it is supported by a number of vendors. SDLC Relay is currently supported only in certain IBM and compatible router products.

You must use DLSw when:

- You require protocol conversion from SDLC to LLC or QLLC
- You want to restrict control traffic (for example, RR frames) to flow outside the IP network

You must use SDLC Relay when:

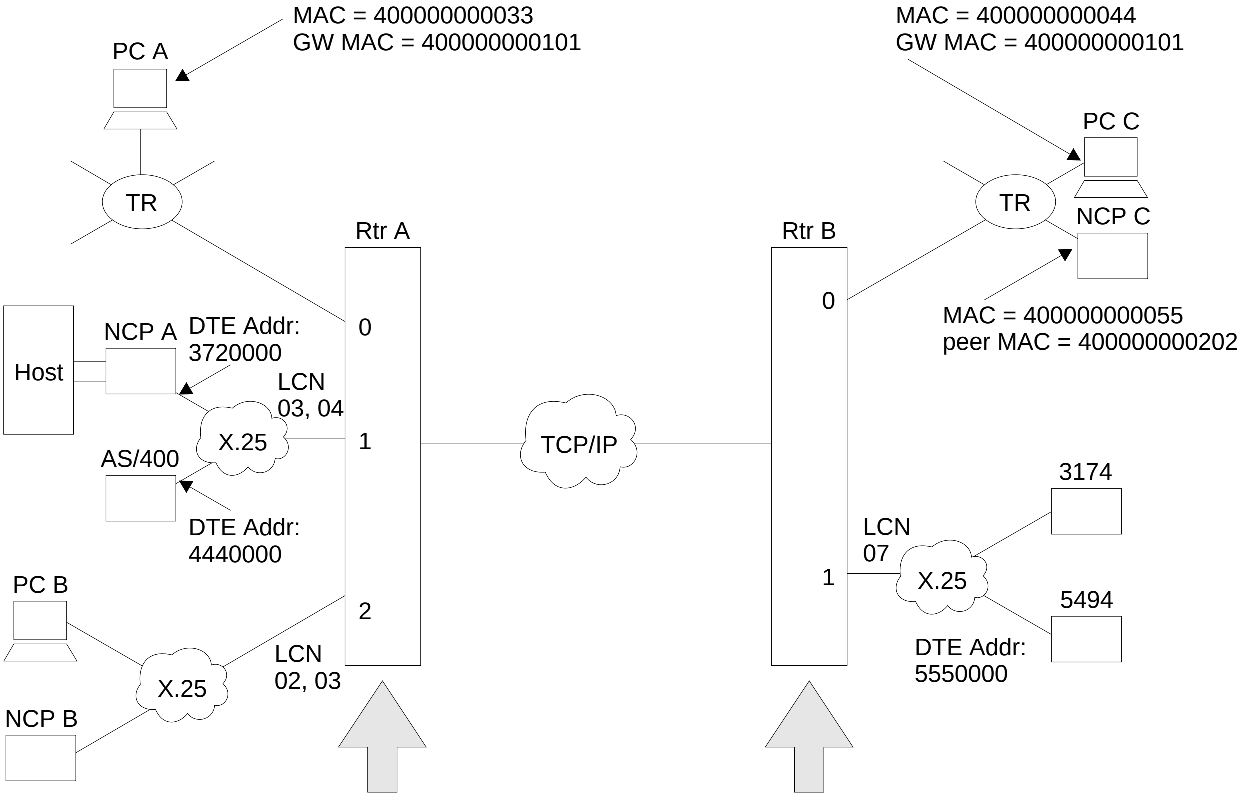
- You need one of the SDLC-SDLC functions or configurations that is not currently supported by DLSw

In other SDLC-SDLC configurations, choose the function that best meets your requirements for ease of configuration, WAN utilization, and support for your current end station environment. For more information on SDLC Relay, refer to *Software User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1*.

QLLC Device Support

QLLC is a protocol that operates above the packet layer protocol of X.25 to provide an SDLC-like station appearance to SNA devices on X.25 networks. QLLC supports a single SNA PU per virtual circuit (either PVC or SVC). X.25 channel multiplexing provides for the attachment of many virtual circuits or PUs through a single physical interface to the X.25 network. QLLC architecture defines primary, secondary, and peer station roles, but these are less important than in SDLC because they do not affect the transmission of end-user data. The data for all virtual circuits on an interface flows on a single LAPB layer-2 link connection, which operates in a balanced mode. Either side has permission to send at all times while the link is connected.

DLSw supports QLLC end stations that may be SNA PU types 2.0, 2.1, 4 (for NCP-NCP traffic), or 4/5 (a host or NCP performing the SNA boundary function). End stations may be attached via configured PVCs, configured SVCs, or dynamic SVCs resulting from an incoming call. The router can resolve to either a primary or secondary QLLC link station role, based on the role configured for the X.25 interface and based on SNA XID negotiation. Different PU types may co-exist on different virtual circuits within the same physical interface, but only a single link station role is supported per interface.



Interface	VC Type	LCN/DTE Addr	PU Type	Source MAC	Dest MAC
1 (neg)	PVC	03	5	400000000102	401111110101
	PVC	04	5	400000000105	400000000201
	SVC	3720000	5	400000000101	000000000000
	SVC	4440000	2	400000000104	000000000000
2 (neg)	PVC	02	2	400000000201	400000000105
	PVC	03	4	400000000202	400000000055

Interfaces enabled for incoming calls
1

Connection ID	Dest MAC
PCA	400000000033
PCC	400000000044

Interface	VC Type	LCN/DTE Addr	PU Type	Source MAC	Dest MAC
1 (pri)	PVC	07	2	401111110101	400000000102

Interfaces enabled for incoming calls
1

Connection ID	Dest MAC
AS400	400000000104

Figure 27-4. Example DLSw QLLC Configurations

Figure 27-4 illustrates some of the QLLC configurations supported by DLSw, and shows a subset of the DLSw configuration required to map between QLLC and DLSw (MAC and SAP) addresses. The diagram shows both *local* (within a single router) and *remote* (across two routers and an IP network) DLSw sessions. No QLLC-to-SDLC pairings are shown, but these are supported in both local and remote configurations.

The following DLSw sessions are configured:

- NCP A to PCs A, B, and C, and to the 3174

  NCP A is attached to Interface 1 on Router A via 2 PVCs and 2 SVCs, each virtual circuit representing one PU. PVCs are addressed within an interface by a *Logical Channel Number*, and SVCs by the DTE address (phone number) of the attached X.25 device. As with SDLC, DLSw configuration maps these "native" DLC addresses (LCN or DTE address) to DLSw addresses (MACs and SAPs).

In this example, NCP A communicates with the 3174 (remote QLLC-QLLC) via PVC 03, and with PC B (local QLLC-QLLC) via PVC 04. These LCNs are actually local to Router A; NCP may use different LCNs for its corresponding PVCs into the X.25 network. Router A connects NCP A with PC C (remote QLLC-LLC) and with PC A (local QLLC-LLC) using two SVCs between the DTE address 3720000 for NCP A and the DTE address for interface 1 on Router A. Since Router A needs to be able to accept calls from NCP A, it has Interface 1 enabled for incoming calls to DLSw. NCP A uses *Connection IDs*, discussed below, to connect out to PCs A and C.

In Router B, PC C is not configured because it is LLC/LAN-attached. The 3174 is connected via Interface 1 LCN 07, which has no relation to the Interface or LCN number used at Router A.

- AS/400 to the 5494

  In addition to NCP A, the AS/400 is also attached to Router A via Interface 1. Unlike SDLC, there is no performance advantage to limiting the number of stations on a given interface. There can be multiple stations on a link regardless of the link role. If the role is negotiable and the stations are T2.1 or PU4 nodes, each station can negotiate independently to become primary or secondary.

  The AS/400 has no destination MAC address configured in Router A, and therefore cannot connect out to the 5494. The 5494 is not configured in Router B, and will therefore be a dynamic SVC. The 5494 uses a Connection ID to indicate that it wants to be connected to the AS/400. Router B has Interface 1 enabled for incoming calls to DLSw so that it can receive calls from the 5494.

- NCP B to NCP C

  NCP B is configured as PU Type 4, indicating that this DLSw session is to carry INN subarea traffic between NCPs, and not BNN traffic from an NCP to a PU 2 device. The example shows a remote QLLC-to-LLC session, but like-to-like sessions and sessions involving SDLC are also supported. DLSw INN function does not support multilink TGs or the NCP remote load/dump functions.

Address Mapping

DLSw provides a mapping between the MAC/SAP pairs used to address end-station entities in the DLSw domain, and the *interface, LCN* (PVC) or *interface, DTE address* (SVC) pairs used in the X.25 domain. This mapping takes place at connection-establishment time, but uses addressing information configured in the router and in end station products.

Connect-out (to QLLC stations): DLSw receives a CUR_ex or CUR_cs message addressed to a particular target MAC and SAP. It searches among its QLLC end stations for one whose SMAC and SSAP (SAP is only checked for CUR_cs) match this target MAC/SAP. There should be either one or no matches, since SMACs are unique within the router.

If a match is found, DLSw initiates connection establishment with the QLLC station using the corresponding interface and LCN for a PVC, or the interface and phone number for an SVC. DLSw can place multiple outgoing calls to the same DTE address using a single QLLC station (SVC) definition. This allows many DLSw devices to connect to the same destination with a minimum of configuration effort.

Connect-in (from QLLC stations): For **PVCs**, QLLC receives a frame that starts circuit establishment from the attached end station. QLLC and DLSw match the interface and LCN on which the frame was received to a QLLC station list entry. Either one or no matches are found, since LCNs must be unique within an interface. If there is no match or the entry has no DMAC/DSAP defined, the connect-in fails. Otherwise a connection is initiated to the defined DMAC/DSAP. The origin MAC/SAP for the connection is the SMAC/SSAP from the same list entry.

For **SVCs**, DLSw derives MAC/SAP addresses using either the X.25 calling party address, or a *connection id* (bytes 4-11) in the call user data field of the received Call_Request packet. If the calling party address is available, DLSw first checks it against all its configured SVC DTE addresses for the called interface. Either one or no matches are found, since DTE addresses must be unique within an interface. If a match is found and the QLLC station list entry has a non-zero DMAC/DSAP, DLSw uses this DMAC/DSAP as the target address for connection establishment. The origin MAC/SAP for the connection is the SMAC/SSAP from the same list entry.

If no calling party address is available, or there is one but it matches an entry with no defined DMAC/DSAP, or it does not match any defined DTE address for the called interface, DLSw checks whether any connection id (CID) received in the Call_Request packet matches any defined in DLSw QLLC Destination records. The CID is interpreted as an EBCDIC alphanumeric string of up to 8 characters.

If there is a CID match, DLSw uses the associated DMAC/DSAP in the Destination Record as the destination address for circuit establishment. If there was also a calling party address match (with no defined DMAC/DSAP), DLSw uses the SMAC/SSAP from the matched station list entry. Otherwise, DLSw dynamically assigns the SMAC and SSAP. For the SMAC, DLSw chooses the next available (round robin) MAC address in the range defined by the global DLSw configuration parameters *QLLC base MAC address* and *Max dynamic addresses*. The dynamically-selected SSAP is always 0x04.

If there is no calling party address or connection id match, DLSw does not take the call. Note that CIDs are the only way a single calling party address can place calls to multiple destinations.

APPN and DLSw may both accept QLLC calls from the same calling party address. DLSw gets first access to the call since it is more restrictive in what calls it will accept. If DLSw finds no calling party or connection id match, DLSw does not clear the call, but allows it to be presented to APPN.

For an incoming call to be accepted, then, either the calling party address or a connection id must be defined to DLSw. While this is required primarily to provide address mapping, it also provides an element of security against incoming calls from unauthorized parties. Other possible security measures include not enabling an interface for incoming calls to DLSw, and setting the number of possible dynamic source MAC addresses to zero. The former will prevent all incoming calls on that interface, even from DTE addresses configured in DLSw. The latter will prevent only dynamic calls in from non-configured DTE addresses.

DLSw Configuration and X.25 Configuration

To use DLSw's QLLC support over a given X.25 interface, you must configure the address mapping as part of DLSw configuration, and you must also configure the following information as part of X.25 interface configuration. See “Configuring X.25 Interfaces” on page 27-22 for an example of these steps, and refer to *Software User’s Guide for Nways Multiprotocol Access Services Version 1 Release 1* for additional information.

1. Configure the interface to be X.25, and configure its base X.25 interface parameters.
2. Add DLS as a protocol to be supported.
3. Configure the PVCs that DLSw is to use, and associate them with DLSw.
4. Configure static SVC DTE addresses that DLSw is to use, and associate them with DLSw. These are the same addresses configured in DLSw. It is not necessary to configure the DTE addresses of QLLC end stations that may call in dynamically.

Unlike SDLC, X.25 has no capability to dynamically create a link station (virtual circuit) definition based on information configured in DLSw.

Relationship to the XTP Function

The X.25 Transport Protocol (XTP) is a router function that takes packets from X.25 virtual circuits, and transports them via TCP/IP to another router that also supports XTP. The destination router then removes XTP header information and delivers the packets onto a destination X.25 virtual circuit.

This function compares with DLSw QLLC support in the following ways:

- Both functions use TCP/IP to communicate between peer routers, and can multiplex the information from multiple virtual circuits (or DLSw sessions) onto a single TCP connection.
- With both functions, the router terminates the layer-2 LAPB and layer-3 packet layer connections to the X.25 end station. LAPB control frames do not flow across TCP/IP.
- XTP supports communication only between two X.25 end stations. DLSw performs protocol conversion between LLC (remotely bridged or on a LAN), SDLC, QLLC, and any other data type supported by a DLSw product.
- XTP is not sensitive to the LLC type (for example, QLLC or PAD) operating above the packet layer. As long as both X.25 end stations support the same LLC type, they can communicate via XTP. DLSw QLLC support can communicate only with SNA end stations running QLLC.
- With XTP, there is a configured association among a virtual circuit on one X.25 network, a peer router, and a virtual circuit on another X.25 network. For SVCs only, it is possible to define multiple peer routers and attempt to bring up a connection through a secondary router should the primary router be unavailable, but XTP does not perform parallel searches or connection establishment attempts. DLSw, on the other hand, maps a virtual circuit to a MAC and SAP address, then conducts a fully dynamic search among multiple peers to locate the destination station. With DLSw multicast support, it is not even required to configure the individual peer IP addresses to be searched.
- XTP can map a PVC only to another PVC, and an SVC only to another SVC. In DLSw QLLC-to-QLLC configurations, it is possible to map a PVC to an SVC.

In practice this may be of limited value, because DLSw will attempt to bring up the SVC whenever the QLLC protocol is active on the PVC.

- With XTP using SVCs, calls are placed to and from the DTE addresses of the X.25 end stations. An X.25 switch or network subscription may need to be configured to allow the router to represent multiple DTE addresses. With DLSw, calls are placed from end stations to the DTE address of the router interface, and vice versa.
- DLSw is a standard developed by the APPN Implementers Workshop and documented in an IETF RFC. As such, it is supported by a number of vendors. XTP is currently supported only in certain IBM and compatible router products.

You must use DLSw when:

- You require protocol conversion from QLLC to SDLC or LLC
- You need multiple concurrent paths to a destination

You must use XTP when:

- You are running a non-QLLC protocol over X.25

In other QLLC-to-QLLC configurations, choose the protocol that best matches the requirements of your network. For more information on XTP, see Chapter 22, “Using, Configuring, and Monitoring XTP” on page 22-1.

APPN Interface Support

DLSw has an internal interface with APPN that connects APPN to end stations attached to remote DLSw routers. The remote routers need not support APPN, which may reduce the amount of memory they require. As shown in Figure 27-5, this internal interface is the equivalent of collapsing a DLC connection (for example, LLC over a LAN) into a single software interface.

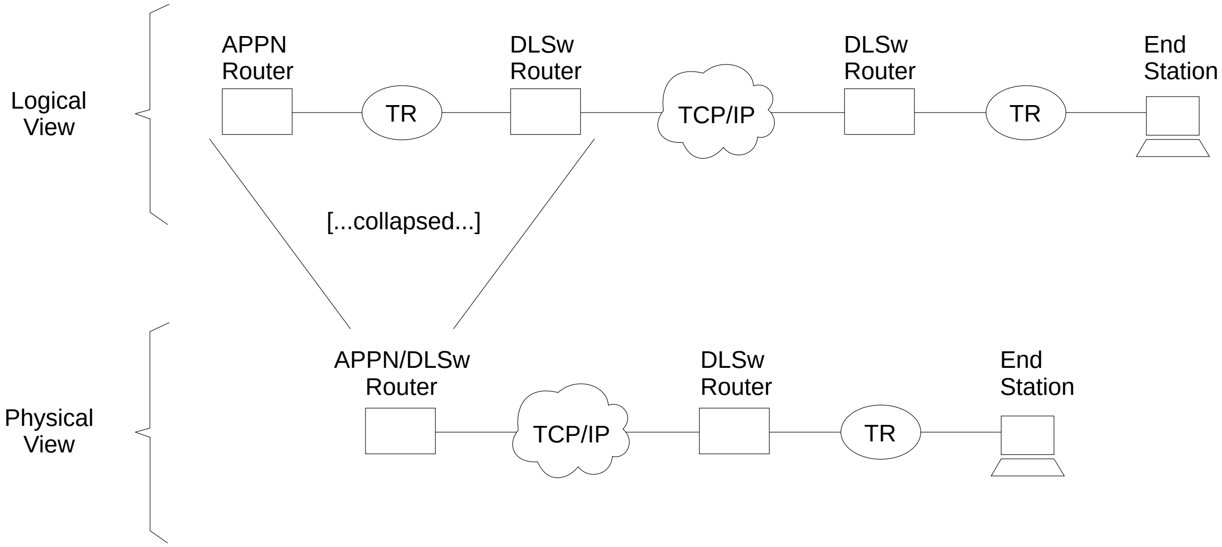


Figure 27-5. APPN-to-DLSw Software Interface

APPN cannot use the DLSw software interface to reach end stations that are locally attached to the APPN/DLSw router. It must use its native DLC support to communicate with these devices.

No additional DLSw configuration is required to support the APPN interface. You should enable TCP keepalive messages to the DLSw remote router, to allow detection of the loss of the link stations on the DLSw port. You must configure APPN to use a DLSw virtual interface to reach a given end station. For information on implementing APPN using DLSw, refer to the chapter on configuring APPN in *Protocol Configuration and Monitoring Reference Volume 2 for Nways Multiprotocol Access Services Version 1 Release 1*.

Using the Neighbor Priority Feature

Many DLSw network configurations provide multiple paths from an origin DLSw router to destination end-stations by making the end-stations local to more than one destination DLSw router. To provide additional control over which remote DLSw router are used for new circuits, you can assign a priority (high, medium, or low) to each defined neighbor. Although the allowable values are similar, neighbor priority is ***not*** the same as priorities for balancing SNA and NetBIOS traffic that is discussed in “Balancing SNA and NetBIOS Traffic” on page 27-17.

For neighbor priority, you assign a priority when you define a neighbor using the **add tcp** or **join group** commands. A group’s priority is inherited by all transport connections brought up within that group.

When DLSw is originating a circuit and finds that the destination MAC address or NetBIOS name is reachable through multiple remote DLSw routers, it establishes the circuit through the one of those neighbors that has the highest priority. If there are multiple remote routers that share this highest priority, DLSw uses a “round-robin” method of allocating new circuits among those routers.

Using neighbor priority, you can establish a primary/backup relationship among remote routers. A lower priority router is not used unless the higher priority router becomes unavailable. In addition, the round-robin method provides for load balancing among routers of equal priority.

Notes:

1. When an SNA frame is received that is destined for a MAC address that does not have cached information for which neighbors can reach the MAC address, an SNA explorer message is sent to all DLSw neighbors. Responses for the SNA explorer message are collected for the period of time specified by the “neighbor priority wait timer.” After this period of time, the MAC address cache entry is updated with information from the responses from the neighbors with the highest priority. One of these neighbors is chosen to handle this SNA circuit, and a response is sent to the original SNA frame that was received. Subsequent SNA circuit requests for this MAC address will use one of the cached highest priority neighbors to bring up the circuit.
2. When a NetBIOS frame is received that is destined for a NetBIOS name that does not have a current cache information entry for that NetBIOS name, a NetBIOS explorer message is sent to all DLSw neighbors supporting NetBIOS. Unlike the SNA case, responses are collected for a specified period of time before the response to the original NetBIOS frame is sent. The end station timers do not usually allow a wait delay at the router.

   Thus, the first response to the NetBIOS explorer message is saved. This neighbor is used to bring up this NetBIOS circuit, and a response is sent to the original NetBIOS frame that was received. In the meantime, subsequent responses to the NetBIOS explorer message are used to update the NetBIOS name cache.

- If a response from a neighbor of equal priority to the currently cached information is received, it is added to the cache.
- If a response from a neighbor of higher priority to the currently cached information is received, the currently cached information is removed and the information for the new higher priority neighbor is added.
- If a response from a neighbor of lower priority to the currently cached information is received, it is ignored. Subsequent NetBIOS circuit requests for the NetBIOS name will use one of the currently cached highest priority neighbors to bring up the circuit.

Balancing SNA and NetBIOS Traffic

With the introduction of DLSw support for NetBIOS traffic, you need to control the mix of SNA and NetBIOS traffic within DLSw transport connections. Without this control, NetBIOS file transfers have a tendency to shut out interactive SNA traffic for undesirably long periods of time, especially if the TCP connections are running over relatively slow WAN links. You can control this traffic mix using configuration parameters of the **set priority** command. Using these parameters, you can:

- Establish a rough ratio of the number of frames from each protocol transmitted onto a TCP connection during periods of congestion
- Establish a maximum frame size for NetBIOS frames so that one large frame will not consume a slow WAN link.

To set up a rough ratio of SNA and NetBIOS frames, you globally select one of four priority values (critical, high, medium, or low) for each protocol. At circuit setup time, the router uses the DLSw Version 1 (RFC 1795) circuit priority mechanism to try to negotiate each new circuit’s priority to the value for the protocol the circuit will be carrying. The neighbor router may be trying to negotiate to a different circuit priority based on some other set of criteria, so circuits originated remotely may end up with a priority other than the two values you selected for SNA and NetBIOS. In any event, each established circuit is assigned one of the four priorities by the router that initiated that circuit’s establishment.

During periods of TCP congestion, the router queues frames (from circuits that have data to transmit) into one of four queues - one queue for each possible circuit priority. The frames are queued FIFO within each priority. To feed the TCP transmit process, the router selects frames from each priority queue as dictated by the “message allocation by priority” parameter. This defaults to 4/3/2/1, meaning that at most, four messages are taken from the critical priority queue, followed by at most three from the medium priority queue, and so on. If a queue is empty, it misses its turn in the cycle.

To prevent a single large NetBIOS frame from dominating a slow link for a long time, you can use the “NetBIOS maximum frame size” parameter to provide an upper limit to the size of a single NetBIOS frame. This value is passed to both NetBIOS end-stations during circuit establishment using the Largest Frame (LF) bits in the source-routing MAC header. Source-routing NetBIOS end-stations should observe the LF values and not generate frames larger than the specified value.

Setting up DLSw

The following sections explain the setup procedures for DLSw:

- “DLSw Configuration Requirements”
- “Setting Global Buffers”
- “Configuring OSPF for DLSw”
- “Configuring Adaptive Source Route Bridging (ASRT) for DLSw” on page 27-19
- “Configuring the Internet Protocol (IP) for DLSw” on page 27-20
- “Configuring SDLC Interfaces” on page 27-21
- “Configuring X.25 Interfaces” on page 27-22
- “Configuring DLSw” on page 27-22

In addition, a sample DLSw configuration with explanatory notes has been included (see “Sample DLSw Configuration” on page 27-23).

DLSw Configuration Requirements

To use DLSw, configure the following protocols: ASRT, IP, and DLSw. In addition, you may need to configure the protocols listed in Table 27-1.

Table 27-1. DLSw Optional Protocols

Optional Protocol	When Used
LLC2	When non-default LLC2 parameters need to be used
SDLC	To connect to devices using SDLC
OSPF	To use DLSw multicast groups
X.25	To connect to devices using QLLC

The sections that follow explain how to configure these required and optional protocols in a step-by-step fashion.

Setting Global Buffers

When running DLSw in a 4M DRAM 2216, it may be necessary to allow more memory for DLSw by reducing the number of global packet buffers. Enter the **set global** command at the `Config>` prompt, then enter the number of global packet buffers (the recommended number for a 4M DRAM 2210 is 50).

Configuring OSPF for DLSw

If you want to use the DLSw group feature, you need to configure OSPF as follows:

1. *Enter the OSPF Configuration process*. Use the **protocol ospf** command from the `Config>` prompt.
2. *Assign the OSPF address to the hardware interface*. Use the **set interface** command to assign the OSPF address to the hardware interface you are using to connect to the other DLSw peer.
3. *Enable Dynamic Routing*. Use the **enable rip** or **enable ospf** command to enable routing. If you are using DLSw group function, you must enable the OSPF routing protocol and OSPF multicast routing from the OSPF `Config>`

prompt. All defaults for OSPF work fine. You only need to enable OSPF and multicast OSPF after using the **join-group** command rather than using add TCP neighbor to explicitly define the TCP connection.

Configuring Adaptive Source Route Bridging (ASRT) for DLSw

Since the DLSw router appears as a bridge to attached end-stations, you need to configure source route bridging. Do this by following these steps:

1. Enter the ASRT (Adaptive Source Route Bridging) configuration process. Use the **protocol asrt** command from the `Config>` prompt.

2. Enable bridging to occur on the router using the **enable bridge** command. Each bridge must have an unique bridge address in each DLSw.

3. Add a bridge port with the **add port** command. The display prompts you for an interface number and a port number.

   - ***For token-ring interfaces:***

     Running DLSw over token ring requires that only source route bridging be present on the designated bridge port. Thus, you must disable transparent bridging. Do this with the **disable transparent** command. Then, issue the **enable source routing** command to turn on source routing for the bridge port.

   - ***For Ethernet interfaces:***

     Ensure that the transparent bridging is enabled on the bridge port. Issue the **enable transparent** command.

4. If you are configuring the router for ***concurrent bridging and DLSw:***

   Create a protocol filter against the SAPs (service access points) you intend DLSw to use. If the router is performing bridging operations, plus forwarding packets via DLSw, it is essential to do this. If you do not, DLSw packets that are received by the bridge will be forwarded by DLSw and bridged by the router. The idea is to prevent DLSw packets from being forwarded (bridged) in parallel with DLSw routing.

   To create a SAP filter, issue this command at the `Config ASRT>` prompt:

   ```
   add protocol-filter dsap 4
   ```

   In addition to this command, you must specify the bridge port to which it applies. This command tells the router to filter all traffic that has a DSAP of 4 except on the port designated for DLSw. (Note that this assumes you have chosen a SAP of 4 for DLSw traffic. This is something you do during the DLSw configuration.)

5. Enable DLSw using the **enable dls** command. This enables the DLSw protocol on the bridge port you have designated.

6. Verify the ASRT configuration. You do not have to do this, but it is a good idea to check the bridge configuration before proceeding. Use the **list bridge** command to verify the configuration of the ASRT protocol. The following example shows the results of the list bridge command after configuring ASRT.

```
                 Source Routing Transparent Bridge Configuration
                 ===============================================

 Bridge:                  Enabled                 Bridge Behavior: Unknown
                    +----------------------------+
 -------------------| SOURCE ROUTING INFORMATION |-------------------------------
                    +----------------------------+
 Bridge Number:          01                       Segments: 1
 Max ARE Hop Cnt:        14                      Max STE Hop cnt:  14
 1:N SRB:                 Not Active              Internal Segment:  0x000
 LF-bit interpret:        Extended

                    +-------------------+
 -------------------| SR-TB INFORMATION |----------------------------------------
                    +-------------------+
SR-TB Conversion:        Disabled
TB-Virtual Segment:      0x000                   MTU of TB-Domain:  0

                   +-----------------------------------+
-------------------| SPANNING TREE PROTOCOL INFORMATION |-------------------------
                   +-----------------------------------+
Bridge Address:          Default                 Bridge Priority:   32768/0x8000
STP Participation:       IEEE802.1d

                   +-------------------------+
-------------------| TRANSLATION INFORMATION |-----------------------------------
                   +-------------------------+
 FA<=>GA Conversion:      Enabled                 UB-Encapsulation :  Disabled
 DLS for the bridge:      Enabled

                    +------------------+
 -------------------| PORT INFORMATION |------------------------------------------
                    +------------------+
 Number of ports added: 1
 Port:   1      Interface:       0      Behavior:    SRB Only STP:  Enabled
```

Configuring the Internet Protocol (IP) for DLSw

You need to configure IP so that the local DLSw router can form TCP connections to other DLSw peers. To do this:

1. Enter the IP configuration process by issuing the **protocol ip** command from the `Config>` prompt.
2. Assign the IP address to the hardware interface. Use the **add address** command to assign the IP address to the hardware interface you are using to connect to the other DLSw peer.
3. ***Enable Dynamic Routing***. You must choose either OSPF or RIP as your routing protocol. Using OSPF is recommended because it requires less network overhead than RIP.
    - To enable OSPF: enter the **protocol ospf** command from the `Config>` prompt. This brings you to the `OSPF Config>` prompt. If you want to use DLSw Group function, you must enable Multicast-OSPF.
    - To enable RIP: enter **enable RIP** at the `IP Config>` prompt.
4. Set the Internal IP Address. Use the **set internal-ip-address** command to set the address that belongs to the router as a whole, and not to any particular interface. The internal ip address is used by the router when making the TCP connection to the other DLSw peer.
    - If you are using RIP, choose one of the interface addresses as your internal-ip-address.

- If you are using OSPF, choose an address that has a different subnet from any subnets that are being used in your network.

Configuring SDLC Interfaces

The SDLC configuration command allows you to create or modify the SDLC interface configuration as part of the DLSw configuration process.

You must configure SDLC links if you intend to support SDLC over DLSw. This section explains how to access the SDLC configuration console, and describes SDLC-related commands.

If there is an SDLC device directly connected, configure the SDLC protocol as follows:

1. Set the data link to SDLC: At the `Config>` prompt, use the **set data-link SDLC** command to configure the data-link type for the serial interface. You will be prompted for an interface number.
2. Enter the SDLC configuration process: Use the **network** command at the `Config>` prompt to enter the SDLC configuration process. You will be prompted for an interface number.
3. When you configure DLSw, you add SDLC stations and the software assigns the following defaults to the stations:
   - Maximum BTU is maximum allowable by the interface
   - Tx and Rx Windows are 7 for Mod 8, and 127 for Mod 128

   If the defaults are satisfactory, you do not need to add SDLC stations in the SDLC configuration. If they are unsatisfactory, use the **add station** command to change the defaults.
4. The link role defaults to primary. If necessary, change the link role to secondary or negotiable using the **set link role** command.
5. You can set up group polling for secondary stations on the link. To do so, set the group poll address using the **set link group-pool** command, and use the **add station** and **set station group-inclusion** commands to include stations in the group poll list.
6. Set the link clocking source (Optional): If you want to connect directly to an SDLC device without using a modem eliminator, use a DTE cable and the command **set link clocking internal**.
7. Set the link speed (Optional): If you are using internal clocking, use the **set link speed** command to choose the clock speed for this line.

   Note: If you are using SDLC to connect from a PC, you must also set the encoding (NRZ/NRZI), and duplex (full/half) to match the PC's configuration.
8. Set the link cable to RS-232, X.21, V.35, or V.36.
9. Verify the SDLC configuration: Use the **list link** command to verify the SDLC interface configuration.

Configuring X.25 Interfaces

Configure the X.25 interface if you intend to use DLSw's support for QLLC devices. Follow these steps:

1. Set the interface to be X.25. At the `Config>` prompt, use the **set data-link X25** command to set the type of the serial interface. You will be prompted for an interface number.
2. Enter the X.25 configuration process, using the **net** command at the `Config>` prompt. You will be prompted for an interface number, and thereafter you will enter commands at the `X.25 Config>` prompt.
3. Use the **set address** command to define the router's DTE address on this interface.
4. Use the **set pvc** and **set svc** commands to define the range of logical channel numbers to be used for PVCs and available for use by SVCs. Any PVCs you define in DLSw configuration must have channel numbers within the PVC range you define here. For SVCs, you should make sure that the number of channels available for incoming and outgoing calls is sufficient for the number of simultaneous calls you expect DLSw to be able to place or answer.
5. Use the **add protocol** command to add "dls" as a protocol to operate above X.25 on this interface. X.25 understands that this implies QLLC support, and prompts for a series of QLLC operational parameters whose value will apply to all DLSw virtual circuits on this interface.
6. Use the **add pvc** command to associate a given PVC logical channel number with the DLSw protocol. You should do this for every PVC on this interface that DLSw is configured to use (i.e., every PVC for which you do an **add qllc station** command in DLSw configuration). The logical channel number is the key that will match the DLSw configuration for this station, with this X.25 PVC definition.
7. Use the **add address** command to create a list of X.25 DTE addresses for all PVCs and SVCs that are defined in DLSw configuration. Note that DLSw does not use DTE addresses for PVCs, but they are required within X.25 configuration. It is not necessary to add the DTE addresses of QLLC end stations that may dynamically call in to DLSw and are not configured in DLSw.
8. Set any physical layer or national personality characteristics required for attachment to the X.25 network. For a description of X.25 configurable parameters, refer to the chapter on configuring the X.25 network interface in *Software User’s Guide*.

Configuring DLSw

Before configuring DLSw, enter the **list device** command at the `Config>` prompt to list the interface numbers of different devices.

To configure the DLSw protocol:

1. At the `Config>` prompt, enter the **protocol dls** command. This brings up the `DLSw config>` prompt.
2. Enter the **enable dls** command to enable DLSw in the router.
3. Enter the **set srb** command to designate the SRB (source route bridging) segment number for the DLS router.

This SRB segment number must be the same for all DLSw routers attached to the same LAN, and should be unique in the source route bridge domain. The bridge uses this number in the Routing Information Field (RIF) when the frames are sent on the LAN. The segment number is the key for preventing loops.

4. Enter the **open-sap** command for each SAP that you want DLSw to switch. The router prompts for interface numbers. To open commonly-used SNA SAPs (4, 8, and C), specify SNA. Minimally, open SAPs 0 and 4. To open the NetBIOS SAP, specify NB or F0. To open the LNM SAPs, specify LNM or, minimally, 0 and F4.

5. Use the **add tcp** command to add the IP address of each DLSw peer that you want to configure. If you want the router to accept connections from non-configured peers, use the **enable-dynamic neighbor** command. TCP connections also can be established using multicast OSPF and the **join-group** command.

   Note: A router can participate in a group ***only*** if its peer router is an MAS-based platform running DLSw. If you configure one DLSw router for a group, you must enable OSPF and MOSPF on all DLSw routers in the group.

6. For your DLSw configuration to support SDLC, you must add an SDLC link station using the **add sdlc** command.

7. For your DLSw configuration to support QLLC, add a QLLC link station with the **add qllc station** command.

   Or, if you want to support dynamic SVCs, enable X.25 interfaces for call-in with the **enable qllc callin** command and define DLSw destinations with the **add qllc destination** command.

Sample DLSw Configuration

The following sample DLSw configuration assumes that the router has not been configured for any other protocols or data-links. For this reason, the script begins at the `Config (only)>` prompt, rather than at `Config>`.

Sample Diagram

The example is based on the information shown in Figure 27-6 on page 27-24.

The DLSw router being configured (R1 in the diagram) supports one LLC and one SDLC connection to its DLSw peer (R2). The TCP connection between the two routers is over serial line.

Configuring R1 for DLSw requires all of the information shown. This information includes the :

- Internal IP Address of R1 and R2
- IP address of each port used to maintain the TCP connection between the routers
- Interface numbers assigned to the token-ring and SDLC devices, and that used for the TCP connection
- MAC address of the attached SDLC device
- MAC address of the attached QLLC device

- Source route bridge segment number of the attached token-ring device

The example indicates where this information is provided in the course of the configuration procedure.

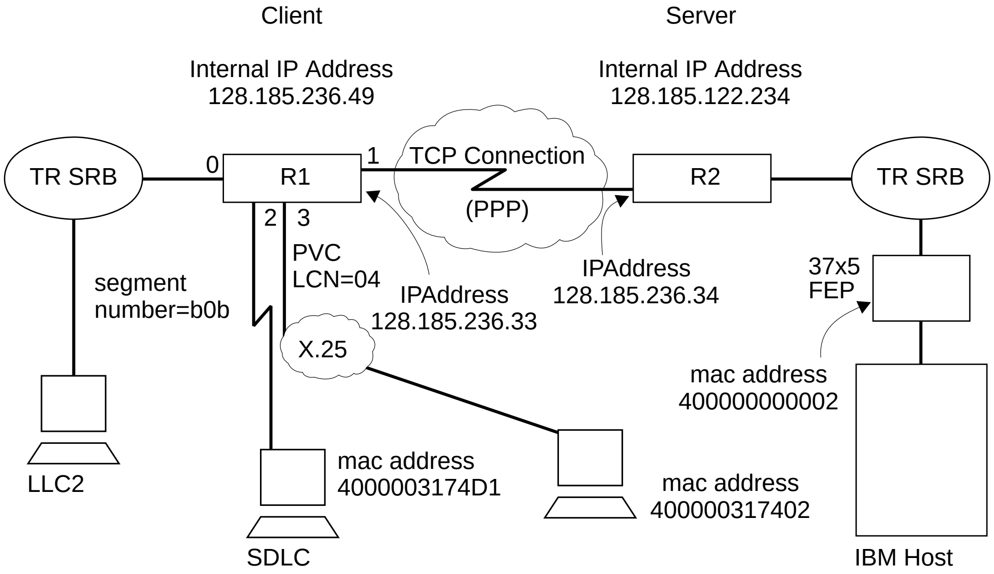


Figure 27-6. Sample Diagram for DLSw Configuration

Sample Configuration Commands

This section provides examples of the following:

- "Step 1: Adding Devices"
- "Step 2: Configuring Protocols" on page 27-28
- "Step 3: Implementing Protocol Filtering" on page 27-32
- "Step 4: Configuring DLSw" on page 27-33

Step 1: Adding Devices

The devices you will add are token ring, SDLC, or QLLC. You may also add Ethernet as a transparent bridge port. For purposes of illustration, this sample DLSw configuration supports SDLC, LLC, and QLLC. However, it is only necessary for an actual configuration to support one of these data-links.

In the case of SDLC and QLLC, you must explicitly set the data link, because the interface also supports other data links such as FR, X.25, and SDLC Relay.

```
Config (only)>set data-link sdlc 2
Config (only)>set data-link x25 3
```

After adding devices, you can list the devices to verify that they have been assigned to the appropriate router interfaces.

If you are configuring IBM 2216 Model 400 Switch, enter the list device command at the `config>` prompt to display a list of the configured devices and their interface numbers.

Notice that this **list** command shows that a token-ring device has been assigned to interface 0, while an SDLC device is assigned to interface 2.

1. ***Add a token-ring device***:

   Configure the token-ring setup. 16 Mbps is usually used with UTP cables, so this is done here. The **list** command shown in these procedures is not required either at this point, or at any other time during configuration of the router.

   ```
   Config (only)> network 0
   Token-Ring interface configuration

   TKR config>speed 16
   TKR config>media utp

   TKR config>list

   Token-Ring configuration:
   Packet size (INFO field): 2052
   Speed:                    16 Mb/sec
   Media:                    Unshielded
   RIF Aging Timer:          120
   Source Routing:           Enabled
   MAC Address:              000000000000
   IPX interface configuration record missing

   TKR config>exit
   ```

   Configuring the WAN Interface. The first port (interface 1) is used for the WAN (TCP/IP) link. The data link selected for the WAN is PPP. This is the default choice for the data link. The other possibilities are frame-relay and X.25.

   ```
   Config (only)>network 1
   Point-to-Point user configuration
   PPP Config>list hdlc
   Mode: Synchronous
   Encoding: NRZ
   Idle State: Flag
   Clocking: External
   Cable type: RS-232 DTE
   Speed (bps): 0

   Transmit Delay Counter: 0
   Lower DTR: Disabled
   ```

   You must also set the cable type. For PPP the cable type is set using the **set hdlc cable** command.

   Next, set the line speed and clocking type for the serial interface, if necessary.

   ```
   PPP Config>set hdlc clock internal
   Must also the line speed to a valid value
   Line speed (2400 to 2048000) [0]? 56000
   ```

   After setting the line speed and clocking type, you can check the configuration with the **list hdlc** command as shown

   ```
   PPP Config>list hdlc
   Mode: synchronous
   Encoding: NRZ
   Idle State: Flag
   Clocking: Internal
   Cable type: RS-232 DTE
   Speed (bps): 56000

   Transmit Delay Counter: 0
   Lower DTR: Disabled

   PPP Config>exit
   ```

2. Add an SDLC device

If you are configuring DLSw to support SDLC, the next step is to configure SDLC. Most of the configurable items do not need modification.

To access the SDLC configuration, use the **network** command and the number of the interface to which an SDLC device has been assigned (in this case, 2).

```
Config>network 2
SDLC user configuration
```

Most of the information that you add when you configure SDLC is hardware-related.

The example begins with a **list link** command. The **list** command does not alter the configuration, but shows you the values that are currently associated with the SDLC link.

If you are configuring a IBM 2216 Model 400 Switch:

```
SDLC 2 Config>list link
Link configuration for: LINK_2   (ENABLED)

Role:         PRIMARY          Type:       POINT-TO-POINT
Modulo        8                Frame Size   2048

Timers:    XID/TEST response:  2.0 sec
           SNRM response:      2.0 sec
           Poll response:      0.5 sec
           Inter-poll delay:   0.2 sec
Counters:  XID/TEST retry:  4
           SNRM retry:      6
           Poll retry:      10
```

In the same way that we configured a token-ring device, the clocking type and line speed must be modified for the SDLC device. If you are using an external modem eliminator, this is unnecessary.

```
SDLC 2 Config>set link clock internal
Must also set the line speed to a valid value
Line speed (2400 to 2048000) [0]? 9600
SDLC 2 Config>exit
```

3. Add a QLLC device

In order to support the QLLC station shown in Figure 27-6 on page 27-24, you must configure interface 3 to be X.25 and have QLLC support for DLSw on the indicated PVC. The following sample configuration shows QLLC support for DLSw on a PVC. You should:

a. Use the list device command to get a list of the configured interfaces.

b. Select the serial interface you want to configure X.25 on.

c. Record that interface number and use it on the set data-link command to configure X.25 on the interface.

In the example, X.25 is configured on interface 1.

```
Config>net
Network number [0]? 1
X.25 User Configuration

X.25 Config>li sum

X.25 Configuration Summary

Node Address:       <none>
Max Calls Out:           4
Inter-Frame Delay:       0    Encoding:  NRZ
```

```
Speed:          56000         Clocking:  Internal
MTU:             2048         Cable:     RS-232 DTE
Lower DTR:   Disabled
Default Window:     2         SVC idle:  30 seconds
National Personality: GTE Telenet (DTE)
PVC             low: 0    high: 0
Inbound         low: 0    high: 0
Two-Way         low: 1    high: 64
Outbound        low: 0    high: 0
Throughput Class in bps Inbound:  2400
Throughput Class in bps Outbound: 2400

X.25 Config>set addr
address [ ]? 3721111
X.25 Config>set pvc low 1
X.25 Config>set pvc high 4
X.25 Config>set svc low-two 5
X.25 Config>set svc high-two 64
X.25 Config>li sum

X.25 Configuration Summary

Node Address:       3721111
Max Calls Out:           4
Inter-Frame Delay:       0    Encoding:  NRZ
Speed:          56000         Clocking:  Internal
MTU:             2048         Cable:     RS-232 DTE
Lower DTR:   Disabled
Default Window:     2         SVC idle:  30 seconds
National Personality: GTE Telenet (DTE)
PVC             low: 1    high: 4
Inbound         low: 0    high: 0
Two-Way         low: 5    high: 64
Outbound        low: 0    high: 0
Throughput Class in bps Inbound:  2400
Throughput Class in bps Outbound: 2400

X.25 Config>li prot

X.25 protocol configuration

No protocols defined
X.25 Config>add prot
Protocol [IP]? dls
Idle timer [20]?
QLLC response timer [20]?
QLLC response count [10]?
Accept Reverse Charges [N]?
Request Reverse Charges [N]?
Station Type (1) PRI  (2) SEC  (3) (PEER) [3]?
Non standard packet size [32]?
Packet window size [128]?
Max message size [256]?
Call User Data (in HEX) [0000000000000000]?

X.25 Config> li prot

X.25 protocol configuration

Prot         Window      Packet-size        Idle     Max     Station
Number       Size      Default Maximum      Time     VCs      Type
26 -> DLS     128        32      256         20       4       PEER

X.25 Config> li pvc

X.25 PVC configuration

No PVCs defined
X.25 Config>add pvc
Protocol [IP]? dls
Packet Channel [1]? 4
Destination X.25 Address [ ]? 4444
```

```
Window Size [2]?
Packet Size [128]?

X.25 Config> li pvc

X.25 PVC configuration

Prtcl        X.25_address     Window    Pkt_len   Pkt_chan
26 -> DLS    4444             2         128       4

X.25 Config> li add

X.25 address translation configuration

No address translations defined

X.25 Config> add addr
Protocol [IP]? dls
Enter an DLS address identifier (upto 12 chars) [ ]? Chicago
X.25 Address [ ]? 4444
X.25 Config> li addr

X.25 address translation configuration

IF #    Prot #       Protocol address -> X.25 address
1       26 -> DLS    Chicago          -> 4444
```

Note: The DTE address “4444” used for the PVC with logical channel number “4” is not used by DLSw, but is used only by X.25 for correlating configuration information. Likewise, the DLSw protocol address (“Chicago” in this example), has no meaning to DLSw but is solely for ease of reference to the various DTE addresses that DLSw can use. Unlike other protocols running on X.25, DLSw address translation is defined as part of DLSw configuration and not in X.25 configuration.

Step 2: Configuring Protocols

Once device configuration is complete, you must configure the necessary protocols. To run over DLSw you must configure IP, OSPF (or RIP), ASRT, and the DLSw protocol.

1. *Configure IP*

This example begins with the IP configuration:

```
Config>protocol ip
Internet protocol user configuration
```

The **list all** command shows the default IP configuration.

```
IP config>list all
Interface addresses
IP addresses for each interface:
   intf  0   192.1.1.3        255.255.255.0    Local wire broadcast, fill 1
   intf  1                                     IP disabled on this interface
   intf  2                                     IP disabled on this interface

Routing

Protocols
BOOTP forwarding: disabled
IP Time-to-live: 64
Source Routing: enabled
Echo Reply: enabled
Directed broadcasts: enabled
ARP subnet routing: disabled
ARP network routing: disabled
Per-packet-multipath: disabled
OSPF: enabled
BGP: disabled
```

```
RIP: enabled
RIP default origination: disabled
  Per-interface address flags:
     intf  0   192.1.1.3        Send net, subnet, static and default routes
                                Received RIP packets are ignored.
     intf  1                    IP & RIP are disabled on this interface
     intf  2                    IP & RIP are disabled on this interface

Accept RIP updates always for:
[NONE]
```

This example shows the creation of a minimal IP configuration. For more information on this important protocol, see Chapter 16, “Using and Configuring IP” on page 16-1.

- The first thing to do is to add an internet address and assign it to an interface over which you intend to run IP traffic:

  ```
  IP config>add address
  Which net is this address for [0]? 1
  New address [0.0.0.0]? 128.185.236.33
  Address mask [255.255.0.0]? 255.255.255.0
  ```

- Set the internal IP Address. This is the address that remote DLSw routers use to connect to the router you are configuring. If RIP is the routing protocol selected for IP, the internal IP address must match the IP address configured for an interface.

  ```
  IP config>set internal-ip-address 128.185.236.49
  ```

- Subsequent use of the **list** command displays the newly added information.

```
IP config>list all
Interface addresses
IP addresses for each interface:
   intf  0   192.1.1.3        255.255.255.0    Local wire broadcast, fill 1

   intf  1   128.185.236.33   255.255.0.0      Local wire broadcast, fill 1

   intf  2                                     IP disabled on this interface
Internal IP address: 128.185.236.49

Routing

Protocols
BOOTP forwarding: disabled
IP Time-to-live: 64
Source Routing: enabled
Echo Reply: enabled
Directed broadcasts: enabled
ARP subnet routing: disabled
ARP network routing: disabled
Per-packet-multipath: disabled
OSPF: enabled
BGP: disabled
RIP: enabled
RIP default origination: disabled
  Per-interface address flags:
     intf  0   192.1.1.3        Send net, subnet, static and default routes
                                Received RIP packets are ignored.
     intf  1   128.185.236.33   Send net, subnet, static and default routes
                                Received RIP packets are ignored.
     intf  2                    IP & RIP are disabled on this interface

Accept RIP updates always for:
[NONE]

     IP config>exit
```

2. Configure OSPF or RIP

In this configuration, OSPF is used rather than RIP. You can use either of these routing protocols. However, if you choose RIP, you will not be able to use DLSw Group function.

First, enter a **list** command. The command displays the default OSPF configuration. You must modify this configuration to run DLSw.

```
Config>protocol ospf
Open SPF-Based Routing Protocol configuration console
OSPF Config>list all

                  --Global configuration--
          OSPF Protocol:          Enabled
          # AS ext. routes:       1000
          Estimated # routers:    50
          External comparison:    Type 2
          AS boundary capability: Disabled
          Multicast forwarding:   Disabled

                  --Area configuration--
Area ID       AuType     Stub?   Default-cost Import-summaries?
0.0.0.0       0=None     No      N/A           N/A
```

- Now, enable OSPF and estimate the number of external routes and OSPF routers.

```
OSPF Config>enable ospf
Estimated # external routes [0]? 100
Estimated # OSPF routers [0]? 25
```

- Because this example implements DLSw Group Function, you must enable multicast OSPF, as shown:

```
OSPF Config>enable multicast
Inter-area multicasting enabled? [No]:
```

- Issue the **set interface** command for every physical IP interface that will use OSPF. This example assumes that the backbone is the OSPF area (0.0.0.0). At this point, only one IP interface has been defined.

```
OSPF Config>set interface 128.185.236.33
Attaches to area [0.0.0.0]?
Retransmission Interval (in seconds) [5]?
Transmission Delay (in seconds) [1]?
Router Priority [1]?
Hello Interval (in seconds) [10]?
Dead Router Interval (in seconds) [40]?
Type Of Service 0 cost [1]?
Authentication Key [ ]?
Retype Auth. Key [ ]?
Forward multicast datagrams? [Yes]:
Forward as data-link unicasts? [No]:
IGMP polling interval (in seconds) [60]?
IGMP timeout (in seconds) [180]?
OSPF Config>
```

- The following example shows the OSPF display after it has been configured. To see what has changed in the configuration, compare this display with the display of the default OSPF configuration shown earlier.

```
OSPF Config>list all

       --Global configuration--
OSPF Protocol:          Enabled
# AS ext. routes:       100
Estimated # routers:    25
External comparison:    Type 2
```

```
AS boundary capability: Disabled
Multicast forwarding:   Enabled
Inter-area multicast:   Disabled

                    --Area configuration--
Area ID      AuType       Stub? Default-cost Import-summaries?
0.0.0.0      0=None       No         N/A           N/A

                    --Interface configuration--
IP address       Area     Cost  Rtrns  TrnsDly  Pri  Hello  Dead
192.1.1.3        0.0.0.0     1      5       1     1     10    40
128.185.236.33   0.0.0.0     1      5       1     1     10    40

                            Multicast parameters
IP address    MCForward    DLUnicast    IGMPPoll    IGMPtimeout
192.1.1.3         On           Off          60           180
128.185.236.33    On           Off          60           180

OSPF Config>exit
```

3. Configure ASRT

Configure the router for source route bridging and enable the port as shown:

```
Config (only)>protocol asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>enable bridge
```

- The **list port** command shows that the port defaults to transparent bridging. Transparent bridging is what you want if your attached device is Ethernet, but it will not work if your device is token-ring. Note that port number 1 is port 1 on interface 0. In other words, port 1 is the logical bridge port for the physical interface set up for the token-ring (see Figure 27-6 on page 27-24).

```
ASRT config>list port
Port Id (dec)    : 128:01, (hex): 80-01
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface  : 0
Path Cost        : 0
+++++++++++++++++++++++++++++++++++++++++++
```

- To run over an LLC data link (such as token-ring), DLSw requires SRB (source route bridging). In this case, the first thing to do is disable transparent bridging on the port.

```
ASRT config>disable transparent
Port Number [1]?

ASRT config>enable source-routing
```

- Now, assign a segment number for the port. You only have to assign segment numbers when configuring a source route bridge device, such as token ring. In this example (see Figure 27-6 on page 27-24) **b0b** is the hexadecimal number assigned to the token-ring device.

```
Port Number [1]?
Segment Number for the port in hex(1 - FFF) [1]? b0b
Bridge number in hex (1 - 9, A - F) [1]?
```

  Next enable DLSw on the bridge port.

```
ASRT config>enable dls
```

After completing these steps, enable DLSw as shown. Listing the bridge configuration will confirm that you have configured ASRT correctly.

```
Segment Number for the port in hex(1 - FFF) [1]? b0b
Bridge number in hex (1 - 9, A - F) [1]?
ASRT config>enable dls
ASRT config>list bridge

          Source Routing Transparent Bridge Configuration
          ===============================================
Bridge:                 Enabled                 Bridge Behavior:
Unknown
                   +----------------------------+
-------------------| SOURCE ROUTING INFORMATION |------------------------------
                   +----------------------------+
Bridge Number:          01                      Segments:  1
Max ARE Hop Cnt:        14                      Max STE Hop cnt: 14
1;N SRB:           Not Active              Internal Segment: 0x000
LF-bit interpret:       Extended

                   +-------------------+
-------------------| SR-TB INFORMATION |---------------------------------------
                   +-------------------+
SR-TB Conversion:       Disabled
TB-Virtual Segment:  0x000                      MTU of TB-Domain:  0

                   +------------------------------------+
-------------------| SPANNING TREE PROTOCOL INFORMATION |----------------------
                   +------------------------------------+
Bridge Address:         Default                 Bridge Priority: 32768/0x8000
STP Participation:      IEEE802.1d

                   +-------------------------+
-------------------| TRANSLATION INFORMATION |---------------------------------
                   +-------------------------+
FA<=>GA Conversion:     Enabled                    UB-Encapsulation: Disabled
DLS for the bridge:     Enabled

                   +------------------+
-------------------| PORT INFORMATION |----------------------------------------
                   +------------------+
Number of ports added: 1
Port:   1      Interface:      0      Behavior:    SRB Only   STP: Enabled
```

Step 3: Implementing Protocol Filtering

This is an important step that is often neglected when configuring DLSw.

Because DLSw, rather than bridging, will be used to forward traffic on SAPs (service access points) 04, 08, 0C, we must add a special protocol filter to the bridging setup.

Note: You need to implement the filter described here only if bridging, in addition to DLSw, has been configured across the WAN links. This is not the case in this example. In this example, the procedure for creating a SAP filter is provided for reference purposes only.

The filter's purpose is to prevent the bridge from forwarding, on other ports, packets that should be handled only by DLSw. It is not optimal for DLSw and the bridging function to forward the same packets. When this occurs, race conditions develop that can cause degradation of network performance.

This command creates a filter that works on all packets with a destination SAP of 4. The **list** command issued subsequently displays the filter characteristics.

```
ASRT config>add prot-filter dsap 4
Filter packets arriving on all ports?? [No]: yes

ASRT config>list prot-f dsap
Protocol Class: DSAP
Protocol Type : 04
```

```
 Protocol State: FILTERED
 Port Map      : 1
 ==========================
 No ETHER type Filter Records Associated
 No SNAP Filter Records Associated
```

Once the filtering you need is in place, exit the ASRT configuration.

```
ASRT config>exit
```

Step 4: Configuring DLSw

The final step is configuring the DLSw protocol. The following **list** command shows the defaults.

```
Config>protocol dls
DLSw protocol user configuration

DLSw config>list dls
DLSw is                           DISABLED
LLC2 send Disconnect is           ENABLED
Dynamic Neighbors                 ENABLED

SRB Segment number                000
MAC <-> IP mapping cache size     128
Max DLSw sessions                 1000
DLSw global memory allotment      141056
LLC per-session memory allotment  8192
SDLC per-session memory allotment  4096
QLLC per-session memory allotment  4096
NetBIOS UI-frame memory allotment  40960

Dynamic Neighbor Transport Buffer Size 5120
Dynamic Neighbor Receive Buffer Size   5120
Dynamic Neighbor Maximum Segment Size  1024
Dynamic Neighbor Keep Alive            DISABLED
Dynamic Neighbor Priority              MEDIUM

QLLC base source MAC address           40514C430000
QLLC maximum dynamic addresses;        64
```

You enable DLSw, and set the SRB segment number. The segment number refers to the token-ring device, as shown in “Sample DLSw Configuration” on page 27-23.

```
DLSw config>enable dls
DLSw config>set srb 020
```

Configuring DLSw Groups and Static Sessions: This example defines both a group and a configured TCP session. Configuring DLSw does not require this. However, you must define one or the other (either a DLSw group or a configured TCP session) to connect-out to a neighbor DLSw router. If you want non-configured routers to connect-in, issue the **enable dynamic-neighbors** command.

The Join-Group Command: The **join-group** command is used to create a DLSw group. You designate each group member as Client/Server or Peer. Peer is the default.

Here, the **join-group** command is executed for R1 (see “Sample DLSw Configuration” on page 27-23), designating this DLSw router as a Client in group 1. To join this group, R2 would have to be added as a Server, and the **join-group** command issued on R2.

```
DLSw config>join
Configure group member (G) or specific multicast address (M) -  [G]?
Group ID (1-64 Decimal) [1]?
Client/Server or Peer Group Member(C/S/P)- [P]? c
```

```
Connectivity Setup Type (a/p) [p]?
Transmit Buffer Size (Decimal) [5120]?
Receive Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive (E/D) [D]?
Neighbor Priority (H/M/L) [M]?

DLSw config>list group

   Group# /                      Xmit     Rcv      Max
Mcast IP Addr    Role        CST  Bufsize  Bufsize  Segsize  Keepalive
Priority

   1             CLIENT       p    5120     5120     1024    DISABLED    MEDIUM
```

The Add TCP Command: The **add TCP** command is used to define explicitly configured DLSw neighbors. The neighbor DLSw IP Address added here is the internal IP Address of the peer DLSw router (called R2 in “Sample DLSw Configuration” on page 27-23). You also can configure R2 with the neighbor IP Address of R1, or you can configure R2 to accept dynamic neighbors.

```
     DLSw config>add tcp
Enter the DLSw neighbor IP Address [0.0.0.0]? 128.185.122.234
Connectivity Setup Type (a/p) [p]?
Transmit Buffer Size (Decimal) [5120]?
Receive Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive (E/D) [D]?
Neighbor Priority (H/M/L) [M]?

     DLSw config>list tcp
     Neighbor       CST Xmit Bufsize  Rcv Bufsize   Max Segsize  Keepalive  Priority
     --------       --- ------------  -----------   -----------  ---------  --------
     128.185.234.98  p      5120          5120         1024      DISABLED    MEDIUM
```

Define Each SDLC Link Station: You must define each SDLC link station.

```
DLSw config>add sdlc
Interface # [0]? 2
SDLC Address [C1]?
Source MAC address [4000112402C1]? 4000003174d1
Source SAP in hex [4]?
Destination MAC address [000000000000]? 400000000002
Destination SAP in hex [0]? 4
PU type (2/4/5) [2]?
XID0 block num in hex (0-0xfff) [0]? 017
XID0 id num in hex (0-0xfffff) [0]? 00001
Poll with TEST (T) or SNRM (S) [T]?

DLSw config>li sdlc all
Net Addr  Status    Source SAP/MAC   Dest SAP/MAC      PU  Blk/IdNum  PollFrame
 2   C1   Enabled   04 4000003174D1  04 400000000002   2   017/00001  TEST
```

Define Each QLLC Link Station: Define the address mapping for each PVC and configured SVC. In the example configuration, there is one QLLC device attached to a PVC.

```
DLSw config> add qllc sta
Interface # [0]? 3
PVC or SVC [PVC]?
Logical channel number (1-4095) [0]? 4
Source MAC address [400000310101]? 400000317402
Source SAP in hex [4]?
Destination MAC address [000000000000]? 400000000002
Destination SAP in hex [0]? 4
PU type (2/4/5) [2]?
XID0 block num in hex (0-0xfff) [0]? 017
XID0 id num in hex (0-0xfffff) [0]? 00001
New QLLC station record added
```

```
DLSw config> li q st
If  P/S  LCN/DTE addr   E/D Source SAP/MAC   Dest SAP/MAC    PU Blk/IdNum
 3  PVC  4               E  04 400000317402  04 400000000002  2 017/00001
```

Open Service Access Points (SAPs): The next thing to do is open service access points (SAPs) on each of the bridging interfaces.

SAP numbers 0, 4, 8, and C are commonly used SNA SAPs. To open all of these SAPs, use the SNA option with the **open-sap** command as shown. To open SAPs for NetBIOS, choose the NB option. If you prefer, you can also enter SAPs individually by entering a hexadecimal number.

```
DLSw config> open-sap
Interface #[1]?
Enter SAP in hex (range 0–F4), 'SNA', 'NB', or LNM [4]? sna
SAPs 0 4 8 C opened on interface 1

DLSw config>
```

The following is the DLSw display after configuring.

```
DLSw config>list dls

DLSw is                           ENABLED
LLC2 send Disconnect is           ENABLED
Dynamic Neighbors                 ENABLED

SRB Segment number                020
MAC <-> IP mapping cache size     128
Max DLSw sessions                 1000
DLSw global memory allotment      141056
LLC per-session memory allotment  8192
SDLC per-session memory allotment 4096
QLLC per-session memory allotment 4096
NetBIOS UI-frame memory allotment 40960

Dynamic Neighbor Transport Buffer Size 5120
Dynamic Neighbor Receive Buffer Size   5120
Dynamic Neighbor Maximum Segment Size  1024
Dynamic Neighbor Keep Alive            DISABLED
Dynamic Neighbor Priority              MEDIUM

QLLC base source MAC address           40514C430000
QLLC maximum dynamic addresses         64
```

When you have finished configuring DLSw, exit the DLSw configuration and restart the router.

```
DLSw config>exit
Config (only)>restart
Are you sure you want to restart the gateway? (Yes or [No]): yes
```

Accessing the DLSw Configuration Environment

Use the CONFIG process to change the configuration of the router. The new configuration takes effect when the router is restarted.

To enter the configuration process, enter **talk 6** (or **t 6**), at the OPCON (*) prompt. This brings you to the `CONFIG>` prompt as shown in the following example:

```
MOS Operator Control

* talk 6
Gateway user configuration

CONFIG>
```

If the CONFIG> prompt does not appear immediately, press the **Enter** key again.

All DLSw configuration commands are entered at the DLS config> prompt. To access this prompt, enter the **protocol DLSw** command as shown:

```
Config>protocol dls
DLSw protocol user configuration
DLSw config>
```

Pre-Configuration Requirements

Before you begin any configuration procedure, use the **list device** command from the **config** prompt to list the interface numbers of different devices. If you need any further configuration command explanations, see the configuration commands described in this chapter.

DLSw Configuration Commands

This section summarizes and explains the DLSw configuration commands. The DLSw configuration commands allow you to create or modify a DLSw configuration. Table 27-2 provides a brief summary of each command. Enter all the DLSw configuration commands following the `DLSw Config>` prompt. Defaults for any command and its parameters are enclosed in brackets immediately following the prompt.

Changes made to the router’s configuration do not take effect immediately, but become part of the router's SRAM configuration when it is restarted.

Table 27-2. DLSw Configuration Commands Summary

Command	Function
? (Help)	Lists the configuration commands or lists any parameters associated with that command.
Add	Adds an SDLC link station, a TCP neighbor IP address, or a QLLC station or destination.
Ban	Allows access to the Boundary Access Node (BAN) configuration prompt so that BAN configuration commands can be entered.
Close-Sap	Closes a currently-opened service access point (SAP). DLSw uses SAPs for communication on interfaces that support LLC.
Delete	Removes configured SDLC link station, TCP connections, or a QLLC station or destination.
Disable	Disables the DLSw protocol, SDLC link station, LLC disconnect function, dynamic neighbors, or a QLLC station or interface.
Enable	Enables the DLSw protocol, SDLC link station, LLC disconnect function, dynamic neighbors, or a QLLC station or interface.
Join-Group	Allows DLSw neighbors to dynamically find each other.
Leave-Group	Removes the router from the specified DLSw group.
List	Displays information for SDLC link stations, SAPs, circuit priority, DLSw groups, DLSw global information, and QLLC destinations, stations, and interfaces. The command also provides detailed information on TCP connections.
NetBIOS	Provides access to the NetBIOS configuration prompt.
Open-SAP	Allows DLSw to transmit data over the specified SAP. DLSw uses SAPs for communication on interfaces that support LLC.
Set	Configures LLC2 parameters, number of DLSw sessions, SRB segment number, TCP buffer size, memory allocation, protocol timers, circuit priority, parameters for dynamic neighbors, and parameters for QLLC operation.
Exit	Exits the DLSw configuration process and returns you to the `Config>` prompt.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADD
BAN
CLOSE-SAP
DELETE
DISABLE
ENABLE
JOIN-GROUP
LEAVE-GROUP
LIST
NETBIOS
OPEN-SAP
SET
EXIT
```

Add

Use the **add** command to configure an SDLC link station, a TCP neighbor IP address, or a QLLC station or destination.

Syntax: add qllc...
sdlc
tcp

qllc

Adds support for a QLLC station on an X.25 network, or for a DLSw destination for QLLC stations. A QLLC station is the local link station representing a QLLC device attached to the router through an X.25 interface. A QLLC destination is an address mapping that points to a device in the DLSw network. That device is attached to a neighbor DLSw router via any of its supported DLC types, and frequently will not be a QLLC device itself.

Syntax: qllc destination
station

Example: **add qllc destination**

```
Enter the connection id (1-8 alphanumeric chars) [ ]? conn1
Destination MAC address [000000000000]? 400031740000
Destination SAP in hex [4]?
QLLC destination record added/updated
```

Connection id

Alphanumeric character string to be matched by bytes 4-11 of the call user data in incoming Call_Request packets. In many QLLC products, this value is configured as a password.

Destination MAC address

The MAC address to be used as a target for sessions initiated by an incoming QLLC call, where the Call_Request packet matches the above connection id.

Destination SAP

The target SAP to be used for the same type of session.

Example: **add qllc station**

```
Interface # [0]? 1
PVC or SVC [PVC]?
Logical channel number (1-4095) [0]? 2
Source MAC address [400000310104]?
Source SAP in hex [4]?
Destination MAC address [000000000000]? 400011112323
Destination SAP in hex [0]? 4
PU type (2/4/5) [2]?
XID0 block num in hex (0-0xfff) [0]?
XID0 id num in hex (0-0xfffff) [0]?
New QLLC station record added
```

Interface #

The number of the X.25 interface by which the QLLC device is attached to the router.

PVC or SVC

The type of virtual circuit (permanent or switched) by which the QLLC device is to be connected.

Logical channel number

For PVCs, the X.25 channel number to which the QLLC station is subscribed. This field is not applicable to SVCs, which use dynamically assigned channel numbers.

DTE address

For SVCs, the "phone number" by which the QLLC station is known to its X.25 network. This is the called party address for calls being placed by the router, and the calling party address for calls from the QLLC station. This field is not applicable to PVCs, which can be uniquely identified by a fixed logical channel number.

Source MAC address

The Medium Access Control address that represents this QLLC station to the rest of the DLSw network. This is the origin address for DLSw sessions started by the QLLC station, and the target address for sessions started by other devices in the DLSw network.

This address is required for each station and must be unique among all the source MAC addresses for QLLC and SDLC devices configured in the router. To work reliably, it also should be unique among all end-station MAC addresses in the DLSw network. The default value is constructed to be likely to be unique within the network. This and all DLSw MAC addresses are in noncanonical (token-ring) bit order format .

Source SAP

The Service Access Point address paired with the source MAC address. It is used in the same way.

Destination MAC address

The Medium Access Control address that represents a station in the DLSw network to which the QLLC device is to be connected. For PVCs, DLSw attempts to start a session to this target address as soon as the QLLC device is successfully contacted. For SVCs, DLSw attempts to start a session to this target address as soon as the QLLC device places an incoming call.

This address is not required. If you do not configure it, the QLLC station can only be the target of a DLSw session, and not the origin.

Destination SAP

The Service Access Point address paired with the destination MAC address. It is used in the same way. Both the destination MAC address and destination SAP must be non-zero for DLSw to use them as a target for a DLSw session.

PU type

The SNA Physical Unit type of the QLLC station. This may have one of the following values:

2 A PU 2.0 or T2.1 node. This may also represent devices that send XID_1s in response to an XID_null poll.

4 An intermediate SNA controller performing subarea SNA routing functions. These typically run IBM's NCP software in an intermediate network node (INN) mode to another NCP, and is *not* for NCP boundary function connections to PU 2 devices.

5 A host or host with a front-end processor (for example, 37xx with NCP) making a boundary function connection to a PU 2.0 device in the DLSw network. If the host is making a connection to a T2.1 device in the DLSw network, it is preferable, but not required, to configure the host itself as a T2.1 device (that is, PU type=2, XID0 block/id num=0).

XID0 block num

The XID block number field for the router to use when building an XID_0 on behalf of the QLLC station. This field is applicable, and is prompted for, only when the PU type is 2. For T2.1 devices and any PU 2.0 device that can respond on its own to an XID_null poll, this field is optional and should be left zero. If you are unsure, it is safest to fill this in for all PU2.0 QLLC devices, and leave zero for all T2.1 devices. If non-zero, it must match the corresponding PU address field in the IBM NCP switched major node configuration for the link station.

XID0 id num

The XID identifier number field that goes along with the XID0 block number field. It is used for the same purposes and is needed in the same situations.

sdlc

Adds SDLC information specifically for adding an SDLC link station to the configuration on a given SDLC serial interface. The **sdlc** command should be used once for each secondary station on the SDLC line.

Example: **add sdlc**

```
DLSw config>add sdlc
Interface # [0]? 2
SDLC Address [C1]?
Source MAC address [4000112402C1]? 4000003174d1
Source SAP in hex [4]?
Destination MAC address [000000000000]? 400000000002
Destination SAP in hex [0]? 4
PU type (2/4/5) [2]?
XID0 block num in hex (0-0xfff) [0]? 017
XID0 id num in hex (0-0xfffff) [0]? 00001
Poll with TEST (T) or SNRM (S) [T]?
```

Interface #
: The number of the SDLC interface by which the SDLC device is attached to the router.

SDLC Address
: The SDLC address of the link station that you are connecting, between 01–FE.

Source MAC address
: The MAC address for this SDLC PU. This value identifies the attached SDLC station within the DLSw domain. It must be unique among the SDLC and QLLC stations attached to this router, and it should be unique among all LAN, SDLC, and QLLC stations in the DLSw network.

Source SAP in hex
: Along with the source MAC address, represents the SDLC end-station within the DLSw domain.

Destination MAC Address
: The MAC address of the remote link station that you are connecting to. The MAC address is in noncanonical bit order (token-ring) format. This is true even if the remote end-station is on the Ethernet. Use the ASRT console **flip** command to help flip the MAC address, in such cases.

Destination SAP in hex
: Defines the SAP to be used when automatically attempting a connection when the link station comes up. If this SAP is 0, then the link station is in passive mode and does not initiate line establishment. In this case, the destination MAC address is ignored.

PU type
: The SNA Physical Unit type of the SDLC station. This may have one of the following values:

2 A PU 2.0 or T2.1 node.

4 An intermediate SNA controller performing subarea SNA routing functions. Typically, these run IBM's NCP software in an intermediate network node (INN) mode to another NCP, and is *not* for NCP boundary function connections to PU 2 devices.

5 A host, with or without a front-end processor (for example, a 37xx with NCP), making a boundary function connection to a PU 2.0 device in the DLSw network. If the host is making a connection to a T2.1 device in the DLSw network, you must configure the host itself as a T2.1 device (that is, PU type=2, XID0 block/id num=0).

XID0 block num
: The XID block number field for the router to use when building an XID_0 on behalf of the SDLC station. This field is applicable and is prompted for only when the PU type is 2. It is optional, and should be left zero for T2.1 devices and any PU 2.0 device that can respond on its own to an XID_null poll. If you are unsure, it is safest to fill this in for all PU2.0 SDLC devices, and leave zero for all T2.1 devices. If non-zero, it must match the corresponding PU address field in the IBM NCP switched major node configuration for the link station.

XID0 id num

The XID identifier number field that goes along with the XID0 block number field. It is used for the same purposes and is needed in the same situations.

tcp

Adds the internal address of a DLSw peer with which this DLSw can make a connection.

Example: **add tcp**

```
Enter the DLSw neighbor IP Address [0.0.0.0]? 128.185.14.1
Connectivity setup type (A/P) [P]?
Transmit Buffer Size (Decimal) [5120]?
Receive Buffer Size (Decimal)  [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive? (E/D) - [D]?
Neighbor Priority (H/M/L) [M]?
```

Enter the DLSw neighbor IP Address

6 Indicates the IP address of the remote DLSw peer in the IP network to which you want to make a connection.

Connectivity setup type

Indicates whether the TCP connection to this DLSw should be made at router startup (Active), or as needed (Passive). For an overview of these options, see “TCP Connections, Neighbor Discovery, and Multicast Exploration” on page 27-4.

Transmit Buffer Size

The size of the packet transmit buffer from 1024 to 32768. Default is 5120.

Receive Buffer Size

The size of the packet receive buffer from 1024 to 32768. The default size is 5120.

Maximum Segment Size

The maximum size of the TCP segment from 64 to 16384. The default is 1024.

Enable/Disable Keepalive (E/D)

Indicates whether you want DLSw to send TCP connection keepalive messages. The default is D (Disable).

Neighbor Priority

Allows you to specify the neighbor priority as High, Medium, or Low. If a destination station is reachable through several neighbor routers with different priorities, DLSw tries to establish circuits to that station through the highest-priority neighbor.

BAN

Use the **ban** command to access the Boundary Access Node (BAN) configuration prompt. BAN commands are entered at the BAN configuration prompt (`BAN config>`). See “BAN” on page 7-12 for an explanation of each of these commands.

Syntax: ban

Example: **ban**

```
BAN config>
```

Close-Sap

Use the **close-sap** command to disable DLSw switching for the specified service access point (SAP). These SAPs are used by LLC for configuration on the network.

Syntax: close-sap

Example: **close-sap**

```
Interface #[1]?
Enter SAP in hex (range 0–F4), 'SNA', 'NB', or LNM [0]? sna
SAPs 0 4 8 C closed on interface 1
Closing SAP 0 disables all SNA DLSw function on interface 1
```

Interface #

The interface number used by the open SAP.

Enter SAP

You can enter individual SAPs in hex or you can enter SNA, NB (NetBIOS), or LNM (LAN Network Manager).

If you enter SAPs in hex, the range is 0 to FE and the SAP must be an even number.

If you enter SNA, SAPs 0, 4, 8, and C are closed.

If you enter NB, SAP F0 is closed.

If you enter LNM, SAPs 0, 2, D4, F2, F4, F8, and FC are closed.

Delete

Use the **delete** command to remove an SDLC link station, a TCP neighbor IP address, or a QLLC station or destination from the DLSw configuration.

Syntax: delete qllc...
sdlc
tcp

qllc

Removes support for a QLLC station on an X.25 network, or for a DLSw destination for QLLC stations.

Syntax: qllc destination
station

Example: **del q destination**

```
DLSw config>del qllc dest
Enter the connection id (1-8 alphanumeric chars) [ ]? conn1
QLLC Destination record deleted
```

Example: **del q station**

```
DLSw config>del qllc st
Interface # [0]? 2
PVC or SVC [PVC]?
Logical channel number (1-4095) [0]? 4
QLLC station record deleted
```

sdlc

Removes the specified SDLC link station from the list of stations that DLSw can provide services to when the router is restarted.

Example: **`delete sdlc`**

```
Interface #[0]? 1
SDLC Address [C1]?
Record deleted
```

Interface #
The interface number of the router that connects to the SDLC link station.

SDLC Address
The SDLC address of the remote link station that you are deleting. Values are in the range 01–FE.

tcp
Removes the IP address (*ip_address*) of the DLSw peer to which you can make a TCP connection.

Example: **`delete tcp`**

```
IP Address [0.0.0.0]? 128.185.14.1
```

Disable

Use the **disable** command to disable the DLSw protocol, an SDLC link station, the LLC disconnect function, dynamic neighbors, or a QLLC station or interface.

Syntax: <u>di</u>sable <u>dl</u>s
<u>dy</u>namic-neighbors
<u>l</u>lc
<u>q</u>llc...
<u>s</u>dlc

dls
Prevents the bridging router from performing DLSw functions over all DLSw configured interfaces.

Example: **`disable dls`**

dynamic-neighbors
Prevents the router from accepting incoming DLSw TCP connections from IP addresses *other than* those of DLSw neighbors that are configured using the **add tcp** command.

Example: **`disable dy`**

llc
Prevents the router from terminating an LLC connection actively by issuing a DISC LLC frame. Instead, it terminates LLC connections passively. This causes the LLC connection at the end-station to detect the link termination. The IBM host responds differently to active and passive disconnections.

This command does not affect switching function for LLC in DLSw. Use the **close-sap** command to stop LLC switching function.

Example: **`disable llc`**

qllc
Specifying “callin” prevents DLSw from accepting incoming QLLC calls on the specified X.25 interface. This is the default state; an interface must be specifically enabled to permit incoming calls to DLSw.

Specifying “station” prevents a configured QLLC station from being the origin or target of DLSw sessions.

Syntax: qllc callin
station

Example: **dis q callin**

```
Select the interface to be disabled for incoming QLLC calls:
Interface # [0]? 1
Interface 1 is now disabled for incoming QLLC calls
```

Example: **dis q station**

```
Interface # [0]? 1
PVC or SVC [PVC]?
Logical channel number (1–4095) [0] 2
This QLLC station has been marked disabled
```

sdlc

Prevents DLSw connections to the specified SDLC Link station.

Example: **disable sdlc**

```
Interface #[0]? 1
SDLC Address [C1]?
Record updated
```

Enable

Use the **enable** command to enable the DLSw protocol, SDLC link station, LLC disconnect function, dynamic neighbors, or a QLLC station or interface.

Syntax: enable dls
dynamic-neighbors
llc
qllc...
sdlc

dls

Enables DLSw operation on the router.

Example: **enable dls**

dynamic-neighbors

Sets the router to accept incoming DLSw TCP connections from IP addresses *other than* those of neighbors configured using the **add tcp** command. This is the default state.

Example: **enable dynamic**

llc

Allows the router to terminate an LLC connection upon the loss of the TCP connection.

Example: **enable llc**

qllc

Specifying “callin” causes DLSw to receive incoming QLLC calls on the specified X.25 interface.

Specifying “station” allows a configured QLLC station to be the origin or target of DLSw sessions. This is the default state of every configured QLLC station.

Syntax: callin
station

Example: **en q callin**

```
Select the X.25 interface to be enabled for incoming QLLC calls:
Interface # [0]? 1
Interface 1 now enabled for incoming QLLC calls
```

Example: **en q station**

```
Interface # [0]? 1
PVC or SVC [PVC]?
Logical channel number (1-4095) [0]? 2
This QLLC station has been marked enabled
```

sdlc

Enables DLSw connections to the specified SDLC link station.

Example: **enable sdlc**

```
Interface #[0]? 1
SDLC Address [C1]?
Record updated
```

Join-Group

Use the **join-group** command to allow DLSw neighbors to dynamically find and to create TCP sessions with each other, and to enable multicast exploring and frame forwarding. For an overview of these functions, see “TCP Connections, Neighbor Discovery, and Multicast Exploration” on page 27-4. To use this command, the IP internet that is being used must support multicast routing, and you must configure OSPF and MOSPF from the *OSPF Config>* prompt.

When you add a DLSw router to a group, you select whether you want to use the group ID model of group identification (where the router constructs the corresponding multicast addresses), or specify the multicast addresses yourself. The group ID model is simpler to configure, but you must specify the multicast addresses yourself if you wish to have multicast connectivity with non-IBM DLSw Version 2 products. A router may be a member of both styles of groups at the same time.

You may join a maximum of 64 groups using the Group ID model. When you assign a DLSw router to a group, the DLSw protocol automatically adds one of two addresses to the group number to form a multicast address. The router transmits the multicast address to identify itself to other group members and to transmit packets to those members. The two addresses that are added to the group number are 225.0.1.0 for DLSw clients and peers, and 225.0.1.64 for DLSw servers. For example, the multicast address for a client in group 2 would be 225.0.1.2.

Syntax: join-group

Example: **join-group**

```
DLSw config>join
Configure group member (G) or specific multicast address (M) -  [G]?
Group ID (1-64 Decimal) [1]? 2
Client/Server or Peer Group Member(C/S/P)- [P]? c
Connectivity Setup Type (a/p) [p]?
Transmit Buffer Size (Decimal) [5120]?
Receive Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive (E/D) [D]?
Neighbor Priority (H/M/L) [M]?
```

Group member or specific multicast address
Selects whether you want the router to construct multicast addresses for you, or whether you want to supply the multicast addresses.

Group ID
The number of the group that you want this router to join.

Client/Server or Peer Group Member
The role this router should assume within the group: C for client, S for server, or P for peer.

Connectivity setup type
Indicates whether the router should join the group as an Active or Passive member. This controls when TCP connections are established with other group members, as described in “TCP Connections, Neighbor Discovery, and Multicast Exploration” on page 27-4.

Transmit Buffer Size
The size of the packet transmit buffer from 1024 to 32768. Default is 5120.

Receive Buffer Size
The size of the packet receive buffer from 1024 to 32768. The default size is 5120.

Maximum Segment Size
The maximum size of the TCP segment from 64 to 16384. The default is 1024.

Enable/Disable Keepalive
Indicates whether you want DLSw to send TCP keepalive messages on connections brought up within this group. Default is D (Disable).

Neighbor Priority (H/M/L) [M]?
Allows you to specify the neighbor priority as High, Medium, or Low. If a destination end-station is reachable through several neighbor routers with different priorities, DLSw tries to establish circuits to that end-station through the highest-priority neighbor.

Leave-Group

Use the **leave-group** command to remove the router from a group that was configured using the **join-group** command, or to stop using a configured multicast address.

Leave-group does not affect existing TCP connections belonging to the specified group.

Syntax: leave-group

Example: **leave-group**

```
Configure group member (G) or specific multicast address (M) -  [G]?
Group ID (1-64 Decimal) [1]? 2
```

List

Use the **list** command to display DLSw information on SDLC link stations, circuit priority, SAPs, TCP neighbors, groups, dynamic neighbors, and QLLC stations, destinations, and interfaces.

Syntax: list dls
groups
llc2
open
priority
qllc...
sdlc
tcp
timers

dls

Displays the information that was configured with the **enable** and **set** commands.

Example: `list dls`

(Output from the **list dls** command is the same as the output from the **list dls global** command illustrated on page 28-5.)

groups

Displays group information for this DLSw peer that was configured with the **join-group** command. (For additional information, see the t5 description on page 28-11.)

llc2

Displays the LLC2 parameters that were configured with the **set llc2** command. (For a complete explanation of these parameters see the **set llc2** command later in this chapter.) These parameters are set per interface. If no changes to the LLC2 parameters were made using the **set llc2** command, no output will be generated.

Example: `list llc2`

```
SAP     t1   t2   ti   n2   n3   tw   rw   nw   acc
 0       1    1   30    8    1    2    2    1    0
```

SAP SAP number.

t1 Reply timer.

t2 Receive Ack timer.

ti Inactivity timer.

n2 Maximum retry value.

n3 Number of I-frames received before sending ACK.

tw Transmit window.

rw Receive window.

nw ACKs needed to increment Ww.

acc The current LLC2 implementation does not use access priority. As a result, this parameter always defaults to 0.

open

Displays all open SAPs and their associated interfaces.

Example: `list open`

```
Interface    SAP
0             0
0             4
1             8
```

priority

Lists the circuit priorities selected for SNA and NetBIOS circuits, the transmit ratios between the various circuit priorities, and the largest frame size configured for NetBIOS.

```
DLSw config> list priority
Priority for SNA DLSw sessions is           MEDIUM
Priority for NetBIOS DLSw sessions is       MEDIUM
Message allocation by C/H/M/L priority is   4/3/2/1
Maximum frame size for NetBIOS is           2052
```

Circuit priorities are Critical, High, Medium, or Low. The router uses the priority value you assign to selectively limit the burst-length of specific types of traffic. For example, if you assign SNA traffic a priority of Critical and NetBIOS a traffic priority of Medium, with a message allocation of 4/3/2/1, the router processes 4 SNA frames before it processes 2 NetBIOS frames, and so on. In this example, two thirds of available bandwidth is dedicated to SNA traffic. When the router allocates bandwidth using priorities you specify, *it counts frames rather than bytes*.

qllc...

Lists QLLC interfaces, destinations, or stations.

Syntax: qllc callin
destination
station

Example: `li q callin`

```
Interfaces enabled for incoming QLLC calls to DLSw:
   1
```

Example: `li q destination`

```
Connection ID   Dest  SAP/MAC
  CHICAGO       04   400000112323
```

For a description of the parameters, see the **add qllc destination** command on page 27-38.

Example: `li q station`

```
1f  P/S  LCN/DTE addr  E/D Source SAO/MAC   Dest Sap/MAC     PU Blk/IdNum
 1  PVC  2              E  04 400000310104  04 400011112323   2 000/00000
 1  PVC  4              E  04 400000317402  04 400000000002   2 017/00001
 1  SVC  3721111        E  04 400000310103  00 000000000000   2 000/00000
```

The parameters listed here are discussed on page 27-39. The “E/D” indicates whether the station was disabled via the **disable qllc station** command.

sdlc

Displays the SDLC link station information that was configured with the add sdlc link station command.

Example: `list sdlc all`

```
Net  Addr   Status    Source SAP/MAC    Dest SAP/MAC     PU Blk/IdNum  PollFrame
 2    C1    Enabled   04 4000003174D1   00 400000000002   2 000/00000   TEST
 2    C2    Enabled   04 4000103D01C2   00 000000000000   4
 2    C3    Enabled   04 4000103D01C2   00 000000000000   2 017/00001   SNRM
```

Net
: The ID number of the interface that connects to the SDLC link station.

Addr
: The SDLC address, in the range 01 – FE, of the connecting link station.

Status
: The state, enabled or disabled, of the link station.

Source SAP/MAC
: The LLC SAP and the MAC addresses that represent the attached SDLC station to the DLSw domain.

Dest SAP/MAC
: The LLC SAP and the MAC addresses of a remote end-station to which the attached SDLC station will initiate circuit establishment when the SDLC station becomes active.

PU
: The SNA PU type of the attached SDLC device, as follows:

2 A PU 2.0 or T2.1 node

4 A PU 4 performing INN subarea routing to another PU 4 (that is, NCP-to-NCP)

5 A host or host with a front-end processor (for example, 37xx with NCP) making a boundary function connection to a PU 2.0 device in the DLSw network

Blk/IdNum
: The XID0 Block number and Id number that the router uses to generate an XID0 on behalf of the attached SDLC device. This field is displayed only for PU type 2 devices.

PollFrame
: The type of SDLC frame that the router uses to make initial contact with the SDLC station, either a TEST frame, or an SNRM. This field is displayed only for PU type 2 devices.

tcp
: Displays configured DLSw TCP neighbors. The neighbors were configured with the **add tcp** command.

Example: `list tcp`

```
Neighbor        CST  Xmit Bufsize  Rcv Bufsize  Max Segsize   Keepalive  Priority
128.185.236.49   A       5120        5120          1024        DISABLED   MEDIUM
```

Neighbor
: The IP address of the TCP neighbor

CST
: Connectivity setup type, either Active or Passive.

Xmit Bufsize
: The size of the packet transmit buffer from 1024 to 32768. The default is 5120.

Rcv Bufsize
The size of the packet-receive buffer from 1024 to 32768. The default is 5120.

Max Segsize
The maximum size of the TCP segment from 64 to 16384. The default is 1024.

Keepalive
The status of the keepalive function, enabled or disabled.

Priority
The priority of the neighbor router in the selection process. Neighbor priority is High, Medium, or Low.

timers
The user-specified time to wait for various activities.

Example: list timers

```
Database age timer                1200  seconds
Max wait timer for ICANREACH      20    seconds
Wait timer for LLC test response  15    seconds
Wait timer for SDLC test response 15    seconds
QLLC session retry timer          200   seconds
Join Group Interval               900   seconds
Neighbor priority wait timer      2.0   seconds
Neighbor Inactivity Timer           5   minutes
```

For additional information, refer to the **list timers** command in this chapter.

NetBIOS

Displays the NetBIOS configuration prompt.

For a description of NetBIOS commands, see "NetBIOS Commands" on page 9-24.

Syntax: netbios

Example: netbios

```
NetBIOS Support User Configuration
NetBIOS config>
```

Open-Sap

Issue the **open-sap** command for all SAPs you wish DLSw to use, either as the originator or as the target of DLSw circuits. The commonly-used SNA SAP values are 00, 04, 08, and 0C; all these SAPs can be opened together using the mnemonic "SNA." The NetBIOS SAP is F0 and can be referred to as "NB." SAPs related to the LAN Network Manager function are referred to collectively as "LNM." Open the SAPs for the protocols you select, on the interfaces through which DLSw will be reaching SNA or NetBIOS end stations, LNM, or the bridges that LNM is managing.

Syntax: open-sap

Example: **open-sap**

```
Interface #[1]?
Enter SAP in hex (range 0–F4), 'SNA', 'NB', or LNM [4]? sna
SAPs 0 4 8 C opened on interface 1
```

Interface #
The number of the interface over which you want to open the SAP.

Enter SAP in hex

You can enter individual SAPs in hex or you can enter SNA, NB (for NetBIOS), or LNM (for LAN Network Manager).

If you enter SAPs in hex, the range is 0 to FE and the SAP must be an even number. If you enter SAP 4, 8, or C without having previously opened SAP 0 on the same interface, SAP 0 will be opened automatically for you.

If you enter SNA, SAPs 0, 4, 8, and C are opened.

If you enter NB, SAP F0 is opened.

If you enter LNM, SAPs 0, 2, D4, F2, F4, F8, and FC are opened.

Set

Use the **set** command to configure the size of the MAC address-to-IP address mapping cache, the LLC2 parameters, the maximum number of DLSw sessions, the SRB segment number, protocol timers, TCP receive buffer size, TCP dynamic neighbors, and parameters for QLLC operation.

Syntax: set cache
dynamic-tcp
llc2
maximum
memory
priority
qllc
srb
timers

cache

The **set cache** command allows you to specify the size of the MAC address-to-IP address mapping cache.

DLSw uses information stored in this cache to discover routes to remote stations. The larger the cache, the better the chances of DLSw finding a desired remote station without sending out CANUREACH frames to all known TCP/IP neighbors.

Nonetheless, you should avoid setting this cache size too large. Doing so will use up memory on the router and cut into the memory needed for actual DLSw sessions. The effect will be a reduction in the number of DLSw sessions that can be handled by the router.

Example: `set cache`

```
MAC  IP cache size (4 - 65535) [128]?
```

dynamic-tcp

Allows you to specify various TCP parameters for dynamic neighbor TCP connections (that is, those that connect-in from neighbors not defined by the **add tcp** command). DLSw uses these values only if dynamic neighbors are enabled.

Example: `set dyn`

```
Transmit Buffer Size (Decimal) [ 5120]?
Receive Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive (E/D) [D]?
Neighbor Priority (H/M/L) [M]?
```

For a description of the parameters listed here, see the **add tcp** command on page 27-42.

llc2

Allows you to configure specific LLC2 attributes for a specific SAP.

Example: set llc2

```
Enter SAP in hex (range 0-F0)  [0]?  04
Reply timer (T1) in sec. [1]?
Receive Ack timer (T2) in 100 millisec. [1]?
Inactivity Timer (Ti) in sec. [30]?
Transmit Window (Tw), 1-127, 0=default [2]?
Receive Window (Rw), 127 Max [2]?
Acks needed to increment Ww (Nw) [1]?
Max Retry value (N2) [8]?
Number I-frames received before sending ACK (N3) [1]?
```

Enter SAP in hex

The SAP number that you want to tune. Values in the range 0 - FE.

Reply timer (T1)

This timer expires when the LLC2 peer fails to receive a required acknowledgment or response from the other LLC2 peer.

Receive Ack timer (T2)

The delay it takes to send an acknowledgment for a received I-format frame in milliseconds.

Inactivity Timer (Ti)

This timer expires when the LLC does not receive a frame for a specified time. When this timer expires, the LLC2 peer transmits an RR until the LLC2 peer responds or the N2 retry count is exceeded. Default is 30 seconds.

Transmit Window (Tw)

The maximum number of I-frames that can be sent before receiving an RR. Values in the range 1–127. 0 sets Tw to the default. Default is 2.

Receive Window (Rw)

The maximum number of unacknowledged sequentially numbered I-frames that an LLC2 peer can receive from a remote host.

Acks needed to increment Ww (Nw)

This affects the way the dynamic windowing algorithm works. Specifies the number of Acknowledgments after an error condition. Default is 1. The working window (Ww) is a dynamically changing shadow of the transmit window (Tw). After an LLC error is detected, the working window (Ww) is reset to 1. The 'Acks needed to increment Ww' value specifies the number of acks that the station must receive before incrementing Ww by 1. The Ww will continue to be incremented in this fashion until Ww = Tw.

Max Retry value (N2)

The maximum number of times the LLC2 peer transmits an RR without receiving an acknowledgment when the inactivity timer (Ti) expires.

Number I-frames received before sending ACK (N3)

The value is used with the T2 timer to reduce acknowledgment traffic for received I-frames. This counter is set to a specified value and decrements each time an I-frame is received. When this counter reaches 0 or the T2 timer expires, an acknowledgment is sent.

To ensure good performance, set N3 to a value less than the remote LLC's Tw. Default is 1.

maximum

Sets the maximum number of DLSw sessions that the DLSw protocol can support. This includes both SNA and NetBIOS sessions (circuits).

Example: `set maximum`

```
Maximum number of DLSw sessions (1-60000) [1000]?
```

memory

Allows you to specify the total amount of memory available to DLSw, and the amount of memory available to each DLSw session and for NetBIOS UI-frames. The router uses the per-session and UI-frame values to set limits at which flow control algorithms will begin/stop applying backward pressure to data sources, and begin/stop discarding UI-frame traffic.

The router does not currently use the overall DLSw allocation value, so this can be left at its default. Any DLS.161 messages that refer to the global transmit and receive pools (not the NetBIOS UI-frame pool) can be ignored. Instead of using these logical pools, DLSw pacing algorithms use the status of physical memory to determine the window sizes to advertise.

The LLC, SDLC, and QLLC session allocation values provide per-circuit (end-station pair) limits on the buffering of data flowing from LLC, SDLC, and QLLC-attached devices, respectively, to TCP. When the router reaches these limits, it sends RNRs/RRs to the appropriate end stations. The state of the per-session pools is visible from the DLSw console command **list dlsw memory** as part of the list of active sessions.

Example: `set memory`

```
Number of bytes to allocate for DLSw (at least 2638)[140800]?
Number of bytes to allocate per LLC session [8192]?
Number of bytes to allocate per SDLC session [4096]?
Number of bytes to allocate per QLLC session [4096]?
Number of bytes to allocate for NetBIOS UI-frames [40960]?
```

The NetBIOS UI-frame allocation controls how many UI-frames (includes NetBIOS DATAGRAM, NAME_QUERY, ADD_NAME_QUERY, and so on) DLSw can buffer at any one time. When at this limit, DLSw discards received NetBIOS UI-frames and they must be retransmitted by the originating end-station. Setting this limit too low can therefore cause intermittent failure of NetBIOS circuit establishment attempts. The router reports a frame discard condition using ELS message DLS.161 (referring to the global NetBIOS UI-frame pool).

priority

Lets you specify the circuit priorities to use for SNA circuits and NetBIOS circuits as well as letting you specify a traffic ratio *between* these priorities. You can use the **set priority** command to specify circuit priority as Critical, High, Medium, or Low (in descending order from Critical to Low). The router uses the priority values you assign to selectively limit the burst-length of specific types of traffic it is transmitting to its neighbors.

This function operates only during periods of congestion, when DLSw messages queue up before being sent to TCP. For example, you might assign SNA traffic a priority of Critical, which corresponds by default to a message allocation value of 4. If you then assign NetBIOS traffic a priority of Medium, which corresponds to a message allocation of 2, the router transmits 4 SNA frames before it transmits 2 NetBIOS frames. When the router processes 2 NetBIOS frames, it processes 4 SNA frames again, and so on. When allocating bandwidth using your assigned priorities, the router counts frames rather than bytes. Also, a particular circuit's priority is negotiated with the neighbor router at circuit bring-up time; consequently, the neighbor router may establish a new circuit's priority using some policy other than one based on configuration values you specified for this router.

You also can use the **set priority** command to set a maximum frame size for all NetBIOS circuits going through this router. NetBIOS end-stations have a tendency to generate the largest frames allowed resulting in a single frame on a low-speed link occupying that link for several seconds, thus adversely affecting interactive SNA traffic. To reduce this effect, you can set a smaller maximum frame size value which the router signals to NetBIOS end stations using standard source-route bridging mechanisms. If you have transparently bridged (TB) segments in your network that are running NetBIOS, set the maximum NetBIOS frame size to at least 1470.

Example: **set priority**

```
Priority for SNA DLSw sessions (C/H/M/L) [M]?
Priority for NetBIOS DLSw sessions (C/H/M/L) [M]?
Message allocation by C/H/M/L priority (4 digits) [4/3/2/1]?
Maximum NetBIOS frame size (516, 1470, 2052, or 4399) [2052]?
```

qllc

Lets you specify a range of dynamically-assigned MAC addresses that are used as the origin MAC address for incoming dynamic QLLC calls.

You specify the range by providing a base MAC address “X” for the range, and a maximum number “N” of dynamic addresses. DLSw chooses MAC addresses in the range X to X+(N–1).

Syntax: qllc

Example: **set qllc**

```
QLLC base MAC address [40514C430000]?
Maximum QLLC dynamic addresses (0-max sess) [64]?
```

srb

Sets the Source Routing Bridge (SRB) segment number that identifies DLSw on token-ring networks. Specify the segment number as a three-digit hexadecimal value.

Example: **set srb**

```
Enter segment number hex (1-FFF) [5]?
```

timers

Sets the DLSw protocol timers.

Example: **set timers**

```
DLSw config>set timers
Database age timeout (0-10000 secs. Decimal) [1200]? 480
Max wait timer ICANREACH (1-1000 secs. Decimal) 20?
Wait timer LLC test response (1-1000 secs. Decimal) [15]?
Wait timer SDLC test response (1-1000 secs. Decimal) [15]?
QLLC session retry timer (1-1000 secs. Decimal) [20]?
Group join timer interval (1-60000 secs. Decimal) [900]? 180
Neighbor priority wait timer (1.0-5.0 secs. Decimal) [2.0]?
Neighbor Inactivity Termination Timer (0-255 minutes) [5]?
```

Database age timeout
: Specifies how long to hold unused DLSw database entries. Database entries map destination MAC addresses into the set of DLSw peers that can reach them. A value of zero indicates that entries in this database should not be aged. This may be useful when running neighbor TCP connections over dial interfaces, but is not generally recommended because it disables a number of other DLSw functions.

Max wait timer ICANREACH
: Specifies how long to wait for an ICANREACH response for a previously transmitted CANUREACH.

Wait timer LLC test response
: Specifies how long to wait for an LLC test response before giving up.

Wait timer SDLC test response
: Specifies how long to wait for an SDLC test response before giving up.

QLLC session retry timer
: The time the router waits before trying again to contact a QLLC station to start up a DLSw session.

Group join timer interval
: The amount of time the router waits before broadcasting clusters of group advertisement messages. This can affect how long it takes for group-based DLSw functions to recover from intermediate router failure, and can affect the amount of overhead required for the multicast function to operate. This value is not used if you configure TCP connections rather than use the IP multicast features of DLSw.

Neighbor priority wait timer
: Amount of time to wait during exploration before selecting a neighbor. This allows a higher priority neighbor to be selected even if it is not the first to respond with an ICANREACH message.

Inactive neighbor termination timer
: The time DLSw waits before taking down an inactive (zero sessions) passive TCP connection.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 28. Monitoring DLSw

This chapter describes the DLSw console commands. These commands take effect immediately but do not become part of router's SRAM configuration. Thus, while monitoring commands allow you to make real-time changes to the router's configuration, these changes are overridden by the SRAM configuration when the router is restarted. Monitoring consists of these actions:

- Monitoring the protocols and network interfaces currently in use by the router.
- Displaying ELS (Event Logging System) messages relating to router activities and performance.
- Making real-time changes to the DLSw configuration without permanently affecting the SRAM configuration.

This chapter includes the following sections:

- "Accessing the DLSw Console Environment"
- "DLSw Console Commands"

Accessing the DLSw Console Environment

To enter the DLS console environment (GWCON process), enter **talk 5** (or **t 5**) at the OPCON (*) prompt and **protocol dls** at the GWCON (+) prompt as shown in the following example:

```
MOS Operator Control

* talk 5
+ protocol dls
DLS>
```

DLSw Console Commands

This section summarizes and explains the DLSw Console commands listed in Table 28-1. Use these commands to gather information from the database.

Table 28-1 (Page 1 of 2). DLSw Console Commands Summary

Command	Function
? (Help)	Displays all the DLSw console commands or any options associated with a specific command.
Add	Dynamically adds an SDLC link station, a TCP neighbor IP address, or a QLLC station or destination to the current configuration.
BAN	Allows you to access the Boundary Access Node (BAN) console prompt for entering specific BAN console commands. See Chapter 4, "Using the Boundary Access Node (BAN) Feature" on page 4-1 for a detailed description.
Close-Sap	Dynamically closes a currently opened LLC SAP. LLC interfaces use SAPs for communication on the network.
Delete	Dynamically removes an SDLC link station, a DLSw session, a TCP neighbor IP address, or a QLLC station or destination.

Table 28-1 (Page 2 of 2). DLSw Console Commands Summary

Command	Function
Disable	Dynamically disables the LLC switching function, an SDLC link station, dynamic neighbors, or a QLLC station or interface.
Enable	Dynamically enables the LLC switching function, an SDLC link station, dynamic neighbors, or a QLLC station or interface.
Join-Group	Dynamically adds the router to a DLSw group that is different from the SRAM configuration.
Leave-Group	Dynamically removes the router from the specified DLSw group.
List	Displays information for SDLC link stations, SAPs, circuit priority, DLSw groups, DLSw sessions, and sessions for QLLC destinations, stations, and interfaces. The command also provides detailed information on TCP capabilities, connections, and statistics.
NetBIOS	Provides access to the NetBIOS Support prompt.
Open-SAP	Dynamically opens an LLC SAP.
Set	Dynamically changes the LLC2 parameters, the maximum DLSw sessions, memory allocation, protocol timers, circuit priority, parameters for dynamic neighbors, or parameters for QLLC operation.
Exit	Exits the DLSw console command process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: ?

```
ADD
BAN
CLOSE-SAP
DELETE
DISABLE
ENABLE
JOIN-GROUP
LEAVE-GROUP
LIST
NETBIOS
OPEN-SAP
SET
EXIT
```

Add

Use the **add** command to dynamically configure an SDLC link station, a TCP neighbor IP address, or a QLLC station or destination without affecting the SRAM configuration.

Syntax: add qllc...
sdlc
tcp

For examples and field descriptions, see the **add** command in the configuration chapter at “Add” on page 27-38.

BAN

Use the **ban** command to access the BAN (Boundary Access Node) console prompt. Enter the **ban** command from the `DLS>` prompt.

Syntax: ban

Example: **ban**

```
BAN>
```

Once you access the BAN console prompt, you can begin entering specific console commands. See Chapter 4, “Using the Boundary Access Node (BAN) Feature” on page 4-1 for an explanation of the BAN console commands.

To return to the `DLSw>` prompt at any time, enter the **exit** command.

Close-SAP

Use the **close-sap** command to dynamically disable DLSw's use of the specified SAP without affecting the DLSw SRAM configuration.

Syntax: close-sap

Example: **close-sap**

```
Interface #[1]?
Enter SAP in hex (range 0-F4), 'SNA', 'NB', or LNM [0]? 04
SAP 4 closed on interface 1
```

(An explanation of the **close-sap** parameters can be found on page 27-43.)

Delete

Use the **delete** command to dynamically remove an SDLC link station, DLSw session, a TCP neighbor IP address, or a QLLC station or destination without affecting the DLSw SRAM configuration. Use of this command also terminates any existing session.

Syntax: delete dls
qllc...
sdlc
tcp

dls

Removes a currently active DLSw session.

Example: **delete dls**

```
Session identifier [1]?
```

qllc

Removes support for a QLLC destination or station. If you delete a station that currently is active, DLSw verifies thet you wish to take down the connection before doing so. Deleting a destination has no effect on existing connections.

Syntax: q destination
station

Example: **del q destination**

```
Enter the connection id (1-8 alphanumeric chars) [ ]? conn1
QLLC Destination record deleted
```

Example: **del q station**

```
Interface # [0]? 2
PVC or SVC [PVC]?
Logical channel number (1-4095) [0]? 4
QLLC station record deleted
```

sdlc

Closes the currently active SDLC link without affecting the SDLC link station configuration information.

Example: **delete sdlc**

```
Interface #[0]? 1
SDLC Address [C1]?
Link closed
```

Interface # The interface number of the router that connects to the SDLC link station.

SDLC Address The SDLC address of the remote link station that you are deleting, in the range 01 – FE.

tcp

Removes the IP address (*ip_address*) of the DLSw peer to which the TCP connection is made. The TCP connection is closed.

Example: **delete tcp**

```
IP Address [0.0.0.0]? 128.185.14.1
```

Disable

Use the **disable** command to dynamically disable the LLC disconnect function, the DLSw protocol, an SDLC link station, dynamic neighbors, or a QLLC station or interface, without affecting the DLSw SRAM configuration. Disabling the ***entire*** DLSw function from the console is not supported.

Syntax: disable dynamic-neighbors
llc
qllc...
sdlc

(Examples using parameters of the **disable** command can be found beginning on page 27-44.)

Enable

Use the **enable** command to dynamically enable the LLC disconnect function, an SDLC link station, dynamic neighbors, or a QLLC station or interface without affecting the DLSw SRAM configuration.

Syntax: enable dynamic-neighbors
llc
qllc...
sdlc

(Examples using parameters of the **enable** command can be found beginning on page 27-45.)

Join-Group

Use the **join-group** command to cause DLSw to start performing neighbor discovery, multicast exploration, and multicast frame forwarding functions.

For additional information and an example, see Chapter 27, “Using and Configuring DLSw” on page 27-1.

Syntax: join-group

Leave-Group

Use the **leave-group** command to cause DLSw to stop performing neighbor discovery, multicast exploration, and multicast frame forwarding functions in the specified group or using the specified multicast address. This change is made without affecting the DLSw SRAM configuration. **Leave-group** terminates existing TCP connections brought up under the specified group or multicast address.

Syntax: leave-group

Example: `leave-group`

```
Configure group member (G) or specific multicast address (M) -  [G]?
Group ID (1-64 Decimal) [1]? 2
```

List

Use the **list** command to display DLSw information on SDLC link stations, circuit priority, SAPs, TCP neighbors, groups, dynamic neighbors, and QLLC stations, destinations, and interfaces.

Syntax: list dls...
groups...
llc2...
priority...
qllc...
sdlc...
tcp...
timers

dls

Displays information that pertains to the DLSw protocol. The options (global, memory, sessions, and cache) for the DLSw parameters are described below and on the following pages.

Global Displays the operational values of configured general DLSw parameters.

Memory Displays configured DLS memory information and current memory usage.

Sessions Displays current DLS session information including source, destination, state, flags, destination IP address, and a session ID.

Cache Lists the addresses in the DLSw MAC address cache.

dls global

Displays DLS global parameter information.

Example: `list dls global`

```
DLSw is                            ENABLED
LLC2 send Disconnect is            ENABLED
Dynamic Neighbors is               ENABLED

SRB Segment number                 020
MAC <-> IP mapping cache size      128
Max DLSw sessions                  1000
DLSw global memory allotment       141056
LLC per-session memory allotment   8192
SDLC per-session memory allotment  4096
QLLC per-session memory allotment  4096
NetBIOS UI-frame memory allotment  40960

Dynamic Neighbor Transmit Buffer Size  5120
Dynamic Neighbor Receive Buffer Size   5120
Dynamic Neighbor Maximum Segment Size  1024
Dynamic Neighbor Keep Alive            DISABLED
Dynamic Neighbor Priority              MEDIUM

QLLC base source MAC address           40514C430000
QLLC maximum dynamic addresses         64
```

DLSw is
Status of the DLSw protocol, enabled or disabled.

LLC2 send disconnect is
Status of preventing the router from terminating an LLC2 connection upon the loss of the TCP connection. Values are enabled or disabled.

Dynamic Neighbors
Indicates whether DLSw is accepting incoming TCP connection attempts from DLSw routers that are not configured (that is, using the **add tcp** command).

SRB Segment number
The SRB segment that identifies DLSw in the RIF.

MAC<->IP mapping cache size
Specifies the size of the MAC-IP mapping cache.

Max DLSw Sessions
The maximum number of DLSw sessions that the DLSw protocol can support (both SNA and NetBIOS sessions).

DLSw global memory allotment
The maximum amount of memory allowed for use by DLSw.

LLC per-session memory allotment
The maximum amount of memory allowed for use by LLC DLSw session.

SDLC per-session memory allotment
The maximum amount of memory allowed for use by each SDLC DLSw session.

QLLC per-session memory allotment
The maximum amount of memory allowed for use by each QLLC DLSw session.

NetBIOS UI-frame memory allotment
The maximum amount of memory allowed for all NetBIOS UI-frames being forwarded by DLSw.

Dynamic Neighbor Transmit Buffer Size
The size of the TCP transmit buffer for dynamic TCP connections.

Dynamic Neighbor Receive Buffer Size
The size of the TCP receive buffer for dynamic TCP connections.

Dynamic Neighbor Maximum Segment Size
The maximum TCP segment size for dynamic TCP connections.

Dynamic Neighbor Keep Alive
Whether TCP keep-alive messages are to be sent on new dynamic TCP connections.

Dynamic Neighbor Priority
The neighbor priority to be used for all new dynamic TCP connections.

QLLC base source MAC address
The lowest MAC address in the range used as source MAC addresses for dynamic incoming QLLC calls (SVCs).

QLLC maximum dynamic addresses
The maximum number of dynamic source MAC addresses that can be in use at any one time for dynamic incoming QLLC calls.

dls sessions all
Displays current dls session information.

Example: `list dls session all`

```
   Source          Destination       State       Flags  Dest. IP Addr   Id
1. 400000000003 04  500000000003 04   Connected          128.185.236.51   2
```

Source The source MAC address and SAP of the session. For sessions with an SDLC, QLLC, or APPN source, the MAC address is replaced by the following character strings so that these sessions can be identified easily:

DLC Type	Characters	Content
SDLC	1-5	"SDLC "
	6-7	Interface number
	8	"-"
	9-10	SDLC station address
	11-12	" "
QLLC	1-5	"QLLC "
	6-7	Interface number
	8	"P" for PVC, or "S" for SVC
	9-12	LCN for PVC, or last 4 bytes of DTE address for SVC
APPN	1-4	"APPN"
	5-12	" "

Destination The destination MAC address of the session.

State The state of the session. The following states can be displayed:

DISCONECT Indicates the initial state with no circuit or connection established.

RSLV_PEND Indicates that the target DLSw either is awaiting an SSP_STARTED indication or follows an SSP_START request.

CIRC_PEND Indicates that the target DLSw is awaiting a SSP_REACHACK response to an SSP_ICANREACH message.

CIRC_EST Indicates that the end-to-end circuit has been established.

CIR_RSTRT Indicates that the DLSw that originated the reset is awaiting the restart of the data link and a SSP_RESTARTED response to an SSP_RESTART message.

CONN_PEND Indicates that the origin DLSw is awaiting an SSP_CONTACTED response to an SSP_CONTACT message.

CONT_PEND Indicates that the target DLSw is awaiting an SSP_CONTACTED confirmation to an SSP_CONTACT message.

CONNECTED Indicates that the circuit is fully active for connection-oriented data transfer.

DISC_PEND Indicates that the DLSw that originated the disconnect is awaiting an SSP_HALTED response to an SSP_HALT message.

HALT_PEND Indicates that the remote DLSw is awaiting an SSP_HALTED indication following an SSP_HALT request.

REST_PEND Indicates that the local DLSw has received a RESTART_DL but not yet returned a DL_RESTARTED.

CIRC_STRT Indicates that the local DLSw has sent a CANUREACH_cs but not yet received an ICANREACH_cs.

HLT_NOACK Indicates that the local DLSw has received a HALT_DL_NOACK but has not completed closing the link station.

Flags Flags can be one of the following:

A - CONTACT MSG PENDING
B - SAP RESOLVE PENDING
C - EXIT BUSY EXPECTED
D - TCP BUSY.
E - DELETE PENDING
F - CIRCUIT INACTIVE

Dest. IP Addr The IP address of the remote DLSw peer.

Id The number used to identify the session. Use this number in any command that requires the session ID.

dls sessions appn

Displays dls session information on sessions that have APPN in this router as an end-point.

Example: **`list dls sess appn`**

```
         Source         Destination      State     Flags    Dest IP Addr    Id
     ---------------  ---------------  ---------  -------   --------------  ----
   1 APPN         04  400000000011 04  CONNECTED           187.7.239.11        0
   2 APPN         04  400000000014 04  CONNECTED           142.7.245.14        1
```

dls sessions ban

Displays current information on BAN sessions

Example: **`list dls session ban`**

```
BAN port number (user 0 for all ports) [0]?
No active sessions
```

dls session dest

Displays dls session information by destination MAC address.

Example: **`list dls session dest`**

```
Destination MAC Address [40000000001]? 50000000003

Source             Destination       State       Flags  Dest. IP Addr  Id
1. 400000000003 04  500000000003 04   Connected          128.185.236.51  2
2. 400000000002 04  500000000003 04   Connected          128.185.236.52  3
```

dls session detail

Displays detailed dls session information.

Example: **`list dls session detail`**

```
           Session Identifier [1]?

   Source           Destination        State       Dest. IP Addr    Id
1. 400000000003 04  500000000003 04    Connected   128.185.236.512  2

          Personality:      TARGET
          XIDs sent:        2
          XIDs rcvd:        0
          Datagrams sent:   0
          Datagrams rcvd:   0
          Info frames sent: 15
          Info frames rcvd: 0
          RIF:              0620 0202 B0B 0
          Local CID:        0136AF74:7E000021
          Remote CID:       014AB030:7E000003
          Priority:         MEDIUM
```

Personality The ORIGINATOR (initiator) or TARGET (recipient) of the connection.

XIDs sent XIDs rcvd The total number of XIDs that this DLSw peer has sent and received from the remote DLSw peer.

Datagrams sent Datagrams rcvd The total number of datagrams that this DLSw peer has sent and received from the remote DLSw peer.

Info frames sent Info frames rcvd The total number of I-frames that this DLSw peer has sent and received from the DLSw peer.

RIF The information that is included in the RIF of the LLC test frame.

Local CID The DLSw circuit ID assigned by this router.

Remote CID The DLSw circuit ID assigned by the neighbor router.

Priority The DLSw circuit priority established for this circuit when it was initiated.

dls session ip

Displays dls sessions to a specified TCP-connected neighbor.

Example: **`list dls session ip`**

```
Enter the DLS neighbor IP address [0.0.0.0]? 128.185.236.512

      Source        Destination      State      Dest. IP Addr     Id
1. 400000000003 04  500000000003 04  Connected  128.185.236.512   2
```

dls sessions nb

Lists information about the current active circuits that support NetBIOS.

Example: **`list dls sessions nb`**

```
      Source        Destination      State      Dest. IP Addr     Id
1. 400000000003 F0  500000000003 F0  Connected  128.185.236.512   2
```

dls session range

The range of dls sessions that you want to display. This number is located to the left of the source MAC address.

Example: **`list dls session range`**

```
          Start [1]?
          Stop  [1]?

   Source          Destination      State       Dest.  IP Addr    Id
1. 400000000003 04  500000000003 04  Connected   128.185.236.512   2
```

dls session src

Displays all dls session information by source MAC Address.

Example: **`list dls session src`**

```
         Source MAC Address [400000000001]?

   Source       Destination      State       Flags Dest. IP Addr  Id
1. SDLC 04   400000000002 04      Connected     10.1.49.401       1
```

Note: In this example source MAC address 400000000001 maps to the “SDLC 04” name. If you do not know the source MAC address required as a parameter for this command, then enter the **list SDLC config all** command to obtain this information.

dls session state

Displays all dls sessions in a specified state.

Example: **`list dls session state`**

```
        DISCONECT = 0,  RSLV_PEND = 1
        CIRC_PEND = 2,  CIRC_EST  = 3
        CIR_RSTRT = 4,  CONN_PEND = 5
        CONT_PEND = 6,  CONNECTED = 7
        DISC_PEND = 8,  HALT_PEND = 9
        REST_PEND = 10  WT_HALTNA = 11
        CIRC_STRT = 12  HLT_NOACK = 13

        Enter state value (0-10) [7]?

   Source            Destination      State       Flags  Dest. IP Addr Id
1. 400000000003 04  10005AF181A4 04  Connected          128.185.236.84  0
2. 400000000002 04  400000000088 04  Connected          128.185.236.84  1
```

dls cache all

The **list dls cache** command lists the entries in the DLSw MAC address cache. This cache contains a database of the most recent MAC address to IP neighbor translations. It provides the MAC address, time to live (in seconds) in the cache, and the neighbor’s IP address.

Example: **`list cache all`**

```
      Mac Address     Secs to live    IP Address(es)    Largest Frame
1.    10005AF1809B    810             128.185.236.84    2052
2.    10005AF181A4    1170            128.185.236.84    2052
3.    400000000088    1170            128.185.236.84    4399
```

dls cache range

Displays information for a specified range of cache entries.

Example: **`list cache range`**

```
Start[2]?

Stop[2]?

      Mac Address   Secs to live   IP Address(es)   Largest Frame
2.    10005AF181A4  1170           128.185.236.84   2052
```

dls memory

This command lists all existing DLSw sessions and the amount of memory in use by each session.

Example: **`list dls memory`**

```
Total DLSw bytes requested:            153600
Global receive pool bytes granted:      92160
  Currently in use:                         0
Global transmit pool bytes granted:     61440
  Currently in use:                       232

NetBIOS UI-frame pool total bytes:      40960
  Currently in use:                         0
```

```
                                        Session    Initial  Current  Congest  DLC Xmits
Id Source           Destination          State      alloc    alloc    State     Queued
5. SDLC 04C1     04  400000000003  04  Connected   16384    16384    READY       0
6. 400000000003  04  0000c9001119  04  Connected   16384    16384    READY       0
```

The “Currently in use” field shows the total amount of memory currently allocated by DLS. This includes all session allocations and control messages.

The “Congest State” field provides information on flow control and can be any of the following:

Ready Indicates that the session is not congested.

Session Indicates that the session has used most of its session allotment and probably has flow controlled the data link.

Global Indicates that the session is congested due to a shortage of memory in the router.

Ses/gbl Indicates that the session is congested due to a combination of session and global memory shortage.

The “DLC Xmits Queued” field shows the total number of frames queued for transmit in DLS to LLC or SDLC, plus the number queued within the DLC awaiting acknowledgement by the attached end station.

groups config

Displays group information for this DLSw peer that was configured with the **join-group** command.

Example: **`list groups config`**

```
   Group# /                        Xmit     Rcv      Max
Mcast IP Addr    Role        CST  Bufsize  Bufsize  Segsize  Keepalive
Priority

224.0.10.0       READ/WRITE   p    5120     5120     1024    DISABLED   MEDIUM
   1             CLIENT       p    5120     5120     1024    DISABLED   MEDIUM
```

Group # / Mcast IP Addr

For client/server/peer groups, the number of the group. For DLSw Version 2 groups, the multicast address is configured to read from or write to.

Role

For client/server/peer groups, the role that this router is configured to assume within the group. For DLSw Version 2 groups, the read/write role of the configured multicast address: read-only, write-only, or read-write.

CST

The Connectivity Setup Type this router is configured to use within the group, either Active (a) or Passive (p).

Xmit Bufsize
The size of the packet transmit buffer from 1024 to 32768. Default is 5120.

Rcv Bufsize
The size of the packet receive buffer from 1024 to 32768. The default is 5120.

Max Segsize
The maximum size of the TCP segment from 64 to 16384. The default is 1024.

Keepalive
Displays the status of the keepalive function, enabled or disabled.

Priority
Displays the priority of the neighbor router in the selection process. Neighbor priority is High, Medium, or Low.

Groups Statistics
Displays statistics on the use of DLSw groups for explorer traffic since the last restart of the router or creation of the group.

Example: `list group stat`

```
Group number    Data pkts   Data Bytes   Ctrl pkts   CURex pkts   NQex pkts
    or               Sent         Sent        Sent         Sent        Sent
Multicast IP@        Rcvd         Rcvd        Rcvd         Rcvd        Rcvd
-------------   ---------   ---------   ----------   ----------   ---------
Group 1                0            0         116           24          10
                       0            0          25           10           2
224.0.10.0             0            0         224           33           0
                       0            0          21            8           0
```

llc2 open
Displays information for all currently open SAPs on interfaces between LLC2 peers.

Example: `list llc2 open`

```
            Interface   SAP
              0          0
              0          4
```

llc2 SAP parameters
Displays LLC2 parameter configuration information. Only configurations that were changed will be displayed. If the **set llc2** command was not used, no output will be generated.

Example: `list llc2 sap parameters`

```
            SAP              t1    t2    ti    n2    n3    tw    rw    nw    acc
            ---              --    --    --    --    --    --    --    --    ---
             0                1     1    30     8     1     2     2     1     0
```

llc2 sessions all
Displays current information for all LLC2 sessions.

Example: `list llc2 sessions all`

```
               SAP   Int.   Remote Addr    Local Addr    State      RIF
            1. 04    6      400000000003   500000000003  CONTACTED  0620 0202 B0B0
```

State The state of the llc session. The following states can be displayed:

DISCONNECTED Indicates the data link control structure exists but no data link is established.

CONNECT_PEND The connect pending state is entered when a test command frame to NULL SAP is received or when a DLC_START_DL command is received from the DLS.

RESOLVE_PEND The resolve pending state is entered when a DLC_RESOLVE_C command has been sent to DLS.

CONNECTED This is a steady state where LLC Type 1 level services are available through the DLS cloud. This state is entered when a DLC_RESOLVE_R command is received from DLS or when a TEST response frame is received from the network.

CONTACT_PEND This state is entered whenever a response to a transmitted or received SABME is outstanding.

CONTACTED This is a steady state that is entered whenever an UA response for a transmitted SABME has been received, or an UA has been previously transmitted for a received SABME. In this state LLC2 information frames are exchanged over the DLS cloud.

DISCONNECT_PENDING This state is entered whenever a DISC command has been transmitted or received, or a DLC_HALT has been received from DLS.

list llc2 sessions ban
Displays current information for LLC2 sessions involving the BAN function.

list llc2 sessions nb
Displays current information for LLC2 sessions carrying NetBIOS protocol traffic.

llc2 sessions range
Displays current information for the selected range of LLC2 sessions.

Example: **list llc2 sessions range**

```
        Start[1]?

        Stop[1]?
    SAP   Int.   Remote Addr   Local Addr    State      RIF
 1. 04    6      400000000003  500000000003  Contacted  0620 0202 B0B0
```

qllc...
Lists QLLC interfaces, destinations, or stations that are enabled.

Syntax: qllc callin
destinations
sessions
stations

Example: **li qllc callin**

```
Interfaces enabled for incoming QLLC calls to DLSw:
   1
```

Example: `li qllc dest`

```
          Connection ID   Dest  SAP/MAC          Hits
            CHICAGO       04   400000112323        0
```

For a description of the configurable fields in this display, see the **add qllc** command in Chapter 27, “Using and Configuring DLSw” on page 27-1. The *Hits* field indicates the number of times that DLSw has used a match between the connection id in an incoming QLLC Call_Request packet and this connection id.

Example: `li qllc sess`

```
If  P/S  LCN/DTE addr    Source SAP/MAC   Dest SAP/MAC      Type   State
 4  PVC  4               04 400000310401  00 000000000000   PERM   NET_DOWN
 4  SVC  3721111         04 400000310402  00 000000000000   STAT   NET_DOWN
    2 Circuits    1 PVC   1 SVC       1 Permanent   1 Static   0 Dynamic
```

For a description of the configurable fields in this display, see the **add qllc** command in Chapter 27, “Using and Configuring DLSw” on page 27-1.

The *Type* field has the following values:

PERM (Permanent) This station definition was part of router configuration the last time the router was started.

STAT (Static) This station definition was added by the user under the DLSw console function after the router was last started.

DYNM (Dynamic) DLSw dynamically created this station definition as a result of an incoming call, or because of the need to place multiple outgoing calls to a single remote DTE address.

The summary line at the bottom of the session list shows how many sessions of each type currently exist.

The *State* field indicates the state of the DLSw connection from a QLLC point of view. These states are different from the main DLS states displayed under the **list dls sess** commands and add information about what is happening on the QLLC interface. Possible values are:

NET_DOWN The X.25 interface is currently down.

PLC_DOWN The X.25 packet layer is currently down.

DISCONNECTED For this and all the following states, the X.25 interface and packet layers are up. In this state, DLSw is waiting for an end station to start connection establishment.

XID_POLL DLSw is polling the QLLC end station with a QXID (XID_null) in an attempt to initially contact the device or recover a lost connection.

SETMODE_POLL DLSw is polling the QLLC end station with a QSM in an attempt to initially contact the device or recover a lost connection.

SENT_EX DLSw has heard from the QLLC end station and is exploring for the appropriate destination in the DLSw network.

CS_PEND DLSw's exploration has been satisfied and has initiated a circuit start request (sent CUR_cs).

CALL_REQ_PEND DLSw has placed a call request out to the QLLC end station and is waiting to see whether the call is answered successfully.

ESTABLISHED The DLSw circuit is in “circuit established” state; it is available for sending and receiving SNA XIDs.

CONTACT_PEND DLSw has sent QSM to the QLLC end station and is awaiting QUA.

CONNECTED The DLSw circuit is all the way up and can carry I-frame end-user data.

DISC_PEND DLSw has requested a circuit disconnect to the QLLC station and is awaiting acknowledgement.

RESET_PEND DLSw has requested a PVC reset or SVC clear call to the QLLC station and is awaiting acknowledgement.

Example: `li qllc sta`

```
If  P/S  LCN/DTE addr   E/D Source SAP/MAC   Dest SAP/MAC    PU Blk/IdNum  Type
 1  PVC  2               E  04 400000310104  04 400011112323  2 000/00000  PERM
 1  SVC  3721111         E  04 400000310103  00 000000000000  2 000/00000  PERM
 1  PVC  4               E  04 400000317402  04 400000000002  2 017/00001  PERM
```

For a description of the configurable fields in this display, see the **add qllc** command in Chapter 27, “Using and Configuring DLSw” on page 27-1. The “E/D” field indicates whether the station is currently enabled. The “Type” field has the same values described above for the **list qllc sessions** command.

sdlc config
Displays configured parameters for the SDLC attached PU.

Example: `list sdlc config`

```
         Interface #, or 'ALL' [0]? all

    Net  Addr   Status    Idblk  Idnum   Source SAP/MAC    Dest SAP/MAC
     1    C1    Enabled    000   00000   04 4000103D01C1   00000000000000
     1    C2    Enabled    000   00000   04 4000103D01C2   00000000000000
```

sdlc sessions
Displays information about all SDLC dls sessions within the router.

Example: `list sdlc sessions`

```
         Net  Addr   Source SAP/MAC    Dest SAP/MAC       OutQ   State
     1.   1    C1    04 4000103D01C1   00 000000000000      0    NET_DOWN
     2.   1    C2    04 4000103D01C2   00 000000000000      0    NET_DOWN
```

Since DLSw and SDLC have the ability to do full XID negotiation, it is possible that the attached SDLC link station will set the link to a different SDLC station address than that configured in the router. When this happens, two SDLC station addresses are shown under the “Addr” column of this display, using the format `xx(yy)`. In this format, `xx` is the station address configured at this router and is still used for all configuration and console commands to refer to this link station. The current operational address that was set by the attached SDLC device is the value `yy` shown in parentheses to the right.

tcp capabilities
Displays the information received from a partner DLSw router in its capabilities exchange message.

Example: **`list tcp capabilities`**

```
Enter the DLSw neighbor IP Address [0.0.0.0]? 1.1.1.2
Vendor ID:                   10005A
Vendor version:              IBM 2216 Nways MNRS (tm) V1 R3.0[]
Initial pacing window:       12
Preferred TCP connections:   1
Supported SAPs:              00 04 08 0C
NetBIOS Exclusivity:         (none supplied)
NetBIOS Name List:           (none supplied)
```

Vendor ID
: The IEEE Organizational Unique Identifier (OUI) of the vendor of the neighbor DLSw. IBM's OUI is X'10005A'.

Vendor version
: A text string that the neighbor DLSw sent to describe itself. "(not available)" indicates that the neighbor implementation did not send such a string.

Initial pacing window
: The number of paced SSP messages this DLSw is allowed to send to the neighbor DLSw upon receiving the initial pacing grant for each new circuit.

Preferred TCP connections
: The number of TCP connections (1 or 2) that this neighbor would like to have. The IBM 2216 adjusts to the requested number, and will have only 1 full-duplex TCP connection to neighbors who request this.

Supported SAPs
: The list of SAPs that the neighbor DLSw has open or will automatically open, on any of its LAN interfaces or representing its attached SDLC stations.

NetBIOS Exclusivity
: Indicates whether the NetBIOS name list sent by this neighbor is to be considered as a complete or partial list of the NetBIOS names that are local to that neighbor. A response of "(none supplied)" indicates that this neighbor did not send a NetBIOS name list as part of its capabilities.

NetBIOS Name List
: Displays all the NetBIOS name qualifiers that this neighbor sent in its NetBIOS name list. A response of "(none supplied)" indicates that this neighbor did not send a NetBIOS name list as part of its capabilities.

tcp config
: Displays the configuration parameters for all configured TCP connections to peer DLSw routers.

Example: **`list tcp config`**

```
Neighbor        CST Xmit Bufsize Rcv Bufsize Max Segsize Keepalive Priority
--------        --- ------------ ----------- ----------- --------- --------
128.185.236.49   p    5120         5120       1024       DISABLED  MEDIUM
```

tcp sessions
: Displays the status of all known TCP sessions to peer DLSw routers.

Example: **`list tcp sessions`**

```
   Group   IP Address      Conn State     CST  Version  Active Sess Sess Creates
   ------- --------------- -------------- ---  -------- ----------- ------------
1          128.185.236.49  ESTABLISHED     p   AIW V1R0      2           4
```

Group
The group through which the neighbor was discovered, if applicable

IP Address
The neighbor IP address used for DLSw

Conn State
The state of the transport connection (which consists of 1 or 2 TCP connections) to this neighbor.

CST
Current Connectivity Setup Type, as follows:

```
a - Locally configured as active
p - Locally configured as passive
A - Locally configured as passive, but operating in active mode
    due to neighbor requirements
D - Not locally configured, but a dynamic neighbor TCP connection
```

Version
The neighbor's DLSw protocol level. May be one of AIW VnRm for AIW standard-compliant routers, RFC1434+ for pre-AIW V1R0 implementations, or UNKNOWN.

Active Sess
The current number of active (in any state) DLSw sessions (circuits) on this transport connection

Sess Creates
The total number of DLSw sessions (circuits) that ever entered the CIRC_EST state, since the last restart of the router or "add tcp" of this transport connection.

tcp statistics
Displays statistics on the use of TCP transport connections since the last restart of the router or "add tcp" of this transport connection.

Example: `list tcp statistics`

```
           Enter the DLSw neighbor IP Address -0.0.0.0-? 192.1.1.3
                                       Transmitted          Received
                                       -----------          -----------
Data Messages                              214                  231
Data Bytes                              372997               413259
Control Messages                            16                   34

CanYouReach Explorer Messages                0                    0
ICanReach Explorer Messages                  0                    0
NameQuery Explorer Messages                  1                    2
NameRecognized Explorer Messages             2                    1
```

timers
The user-specified time to wait for various activities.

Example: `list timers`

```
Database age timer               1200  seconds
Max wait timer for ICANREACH     20    seconds
Wait timer for LLC test response  15    seconds
Wait timer for SDLC test response 15    seconds
QLLC session retry timer         200   seconds
Join Group Interval              900   seconds
Neighbor priority wait timer     2.0   seconds
Neighbor Inactivity Timer          5   minutes
```

Database age timer
The time to hold unreferenced MAC address-to-IP address database entries. Zero indicates that entries in this database are not being timed.

Max wait timer for ICANREACH
The time the router waits for a response to a CANUREACH message before deciding that the session will not come up.

Wait timer for LLC test response
The time the router waits for an LLC test response before retransmitting an LLC test frame.

Wait timer for SDLC test response
The time the router waits before trying again to contact an SDLC station to start a DLSw session.

QLLC session retry timer
The time the router waits before trying again to contact a QLLC station to start a DLSw session.

Join Group Interval
The time between DLSw group advertisement broadcasts.

Neighbor priority wait timer
The time DLSw waits before selecting a neighbor during a given session-establishment attempt.

Neighbor Inactivity Timer
The time DLSw waits before taking down an inactive (zero sessions) passive TCP connection.

NetBIOS

Displays the NetBIOS monitoring prompt.

Syntax: netbios

Example: **netbios**

```
NetBIOS Support User Configuration
NetBIOS config>
```

For a description of NetBIOS commands, see Chapter 9, “Using, Configuring, and Monitoring NetBIOS” on page 9-1.

Open-Sap

Use the **open-sap** command to dynamically enable DLSw switching for the specified service access point (SAP) without affecting the DLSw SRAM configuration.

Syntax: open-sap

Example: **open-sap**

Refer to “Open-Sap” on page 27-51 for additional information and an explanation of the **open-sap** parameters.

Set

Use the **set** command to dynamically change the LLC2 parameters, the maximum number DLSw sessions, protocol timers, TCP dynamic neighbors, and parameters for QLLC operation without affecting the DLSw SRAM configuration.

Syntax: set dynamic-tcp
llc2
memory
priority
qllc
timers

dynamic-tcp

Allows you to specify various TCP parameters for dynamic neighbor TCP connections (that is, those that connect-in from neighbors not defined by the **add tcp** command). DLSw uses these values only if dynamic neighbors are enabled.

Syntax: dynamic-tcp

Example: `set dyn`

.

```
Transmit Buffer Size (Decimal) [ 5120]?
Receive Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive (E/D) [D]?
Neighbor Priority (H/M/L) [M]?
```

For a description of these parameters, see the **add tcp** command in Chapter 27, “Using and Configuring DLSw” on page 27-1.

llc2

Allows you to configure specific LLC2 attributes for a specific SAP.

Example: `set llc2`

(An example of the **set llc2** command can be found on page 27-53.)

memory

This command allows you to dynamically specify the total amount of memory allocated to DLSw, and the total amount of memory to be allotted to each DLSw session.

Example: `set memory`

An example of the use of the **set memory** command can be found on page 27-54.

priority

Allows you to specify the circuit priorities to use for SNA circuits and NetBIOS circuits. You can configure circuit priorities of High, Medium, or Low (in descending order from Critical to Low).

This command also allows you to configure the ratio of transport transmits for each circuit priority, and to set the maximum frame size to use for NetBIOS. If your network contains any transparently bridged (TB) segments, use a maximum NetBIOS frame size of at least 1470.

Example: `set priority`

For more information on the **set priority** command, see page 27-54.

qllc

Lets you specify a range of dynamically-assigned MAC addresses that are used as the origin MAC address for DLSw sessions resulting from incoming dynamic QLLC calls.

Specify the range by providing a base MAC address “X” for the range, and a maximum number “N” of dynamic addresses. DLSw chooses MAC addresses in the range X to X+(N–1).

Syntax: qllc

Example: **`set qllc`**

```
DLSw config>set qllc
QLLC base MAC address [40514C430000]?
Maximum QLLC dynamic addresses (0-max sess) [64]?
```

timers

Sets the DLSw protocol timers.

Example: **`set timers`**

An example of the **set timers** command can be found on page 27-55.

Appendix A. MIB Support

This appendix describes the Management Information Bases (MIBs) or portions of MIBs supported by the IBM 2216 SNMP agent. A MIB is a virtual information store that provides access to management information. This information is defined as MIB objects which can be accessed, and in some cases be modified using network management tools.

This appendix contains the following sections:.

- “How to Access MIBs on the Internet” describes how to access MIB descriptions on the Internet.
- “MIB Support Cross-Reference Table” on page A-2 provides a quick summary of MIB support.
- “Standard MIBs” on page A-3 describes IBM 2216 support for Industry Standard MIBs defined by standards bodies.
- “Enterprise MIBs” on page A-13 describes IBM 2216 support for MIBs that have been defined by enterprises and that have become accepted as standard MIBs. Examples are the Novell IPX/RIP/SAP MIBs.
- “SNMP Generic Traps” on page A-17 describes the IBM 2216 support for generic traps.
- “Enterprise-specific Traps” on page A-17 describes the IBM 2216 support for enterprise-specific traps.
- “Settable Values” on page A-18 describes *settable values* for various MIBs.

  Except for the settable values described in “Settable Values” on page A-18, all supported MIB attributes are supported only in READ-ONLY mode.

How to Access MIBs on the Internet

MIBs supported by the IBM 2216 are accessible through the Internet using anonymous FTP at venera.isi.edu, ds.internic.net, and ftp.nways.raleigh.ibm.com.

The suggested location to obtain MIBs and information is ftp.nways.raleigh.ibm.com. Along with product and release-specific network management information, all attempts have been made to correct errors in the MIBs and to fix any problems when loading MIBs into a network management station.

Note: If you run into problems accessing ftp.nways.raleigh.ibm.com, try the “alias” address, 6611ftp.raleigh.ibm.com, or IP address 204.146.167.70. However, the IP address may change without notice, so try the host names first.

1. To connect to the various FTP servers, enter the appropriate command:

   ```
   For various enterprise MIBs       -- %ftp venera.isi.edu
   For RFC defined MIBs              -- %ftp ds.internic.net
   For MIBs and information          -- %ftp ftp.nways.raleigh.ibm.com
   ```

   At the `name` prompt, enter **anonymous**. You will be prompted for a password. Enter your Internet mail address.

2. Change to the MIBs directory by entering:

```
For various enterprise MIBs       -- %cd /mib
For RFC defined MIBs              -- %cd /rfc
For MIBs and information          -- %cd /pub/netmgmt
```

3. To receive a copy of a specific MIB, enter the **get** command together with the name of the MIB. As an example:

```
For MIB defined by number ####    -- % get rfc####.mib
                                  -- % get ibm2216.mib
                                  -- % get novell-ipx.mib
                                  -- % get novell-rip-sap.mib
```

   Using this command places a copy of the specified MIB in the directory from which you connected to the FTP server.

MIB Support Cross-Reference Table

IBM Nways Multiprotocol Access Services supports the following MIBs shown in Table A-1.

Table A-1 (Page 1 of 2). Supported MIBs

MIB	Description / Cross-Reference	Source
AppleTalk Phase 2	Refer to Table A-2 on page A-3 for a list of supported groups.	RFC 1243
ATM	See Table A-4 on page A-4.	RFC 1695
APPN	See Table A-29 on page A-13	APPN Implementers Workshop (AIW)
Bandwidth Reservation	See Table A-30 on page A-13.	Proteon BRS MIB
BGP4	See Table A-6 on page A-4.	RFC 1657
Bridge	See Table A-7 on page A-5. See also the Source Route Bridge, Table A-23 on page A-10.	RFC 1493
DLSw	See Table A-8 on page A-5.	RFC 2024
DLUR	See Table A-31 on page A-14.	IBM Enterprise MIB
LAN Emulation Client	See Table A-14 on page A-7.	ATM Forum LEC Management Specification
Ethernet	See Table A-10 on page A-6.	RFC 1650
Frame Relay	See Table A-11 on page A-6.	RFC 1315
HPR	High Performance Routing. See Table A-32 on page A-14.	APPN Implementers Workshop (AIW)
IBM Common Routing	See Table A-34 on page A-15.	IBM Enterprise MIB
IBM 2216 Platform MIB	Info relating specifically to the IBM 2216 platform. See Table A-33 on page A-14.	IBM 2216 Enterprise MIB
Interface MIB	See Table A-12 on page A-6.	RFC 1573
IPX	See Table A-35 on page A-15.	Novell IPX MIB
LLC	See Table A-16 on page A-8.	IETF Draft 01

Table A-1 (Page 2 of 2). Supported MIBs

MIB	Description / Cross-Reference	Source
MIB2	See Table A-17 on page A-8.	RFC 1213
OSPF	See Table A-18 on page A-8.	RFC 1253
PPP	See Table A-20 on page A-9.	RFC 1471
Proteon	See Table A-38 on page A-16.	Proteon Enterprise MIB
Remote Ping	See Table A-39 on page A-16.	IBM Enterprise MIB
RIP-SAP	See Table A-37 on page A-16.	Novell RIP-SAP MIB
RS-232 Serial Line	See Table A-21 on page A-9.	RFC 1317
SDLC	See Table A-22 on page A-10.	RFC 1747
SNA NAU	See Table A-24 on page A-10.	RFC 1666
Source Route Bridge	See Table A-23 on page A-10. See also the Bridge MIB, Table A-7 on page A-5.	RFC 1525
X.25 LAPB	See Table A-26 on page A-11.	RFC 1381
X.25 Packet	See Table A-27 on page A-11.	RFC 1382
X.25 MIOX	The IBM 2216 supports all MIOX MIB groups. See Table A-28 on page A-12.	RFC 1461
Token Ring	See Table A-25 on page A-11.	RFC 1748

Standard MIBs

This section describes the standard MIBs supported by the IBM 2216.

AppleTalk MIB

Table A-2 lists the IBM 2216-supported groups for the AppleTalk Phase 2 MIB. These groups are defined by Internet standard RFC 1243. Table A-3 on page A-4 lists exceptions.

Table A-2. AppleTalk Phase 2 Supported MIB Groups

Group Name	Supported	Not Supported
AppleTalk Address Resolution Protocol (AARP)	√	
AppleTalk Echo (ATEcho)	√	
AppleTalk Port (ATPort)	√	
Datagram Delivery Protocol (DDP)	√	
Routing Table Maintenance Protocol (RTMP)	√	
Zone Information Protocol (ZIP)	√	
LocalTalk Link Access Protocol (LLAP)		√
Name Binding Protocol (NBP)		√
Kinetics Internet Protocol (KIP)		√

Table A-3. Exceptions for AppleTalk MIB

Group or Field Name	Exception
atportDescr	Always null

ATM MIB

Table A-4 lists the IBM 2216-supported groups for ATM. These groups are defined by Internet standard RFC 1695. Table A-5 lists exceptions.

Table A-4. ATM Supported MIB Groups

Group Name	Supported	Not Supported
aa15VccTable		√
atmInterfaceConfTable	√	
atmInterfaceDs3PlcpTable		√
atmInterfaceTCTable		√
atmTrafficDescrParamTable	√	
atmVcCrossConnectIndexNext		√
atmVcCrossConnectTable		√
atmVclTable	√	
atmVplTable		√
atmVpCrossConnectIndexNext		√
atmVpCrossConnectTable		√

Table A-5. Exceptions for ATM MIB

Group or Field Name	Exception
atmInterfaceMyNeighborIpAddress	Always null
atmInterfaceMyNeighborIfName	Always null

BGP4 MIB

Table A-6 lists the IBM 2216-supported groups for BGP4. These groups are defined by Internet standard RFC 1657.

Table A-6. BGP4 Supported MIB Groups

Group Name	Supported	Not Supported
bgp4PathAttrTable		√
bgpLocalAs	√	
bgpPeerTable	√	
bgpRcvdPathAttrTable		√
bgpVersion	√	

Bridge MIB

Table A-7 lists the 2216-supported groups for the Bridge MIB. These groups are defined by Internet standard RFC 1493.

Table A-7. Bridge Supported MIB Groups

Group Name	Supported	Not Supported
dot1dBase	√	
dot1dBasePortTable	√	
dot1dStp	√	
dot1dStpPortTable	√	
dot1dTp	√	
dot1dTpFdbTable		√
dot1dTpPortTable	√	
dot1dStaticTable		√

DLSw MIB

Table A-8 lists the supported groups for the DLSw MIB. These groups are identified by Internet Standard RFC 2024. Table A-9 on page A-6 lists exceptions.

Table A-8. DLSw Supported MIB Groups

Group Name	Supported	Not Supported
dlswNode	√	
dlswTConnStat	√	
dlswTConnConfigTable	√	
dlswTConnOperTable	√	
dlswTConnTcpConfigTable	√	
dlswTConnTcpOperTable	√	
dlswIfTable	√	
dlswDirStat	√	
dlswDirMacTable	√	
dlswDirNBTable	√	
dlswDirLocateMacTable	√	
dlswDirLocateNBTable	√	
dlswCircuitStat	√	
dlswCircuitTable	√	
dlswSdlcLsEntries	√	
dlswSdlcLsTable	√	
dlswTrapControl	√	
dlswTraps		√

Table A-9. Exceptions for DLSw MIB

Group or Field Name	Exception
dlswCircuitDiscReasonLocal	Not supported
dlswCircuitDiscReasonRemote	Not supported
dlswCircuitDiscReasonRemoteData	Not supported

Ethernet MIB

Table A-10 lists the 2216-supported groups for the Ethernet MIB. These groups are defined by Internet standard RFC 1650.

Table A-10. Ethernet Supported MIB Groups

Group Name	Supported	Not Supported
dot3StatsTable	√	
dot3CollTable	√	
dot3Tests		√

Frame Relay MIB

Table A-11 lists the supported groups for the Frame Relay MIB defined by RFC 1315.

Table A-11. Frame Relay Supported MIB Groups

Group Name	Supported	Not Supported
frCircuitTable	√	
frDlcmiTable	√	
frErrTable	√	
frTrapState	√	

Interfaces MIB

Table A-12 lists the supported groups for the Interface MIB defined by RFC 1573. Table A-13 on page A-7 lists exceptions.

Table A-12. Interface MIB Supported Groups

Group Name	Supported	Not Supported
ifNumber	√	
ifTable	√	
ifXTable	√	
StackTable	√	
ifTestTable		√
ifRcvAddressTable		√

Table A-13. Exceptions for Interface MIB

Group or Field Name	Exception
ifInBroadcastPkts	Always 0
ifOutBroadcastPkts	Always 0
ifHCIn*	Always 0
ifHCOut*	Always 0
ifInNUcastPkts	Deprecated
ifOutNUcastPkts	Deprecated
ifOutQLen	Deprecated
ifSpecific	Deprecated

LAN Emulation Client MIB

Table A-14 lists the supported groups for the LAN Emulation Client MIB defined by ATM Forum/94 - 0737R4. Table A-15 lists exceptions.

Table A-14. LAN Emulation Client Supported MIB Groups

Group Name	Supported	Not Supported
lecConfigTable	√	
lecStatusTable	√	
lecMappingTable	√	
lecServerVccTable	√	
lecAtmAddressTable	√	
LecMacAddressTable	√	
LecRouteDescrTable	√	
lecStatisticsTable	√	
LeArpTable	√	
LeRDArpTable	√	

Table A-15. Exceptions for LAN Emulation Client MIB

Group or Field Name	Exception
lecOwner	Always a null string

LLC MIB

Table A-16 on page A-8 lists the supported groups for the LLC MIB. These groups are defined in Version 01 of an Internet draft standard dated 20 October 1994.

Table A-16 on page A-8 lists the supported groups for the LLC MIB.

Table A-16. LLC Supported MIB Groups

Group Name	Supported	Not Supported
llcPortAdminTable	√	
llcPortOperTable	√	
llcPortStatsTable		√
llcSapAdminTable	√	
llcSapOperTable	√	
llcSapStatsTable	√	
llcCcAdminTable		√
llcCcOperTable	√	
llcCcStatsTable	√	
llcTraps		√

MIB II

The router software supports the base MIB objects known as Management Information Base II (MIB II) as specified in the Internet standard RFC 1213. Table A-17 lists supported groups for MIB II objects.

Table A-17. MIB II Supported Groups

Group Name	Supported	Not Supported
System Group	√	
Interfaces Group	√	
Address Translation Group	√	
IP Group	√	
ICMP Group	√	
UDP Group	√	
EGP Group		√
SNMP Group	√	
TCP Group	√	

OSPF MIB

Table A-18 lists supported groups for the OSPF MIB defined by Internet Standard RFC 1253. Table A-19 on page A-9 lists exceptions.

Table A-18 (Page 1 of 2). OSPF Supported MIB Groups

Group Name	Supported	Not Supported
ospfGeneralGroup	√	
ospfAreaTable	√	
ospfStubAreaTable	√	
ospfLsdbTable	√	
ospfAreaRangeTable	√	

Table A-18 (Page 2 of 2). OSPF Supported MIB Groups

Group Name	Supported	Not Supported
ospfHostTable	√	
ospfIfTable	√	
ospfIfMetricTable	√	
ospfVirtIfTable	√	
ospfNbrTable	√	
ospfVirtNbrTable	√	

Table A-19. Exceptions for OSPF MIB

Group or Field Name	Exception
ospfLsdbType	Can be multicastLink(6)

PPP MIB

Table A-20 lists the groups of the PPP MIB supported by 2216. These groups are defined by Internet standard RFC 1471.

Table A-20. PPP Supported MIB Groups

Group Name	Supported	Not Supported
PPPLinkStatusTable	√	
PPPLinkConfigTable		√
PPPLqrTable		√
PPPLqrConfigTable		√
PPPLqrExtnsTable		√

RS-232 MIB

Table A-21 lists the supported groups for the RS-232 MIB defined by Internet Standard RFC 1317.

Table A-21. RS-232 Supported MIB Groups

Group Name	Supported	Not Supported
rs232Number	√	
rs232PortTable	√	
rs232AsyncPortTable		√
rs232SynchPortTable	√	
rsInSigTable	√	
rs232OutSigTable	√	

SDLC MIB

Table A-22 lists the supported groups for the SDLC MIB. These are defined in the Internet Standard RFC 1747.

Table A-22. SDLC Supported MIB Groups

Group Name	Supported	Not Supported
sdlcPortAdminTable	√	
sdlcPortOperTable	√	
sdlcPortStatsTable	√	
sdlcLSAdminTable	√	
sdlcLSOperTable	√	
sdlcLSStatsTable	√	

Source Route Bridge MIB

Table A-23 lists the supported groups for the Source Route Bridge MIB defined by RFC 1525.

Table A-23. Source Route Bridge Supported MIB Groups

Group Name	Supported	Not Supported
dot1dPortPairTable		√
dot1dPortPairTableSize		√
dot1dSrBridgeLfMode	√	
dot1dSrPortTable	√	

SNA NAU MIB

Table A-24 lists the IBM 2216-supported groups for SNA NAU. These groups are defined by Internet standard RFC 1666.

Table A-24 (Page 1 of 2). SNA NAU Supported MIB Groups

Group Name	Supported	Not Supported
SnaLuAdminTable	√	
SnaLuOperTable	√	
SnaLuRtmTable		√
SnaLuSessnStatsTable	√	
SnaLuSessnTable	√	
SnaLuTraps		√
SnaNodeAdminTable	√	
SnaNodeAdminTableLastChange	√	
SnaNodeOperTable	√	
SnaNodeOperTableLastChange	√	
SnaPu20StatsTable	√	

Table A-24 (Page 2 of 2). SNA NAU Supported MIB Groups

Group Name	Supported	Not Supported
SnaNodeLinkAdminTable	√	
SnaNodeLinkAdminTableLastChange	√	
SnaNodeLinkOperTable	√	
SnaNodeLinkOperTableLastChange	√	
SnaNodeTraps		√

Token-Ring MIB

Table A-25 lists the IBM 2216-supported groups for the Token-Ring MIB. These groups are defined by Internet standard RFC 1748.

Table A-25. Token-Ring Supported MIB Groups

Group Name	Supported	Not Supported
dot5Table	√	
dot5StatsTable	√	
dot5TimerTable		√

X.25 LAPB

Table A-26 lists the supported groups for the X.25 LAPB MIB defined by Internet Standard RFC 1381.

Table A-26. X.25 LAPB Supported MIB Groups

Group Name	Supported	Not Supported
lapbAdminTable		√
lapbFlowTable	√	
lapbOperTable	√	
lapbXIDTable		√

X.25 Packet

Table A-27 lists the supported groups for the X.25 Packet MIB defined by Internet Standard RFC 1382.

Table A-27 (Page 1 of 2). X.25 Packet Supported MIB Groups

Group Name	Supported	Not Supported
x25AdminTable		√
x25CallParmTable	√	
x25ChannelTable	√	
x25CircuitTable	√	
x25ClearedCircuitEntriesGranted	√	
x25ClearedCircuitEntriesRequested	√	

Table A-27 (Page 2 of 2). X.25 Packet Supported MIB Groups

Group Name	Supported	Not Supported
x25ClearedCircuitTable	√	
x25OperTable	√	
x25StatTable	√	

X.25 MIOX

Table A-28 lists the supported groups for the X.25 MIOX MIB.

Table A-28. X.25 MIOX Supported MIB Groups

Group Name	Supported	Not Supported
mioxPleTable	√	
mioxPeerTable	√	
mioxEncTable	√	

Enterprise MIBs

This section describes enterprise MIBs supported by IBM 2216.

APPN MIB

Table A-29 lists the IBM 2216-supported groups for the APPN MIB.

Table A-29. APPN Supported MIB Groups

Group Name	Supported	Not Supported
Node Group	√	
Topology Group	√	
Local Topology Group	√	
Directory group	√	
APPN Class of Service group	√	

BRS MIB

Table A-30 lists the supported groups for the BRS (Bandwidth Reservation System) MIB.

Note: Although the BRS function, in general, can be used for PPP and Frame Relay links, the BRS MIB applies only to PPP links.

Table A-30. IBM 2216 BRS MIB Groups

Group Name	Supported	Not Supported
BrsNumber	√	
BrsClassTable	√	
BrsFilterTable	√	
BrsPortTable	√	

DLUR MIB

Table A-31 lists the IBM 2216-supported groups for APPN Dependent LU Requester (DLUR). This MIB contains information that is useful for the management of an APPN product that implements DLUR (Dependent Logical Unit Requester). The DLUR product has a client/server relationship with an APPN product that implements DLUS (Dependent Logical Unit Server).

Table A-31. DLUR Supported MIB Groups

Group Name	Supported	Not Supported
dlurNodeCapabilities	√	
dlurPuInfo	√	
dlurDlusInfo	√	
dlurCorrelation		√

HPR MIB

Table A-32 lists the IBM 2216-supported groups for HPR.

Table A-32. HPR Supported MIB Groups

Group Name	Supported	Not Supported
ibmHprNceTable	√	
ibmHprRtpGlobe	√	
ibmHprRtpTable	√	
ibmHprRtpStatusTable	√	
ibmHprNclGlobe	√	
ibmHprNclTable	√	
ibmHprRtGlobe	√	
ibmHprRtGenTable	√	
ibmHprRtDetTable	√	

IBM 2216 MIBs

Table A-33 lists supported groups for the IBM 2216 product-specific MIB.

Table A-33. IBM 2216 MIB Supported Groups

Group Name	Supported	Not Supported
ibm2216admin	√	

IBM Common MIB

Table A-34 on page A-15 lists the IBM 2216-supported groups for the IBM Common MIB.

Table A-34. IBM Common Supported MIB Groups

Group Name	Supported	Not Supported
agentDebug	√	
platformType	√	
platformDRAMSize	√	
platformFLASHSize	√	
platformNVRAMSize	√	
platformFeatureSlot	√	
pppProtocolTable	√	
pppLinkErrTable	√	
pppLCProtoTable	√	
pppPAPTable	√	
pppCHAPTable	√	
ibmdlswTConnOperGroupTable	√	
ibmdlswQLLCLsTable	√	

Novell IPX MIB

Table A-35 lists the IBM 2216-supported groups for the Novell IPX MIB. Table A-36 lists exceptions.

Table A-35. Novell IPX Supported MIB Groups

Group Name	Supported	Not Supported
ipxBasicSysTable	√	
ipxAdvSysTable	√	
ipxCircTable	√	
ipxDestTable	√	
ipxServTable	√	
ipxDestServTable	√	
ipxStaticRouteTable		√
ipxStaticServTable		√

Table A-36. Exceptions for Novell IPX MIB

Group or Field Name	Exception
ipxCircName	Always null string
ipxCircDialName	Always null string

Novell RIP/SAP MIB

Table A-37 lists the Novell RIP/SAP MIB groups.

Table A-37. Novell RIP/SAP Supported MIB Groups

Group Name	Supported	Not Supported
ripCircTable	√	
ripSysTable	√	
sapCircTable	√	
sapSysTable	√	

Proteon Enterprise MIB

This MIB defines the system-wide network management function available on the 2216.

Table A-38. Proteon MIB Groups

Group Name	Supported	Not Supported
proConfig		√
proStatus		√
proSysObjId		√
proResource	√	
proTemp		√
proELS	√	
proPriv	√	

Remote Ping

Table A-39 lists the IBM 2216-supported groups for the Remote Ping MIB.

Table A-39. Remote Ping Supported MIB Groups

Group Name	Supported	Not Supported
nv6saIcmpEchoReq	√	

SNMP Generic Traps

This section describes the SNMP generic traps that are supported by the 2216.

Table A-40. SNMP Generic Traps

Trap Name	Purpose
cold-start	Generated when a device is powered on or reloaded.
warm-start	Generated when a device is restarted.
link-up	Generated when an interface is enabled
link-down	Generated when an interface is disabled
authentication-failed	Generated when the SNMP agent receives an SNMP request that does not authenticate.
egp-neighbor-loss	Not implemented.

Enterprise-specific Traps

The following enterprise-specific traps are supported:

Table A-41 (Page 1 of 2). Enterprise-Specific Traps

Trap Name	Purpose

ELS	Generated by the Event Logging System (ELS). Note that this trap will only be generated if the ELS subsystem has been configured to report ELS events via the SNMP trap. The textual description for the ELS event that generates the trap is contained in the trap payload.
	The trap contains a single variable binding containing a text string in one of the two following formats:
	If ELS timestamping is enabled:
	`hr:min:sec subsys_name.event_num: message_text`
	For example - 09:32:56 IP.008: no rte 9.7.1.8 -> 9.7.4.3 dsc
	If ELS timestamping is not enabled:
	`"subsys_name.event_num: message_text"`
	For example - IP.008: no rt3 9.7.1.8 -> 9.7.4.3 dsc

Transparent Bridge New Root	Generated when the bridge has become the new root of the spanning tree.
Transparent Bridge Topology Change	Generated by a bridge when any of its configured ports transitions from the Learning state to the Forwarding state, or from the Forwarding state to the Blocking state. The trap is not sent if a newRoot trap is sent for the same transition.
Frame Relay DLCI State Change	Generated when a frame relay virtual circuit state has changed. Either it has been created or invalidated or has toggled between the active and inactive states.
Low Router Memory Warning	The amount of free memory used by routing and bridging has dropped below a given percent of its original amount. The default percentage is 10%, but can be adjusted by the user.
SDLC Port Status Change	Generated when an SDLC port has transitioned to active or inactive.

Table A-41 (Page 2 of 2). Enterprise-Specific Traps

Trap Name	Purpose
SDLC Link Station Status Change	Generated when a link station has transitioned to contacted or discontacted.
X.25 Restart	Generated when the X.25 PLE has sent or received a restart packet.
X.25 Reset	Generated when the X.25 PLE has sent or received a reset packet.
IPX Circuit Up	Generated when an IPX circuit state transitions to the up state.
IPX Circuit Down	Generated when an IPX circuit state transitions to the down state.
BGP Backward Transition	Generated when a peer connection state has changed from the established state.
BGP Peer Connection Established	Generated when a connection to a BGP peer is established.

Settable Values

The IBM 2216 supports the following settable values:

Table A-42. Settable Values

MIB Group	Settable Values
MIB-II System Group (RFC 1213)	sysContact sysName sysLocation **Note:** These three set strings are restricted to 79 characters.
MIB-II SNMP Group (RFC 1213)	snmpEnableAuthenTraps
Interfaces MIB (RFC 1573)	ifAdminStatus ifLinkUpDownTrapEnable
Frame Relay MIB (RFC 1315)	frTrapState
Proteon MIB	proActionReset proActionReload proElsPin proElsTimeStamp proElsAction proElsSubSysDisplayDisable proElsSubSysDisplayEnable proElsSubSysTrapDisable proElsSubSysTrapEnable proElsSubSysEventLogToConsole proElsSubSysEventLogToTrap
BRS MIB	proBrsAlloc
APPN MIB	All Groups are supported.
HPR MIB	All Groups are supported.

Appendix B. Interoperating with the IBM 6611 Router

A number of configuration considerations must be addressed for IBM Nways Multiprotocol Access Services's DLSw implementation to interoperate with those of the IBM 6611 router.

The following sections provide an overview of these considerations, and indicate which features of IBM Nways Multiprotocol Access Services's DLSw implementation are not interoperable with those of the IBM 6611.

Note: The considerations cited here are derived from testing performed with the IBM 6611's MPNP V1.2 software. The considerations may not apply to other MPNP software versions.

The considerations have been categorized in the following sections:

- "Bridge Configuration Considerations"
- "DLSw-Related Considerations"
- "IP-Related Configuration Considerations" on page B-2
- "TCP-Related Considerations" on page B-2
- "Miscellaneous Interoperability Considerations" on page B-3

Bridge Configuration Considerations

The following are bridge configuration considerations:

- The LAN identification (Segment number) of the DLSw must match on both the IBM 2216 and IBM 6611 routers. If a mismatch persistently exists, enter the IBM Nways Multiprotocol Access Services Configurator (Task 6) and select the DLSw protocol. The **set srb** command can then be used to set a Segment Number value that matches the IBM 6611 equivalent.
- The maximum MTU value that can be used for the Bridge Frame is 2100 bytes. This is the largest value currently supported by the IBM 6611. If MTU values less than 2100 are specified, it is important that the configured values match on both the IBM 2216 and IBM 6611 routers.

DLSw-Related Considerations

DLSw-related interoperability considerations are as follows:

- The IBM Nways Multiprotocol Access Services DLSw implementation does not support generation of SSP_IAMOKAY message (SSP Message Type `X'x1D'`) while IBM 6611 DLSw implementation is supported. This SSP message is undocumented in RFC 1434, and is silently discarded by the IBM Nways Multiprotocol Access Services DLSw implementation upon receipt.
- The IBM 6611 DLSw implementation processes SSP_ENTER_BUSY/EXIT_BUSY messages received from the IBM Nways Multiprotocol Access Services DLSw implementation but will not generate similar flow control related SSP messages.
- The IBM Nways Multiprotocol Access Services DLSw implementation does support the user defined SSP_TEST_CIRCUIT_REQ message (SSP message type `X'x7A'`) that is generated by an IBM 6611 DLSw router functioning as an

APPN network node. Upon receipt of this message, the IBM Nways Multiprotocol Access Services DLSw implementation will return the user defined SSP_TEST_CIRCUIT_RSP message (SSP message type X'x7B'). This response is expected by the IBM 6611 DLSw router's APPN network node implementation.

IP-Related Configuration Considerations

The following are IP configuration considerations:

- The client/server and peer/peer DLSw group feature that enables IBM Nways Multiprotocol Access Services DLSw neighbors to dynamically find each other is not interoperable with the IBM 6611 DLSw implementation. As a result, the DLSw's **add tcp neighbor** configuration command must be used to define the static IP addresses of adjacent IBM 6611 DLSw peers.
- The preceding interoperability restriction on the IBM Nways Multiprotocol Access Services DLSw group feature has implications for the selection of RIP/OSPF:
  - To utilize DLSw groups on an 2216, you must also configure OSPF/MOSPF. But because these DLSw groups are not interoperable with the 6611, it is possible to configure the 2216 with only RIP enabled and no OSPF configuration.
  - Although OSPF and RIP can both be enabled on the IBM 2216 side, MOSPF (if selected through the OSPF configuration) is not supported by the IBM 6611.
- Within the IBM Nways Multiprotocol Access Services IP configuration, make sure that the fill patterns configured for broadcast addresses on a given interface match their equivalent definitions on the IBM 6611.
- IBM Nways Multiprotocol Access Services's Bandwidth Reservation System (BRS), which can be utilized to guarantee bandwidth for the transport of SNA traffic over DLSw, is not interoperable with the IBM 6611 DLSw implementation.

  Although the prioritization assigned by the IBM 2216 hardware for BRS can be implemented in an outbound direction, the prioritization order will not be guaranteed if intermediate IP routers do not support BRS. Also, because the 6611 does not support BRS on its end of the line, BRS can only be applicable in a single direction.

TCP-Related Considerations

The following are TCP interoperability considerations:

TCP Connection Break Detection Differences
The IBM Nways Multiprotocol Access Services DLSw implementation detects that a TCP connection is broken either when a Keepalive response is not received (assuming that the Keepalive option has been enabled for the connection) or when data cannot be delivered.

TCP Connection Reestablishment Differences
Once a TCP connection is broken, the IBM Nways Multiprotocol Access Services DLSw implementation reestablishes the TCP connection when a new DLSw SSP_CANUREACH is generated upon receipt of a DLC TEST

message from an end station. The IBM 6611 may not exhibit the same behavior.

Keepalive Disable/Enable Related Differences

As indicated previously, the IBM Nways Multiprotocol Access Services DLSw implementation permits the enabling and disabling of a Keepalive option when a TCP neighbor IP address is added (configured). Although TCP in the IBM 6611 DLSw implementation responds to Keepalive messages received on a TCP session, there is no mechanism to configure the resident 6611 TCP to enable the generation of TCP Keepalive messages.

Maximum Number of TCP Connections Supported

In the IBM Nways Multiprotocol Access Services DLSw implementation, there is no hard-coded restriction on the maximum number of TCP connections supported. As a result, the maximum number of TCP connections supported is directly related to a IBM 2216's available memory. In the IBM 6611 case, there is a hard-coded internal restriction of 100 TCP connections that can be supported in the DLSw implementation.

Miscellaneous Interoperability Considerations

Note the following miscellaneous interoperability considerations:

- If a problem is encountered when trying to establish a DLSw connection initiated by the IBM 6611, check the IBM 6611 configuration to ensure that MAC address filtering has not been inadvertently enabled for an associated source or destination MAC address.
- Although RFC 1434 does not specifically address the issue of orphan DLSw sessions (for example, DLSw sessions that remain in a DLSw circuit established state with no subsequent activity), both the IBM Nways Multiprotocol Access Services and IBM 6611 DLSw implementations resolve this issue by providing orphan DLSw session timeouts. DLSw sessions that remain inactive while in DLSw circuit established state for longer than 30 seconds are eliminated by both implementations.

Appendix C. Interoperating with the IBM 6611 Bridge

A number of configuration issues must be addressed before implementing bridging on the 2216 to interoperate with bridging on the IBM 6611.

This appendix provides an overview of these issues, and indicates which features of 2216's bridge implementation are **not** interoperable with the IBM 6611's bridge implementation.

Bridging Configuration Considerations

To help avoid building an incompatible network, the following bridge configuration issues should be considered when using the IBM 6611 and the IBM 2216 as the two end-bridges over PPP and frame-relay serial links.

For PPP, the IBM 2216 bridge supports different MAC types (Ethernet and token-ring) as described in RFC 1638, *PPP Bridging Control Protocol*. For frame-relay, the IBM 2216 supports RFC 1490 *Multiprotocol Interconnect over Frame Relay*.

Currently the IBM 6611 bridge supports Ethernet and token-ring MAC types over PPP and Frame Relay. However, the IBM 6611 bridge only supports token-ring MAC frames when the bridge port associated with PPP or frame-relay is configured as a **source-routing** port. This leads to certain restrictions in network topologies when the IBM 6611 and IBM 2216 are the two end-bridges over PPP or frame-relay.

RFC 1638, section 5.3, describes how a vendor can announce to the peer bridge the MAC type that is supported over PPP so that the peer does not send unsupported MAC type traffic over PPP. Currently, the IBM 2216 bridge does not drop non-Ethernet frames destined for the PPP network. Neither does it attempt to convert all the frames to Ethernet frames before sending them over PPP. This results in the IBM 6611 bridge receiving non-Ethernet frames over PPP and discarding them when there is a mismatch in the configuration.

Configuration Examples

the following examples of network topologies will ***not*** work. Possible alternate configurations are marked **Alt**. When considering WAN, LAN types can be extended to MAC types.

Example 1:
Token-Ring (SR) - IBM 2216 (SR-TB) - PPP (TB) - IBM 6611 (TB) - Ethernet

Alt: Token Ring (SR) - IBM 2216 (SRB) - PPP (SR) - IBM 6611 (SR-TB) - Ethernet

Example 2:
Token Ring (TB) - IBM 2216 (TB) - PPP (TB) - IBM 6611 (TB) - ETH/TKR

Alt: Token Ring (SR) - IBM 2216 (SRB) - PPP (SR) - IBM 6611 (SR-TB) - ETH

Alt: Token Ring (SR) - IBM 2216 (SRB) - PPP (SR) - IBM 6611 (SRB) - TKR

Alt: Token Ring (TB) - IBM 2216 (SR-TB) - PPP (SR) - IBM 6611 (SRB) - TKR

Alt: Token Ring (TB) - IBM 2216 (SR-TB) - PPP (SR) - 6611 (SR-TB) - ETH

The LAN frames generated by Boundary Access Node (BAN) and DLS are source-routed token-ring frames. Based on the media type and the bridge configuration behavior of the associated outgoing bridge port, the IBM 2216 bridge translates or converts the source-routed token-ring frame in the following manner.

1. ETH (TB) in Ethernet
2. PPP / FR / Tunnel / in Token-Ring TB format
3. PPP / FR / Tunnel / in Token-Ring SR format
4. TKR (TB) in Token-Ring TB format
5. TKR (SR) in Token-Ring SR format

List of Abbreviations

AARP	AppleTalk Address Resolution Protocol
ABR	area border router
ack	acknowledgement
AIX	Advanced Interactive Executive
AMA	arbitrary MAC addressing
AMP	active monitor present
ANSI	American National Standards Institute
AP2	AppleTalk Phase 2
APPN	Advanced Peer-to-Peer Networking
ARE	all-routes explorer
ARI/FCI	address recognized indicator/frame copied indicator
ARP	Address Resolution Protocol
AS	autonomous system
ASBR	autonomous system boundary router
ASCII	American National Standard Code for Information Interchange
ASN.1	abstract syntax notation 1
ASRT	adaptive source routing transparent
ASYNC	asynchronous
ATCP	AppleTalk Control Protocol
ATP	AppleTalk Transaction Protocol
AUI	attachment unit interface
ayt	are you there
BAN	Boundary Access Node
BECN	backward explicit congestion notification
BGP	Border Gateway Protocol
BNC	bayonet Niell-Concelman
BNCP	Bridging Network Control Protocol
BOOTP	BOOT protocol
BPDU	bridge protocol data unit
bps	bits per second
BR	bridging/routing
BRS	bandwidth reservation
BSD	Berkeley software distribution
BTP	BOOTP relay agent
BTU	basic transmission unit
CAM	content-addressable memory
CCITT	Consultative Committee on International Telegraph and Telephone
CD	collision detection
CGWCON	Gateway Console
CIDR	Classless Inter-Domain Routing
CIR	committed information rate
CLNP	Connectionless-Mode Network Protocol
CPU	central processing unit
CRC	cyclic redundancy check
CRS	configuration report server
CTS	clear to send
CUD	call user data
DAF	destination address filtering
DB	database
DBsum	database summary
DCD	data channel received line signal detector
DCE	data circuit-terminating equipment
DDLC	dual data-link controller
DDN	Defense Data Network
DDP	Datagram Delivery Protocol
DDT	Dynamic Debugging Tool
DHCP	Dynamic Host Configuration Protocol
dir	directly connected
DL	data link
DLC	data link control
DLCI	data link connection identifier
DLS	data link switching
DLSw	data link switching
DMA	direct memory access
DNA	Digital Network Architecture
DNCP	DECnet Protocol Control Protocol
DNIC	Data Network Identifier Code
DoD	Department of Defense
DOS	Disk Operating System
DR	designated router
DRAM	Dynamic Random Access Memory
DSAP	destination service access point
DSE	data switching equipment
DSE	data switching exchange

DSR	data set ready
DSU	data service unit
DTE	data terminal equipment
DTR	data terminal ready
Dtype	destination type
DVMRP	Distance Vector Multicast Routing Protocol
E1	2.048 Mbps transmission rate
EDEL	end delimiter
EDI	error detected indicator
EGP	Exterior Gateway Protocol
EIA	Electronics Industries Association
ELAP	EtherTalk Link Access Protocol
ELS	Event Logging System
EST	Eastern Standard Time
Eth	Ethernet
fa-ga	functional address-group address
FCS	frame check sequence
FECN	forward explicit congestion notification
FIFO	first in, first out
FLT	filter library
FR	Frame Relay
FRL	Frame Relay
FTP	File Transfer Protocol
GMT	Greenwich Mean Time
GOSIP	Government Open Systems Interconnection Profile
GTE	General Telephone Company
GWCON	Gateway Console
HDLC	high-level data link control
HEX	hexadecimal
HPR	high-performance routing
HST	TCP/IP host services
HTF	host table format
IBD	Integrated Boot Device
ICMP	Internet Control Message Protocol
ICP	Internet Control Protocol
ID	identification
IDP	Initial Domain Part
IDP	Internet Datagram Protocol
IEEE	Institute of Electrical and Electronics Engineers
Ifc#	interface number

IGP	interior gateway protocol
InARP	Inverse Address Resolution Protocol
IP	Internet Protocol
IPCP	IP Control Protocol
IPPN	IP Protocol Network
IPX	Internetwork Packet Exchange
IPXCP	IPX Control Protocol
ISDN	integrated services digital network
ISO	International Organization for Standardization
Kbps	kilobits per second
LAN	local area network
LAPB	link access protocol-balanced
LAT	local area transport
LCP	Link Control Protocol
LED	light-emitting diode
LF	largest frame; line feed
LLC	logical link control
LLC2	logical link control 2
LMI	local management interface
LRM	LAN reporting mechanism
LS	link state
LSA	link state advertisement
LSB	least significant bit
LSreq	link state request
LSrxl	link state retransmission list
LU	logical unit
MAC	medium access control
Mb	megabit
MB	megabyte
Mbps	megabits per second
MBps	megabytes per second
MC	multicast
MCF	MAC filtering
MIB	Management Information Base
MIB II	Management Information Base II
MILNET	military network
MOS	Micro Operating System
MOSDDT	Micro Operating System Dynamic Debugging Tool
MOSPF	Open Shortest Path First with multicast extensions
MSB	most significant bit

MSDU	MAC service data unit
MTU	maximum transmission unit
nak	not acknowledged
NBP	Name Binding Protocol
NBR	neighbor
NCP	Network Control Protocol
NCP	Network Core Protocol
NetBIOS	Network Basic Input/Output System
NIST	National Institute of Standards and Technology
NPDU	Network Protocol Data Unit
NRZ	non-return-to-zero
NRZI	non-return-to-zero inverted
NSAP	Network Service Access Point
NSF	National Science Foundation
NSFNET	National Science Foundation NETwork
NVCNFG	non-volatile configuration
OPCON	Operator Console
OSI	open systems interconnection
OSICP	OSI Control Protocol
OSPF	Open Shortest Path First
OUI	organization unique identifier
PC	personal computer
PDN	public data network
PING	Packet internet groper
PDU	protocol data unit
PID	process identification
P-P	Point-to-Point
PPP	Point-to-Point Protocol
PROM	programmable read-only memory
PU	physical unit
PVC	permanent virtual circuit
RAM	random access memory
RD	route descriptor
REM	ring error monitor
REV	receive
RFC	Request for Comments
RI	ring indicator; routing information
RIF	routing information field
RII	routing information indicator
RIP	Routing Information Protocol
RISC	reduced instruction-set computer
RNR	receive not ready
ROM	read-only memory
ROpcon	Remote Operator Console
RPS	ring parameter server
RTMP	Routing Table Maintenance Protocol
RTP	RouTing update Protocol
RTS	request to send
Rtype	route type
rxmits	retransmissions
rxmt	retransmit
SAF	source address filtering
SAP	service access point
SAP	Service Advertising Protocol
sdel	start delimiter
SDLC	SDLC relay, synchronous data link control
seqno	sequence number
SGMP	Simple Gateway Monitoring Protocol
SL	serial line
SMP	standby monitor present
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
SNAP	Subnetwork Access Protocol
SNMP	Simple Network Management Protocol
SNPA	subnetwork point of attachment
SPF	OSPF intra-area route
SPE1	OSPF external route type 1
SPE2	OSPF external route type 2
SPIA	OSPF inter-area route type
SPID	service profile ID
SPX	Sequenced Packet Exchange
SQE	signal quality error
SRAM	static random access memory
SRB	source routing bridge
SRF	specifically routed frame
SRLY	SDLC relay
SRT	source routing transparent
SR-TB	source routing-transparent bridge
STA	static
STB	spanning tree bridge
STE	spanning tree explorer

STP	shielded twisted pair; spanning tree protocol
SVC	switched virtual circuit
TB	transparent bridge
TCN	topology change notification
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TEI	terminal point identifier
TFTP	Trivial File Transfer Protocol
TKR	token ring
TMO	timeout
TOS	type of service
TSF	transparent spanning frames
TTL	time to live
TTY	teletypewriter
TX	transmit
UA	unnumbered acknowledgment
UDP	User Datagram Protocol
UI	unnumbered information
UTP	unshielded twisted pair
VINES	VIrtual NEtworking System
VIR	variable information rate
VL	virtual link
VR	virtual route
WAN	wide area network
WRS	WAN restoral/reroute
X.25	packet-switched networks
X.251	X.25 physical layer
X.252	X.25 frame layer
X.253	X.25 packet layer
XID	exchange identification
XNS	Xerox Network Systems
XSUM	checksum
ZIP	AppleTalk Zone Information Protocol
ZIP2	AppleTalk Zone Information Protocol 2
ZIT	Zone Information Table

Glossary

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- Internet Request for Comments: 1208, *Glossary of Networking Terms*.
- Internet Request for Comments: 1392, *Internet Users' Glossary*.
- The *Object-Oriented Interface Design: IBM Common User Access Guidelines*, Carmel, Indiana: Que, 1992.

The following cross-references are used in this glossary:

Contrast with: This refers to a term that has an opposed or substantively different meaning.

Synonym for: This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with: This is a backward reference from a defined term to all other terms that have the same meaning.

See: This refers the reader to multiple-word terms that have the same last word.

See also: This refers the reader to terms that have a related, but not synonymous, meaning.

A

AAL. ATM Adaptation Layer, the layer that adapts user data to/from the ATM network by adding/removing headers and segmenting/reassembling the data into/from cells.

AAL-5. ATM Adaptation Layer 5, one of several standard AALs. AAL-5 was designed for data communications, and is used by LAN Emulation and Classical IP.

abstract syntax. A data specification that includes all distinctions that are needed in data transmissions, but that omits (abstracts) other details such as those that depend on specific computer architectures. See also *abstract syntax notation 1 (ASN.1)* and *basic encoding rules (BER)*.

abstract syntax notation 1 (ASN.1). The Open Systems Interconnection (OSI) method for abstract syntax specified in the following standards:

- ITU-T Recommendation X.208 (1988) | ISO/IEC 8824: 1990
- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1: 1994

See also *basic encoding rules (BER)*.

ACCESS. In the Simple Network Management Protocol (SNMP), the clause in a Management Information Base (MIB) module that defines the minimum level of support that a managed node provides for an object.

acknowledgment. (1) The transmission, by a receiver, of acknowledge characters as an affirmative response to a sender. (T) (2) An indication that an item sent was received.

active monitor. In a token-ring network, a function performed at any one time by one ring station that initiates the transmission of tokens and provides token error recovery facilities. Any active adapter on the ring has the ability to provide the active monitor function if the current active monitor fails.

address. In data communication, the unique code assigned to each device, workstation, or user connected to a network.

address mapping table (AMT). A table, maintained within the AppleTalk router, that provides a current mapping of node addresses to hardware addresses.

address mask. For internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address. Synonymous with *subnet mask* and *subnetwork mask*.

address resolution. (1) A method for mapping network-layer addresses to media-specific addresses. (2) See also *Address Resolution Protocol (ARP)* and *AppleTalk Address Resolution Protocol (AARP)*.

Address Resolution Protocol (ARP). (1) In the Internet suite of protocols, the protocol that dynamically maps an IP address to an address used by a supporting metropolitan or local area network such as Ethernet or token-ring. (2) See also *Reverse Address Resolution Protocol (RARP)*.

addressing. In data communication, the way in which a station selects the station to which it is to send data.

adjacent nodes. Two nodes connected together by at least one path that connects no other node. (T)

Administrative Domain. A collection of hosts and routers, and the interconnecting networks, managed by a single administrative authority.

Advanced Peer-to-Peer Networking (APPN). An extension to SNA featuring (a) greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure; (b) dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection; (c) dynamic definition of network resources; and (d) automated resource registration and directory lookup. APPN extends the LU 6.2 peer orientation for end-user services to network control and supports multiple LU types, including LU 2, LU 3, and LU 6.2.

Advanced Peer-to-Peer Networking (APPN) end node. A node that provides a broad range of end-user services and supports sessions between its local control point (CP) and the CP in an adjacent network node. It uses these sessions to dynamically register its resources with the adjacent CP (its network node server), to send and receive directory search requests, and to obtain management services. An APPN end node can also attach to a subarea network as a peripheral node or to other end nodes.

Advanced Peer-to-Peer Networking (APPN) network. A collection of interconnected network nodes and their client end nodes.

Advanced Peer-to-Peer Networking (APPN) network node. A node that offers a broad range of end-user services and that can provide the following:

- Distributed directory services, including registration of its domain resources to a central directory server
- Topology database exchanges with other APPN network nodes, enabling network nodes throughout the network to select optimal routes for LU-LU sessions based on requested classes of service
- Session services for its local LUs and client end nodes
- Intermediate routing services within an APPN network

Advanced Peer-to-Peer Networking (APPN) node. An APPN network node or an APPN end node.

alert. A message sent to a management services focal point in a network to identify a problem or an impending problem.

all-stations address. In communications, synonym for *broadcast address*.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. (A)

analog. (1) Pertaining to data consisting of continuously variable physical quantities. (A)
(2) Contrast with *digital*.

AppleTalk. A network protocol developed by Apple Computer, Inc. This protocol is used to interconnect network devices, which can be a mixture of Apple and non-Apple products.

AppleTalk Address Resolution Protocol (AARP). In AppleTalk networks, a protocol that (a) translates AppleTalk node addresses into hardware addresses and (b) reconciles addressing discrepancies in networks that support more than one set of protocols.

AppleTalk Transaction Protocol (ATP). In AppleTalk networks, a protocol that provides client/server request and response functions for hosts accessing the Zone Information Protocol (ZIP) for zone information.

APPN network. See *Advanced Peer-to-Peer Networking (APPN) network*.

APPN network node. See *Advanced Peer-to-Peer Networking (APPN) network node*.

arbitrary MAC addressing (AMA). In DECnet architecture, an addressing scheme used by DECnet

Phase IV-Prime that supports universally administered addresses and locally administered addresses.

area. In Internet and DECnet routing protocols, a subset of a network or gateway grouped together by definition of the network administrator. Each area is self-contained; knowledge of an area's topology remains hidden from other areas.

asynchronous (ASYNC). Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T)

ATM. Asynchonous Transfer Mode, a connection-oriented, high-speed networking technology based on cell switching.

ATMARP. ARP in Classical IP.

attachment unit interface (AUI). In a local area network, the interface between the medium attachment unit and the data terminal equipment within a data station. (I) (A)

authentication failure. In the Simple Network Management Protocol (SNMP), a trap that may be generated by an authentication entity when a requesting client is not a member of the SNMP community.

autonomous system. In TCP/IP, a group of networks and routers under one administrative authority. These networks and routers cooperate closely to propagate network reachability (and routing) information among themselves using an interior gateway protocol of their choice.

autonomous system number. In TCP/IP, a number assigned to an autonomous system by the same central authority that also assigns IP addresses. The autonomous system number makes it possible for automated routing algorithms to distinguish autonomous systems.

B

backbone. (1) In a local area network multiple-bridge ring configuration, a high-speed link to which the rings are connected by means of bridges or routers. A backbone may be configured as a bus or as a ring. (2) In a wide area network, a high-speed link to which nodes or data switching exchanges (DSEs) are connected.

backbone network. A central network to which smaller networks, normally of lower speed, connect. The backbone network usually has a much higher capacity than the networks it helps interconnect or is a wide-area network (WAN) such as a public packet-switched datagram network.

backbone router. (1) A router used to transmit data between areas. (2) One in a series of routers that is used to interconnect networks into a larger internet.

Bandwidth. The bandwidth of an optical link designates the information-carrying capacity of the link and is related to the maximum bit rate that a fiber link can support.

basic transmission unit (BTU). In SNA, the unit of data and control information passed between path control components. A BTU can consist of one or more path information units (PIUs).

baud. In asynchronous transmission, the unit of modulation rate corresponding to one unit interval per second; that is, if the duration of the unit interval is 20 milliseconds, the modulation rate is 50 baud. (A)

bootstrap. (1) A sequence of instructions whose execution causes additional instructions to be loaded and executed until the complete computer program is in storage. (T) (2) A technique or device designed to bring itself into a desired state by means of its own action, for example, a machine routine whose first few instructions are sufficient to bring the rest of itself into the computer from an input device. (A)

Border Gateway Protocol (BGP). An Internet Protocol (IP) routing protocol used between domains and autonomous systems.

border router. In Internet communications, a router, positioned at the edge of an autonomous system, that communicates with a router that is positioned at the edge of a different autonomous system.

bridge. A functional unit that interconnects multiple LANs (locally or remotely) that use the same logical link control protocol but that can use different medium access control protocols. A bridge forwards a frame to another bridge based on the medium access control (MAC) address.

bridge identifier. An 8-byte field, used in a spanning tree protocol, composed of the MAC address of the port with the lowest port identifier and a user-defined value.

bridging. In LANs, the forwarding of a frame from one LAN segment to another. The destination is specified by the medium access control (MAC) sublayer address encoded in the destination address field of the frame header.

broadcast. (1) Transmission of the same data to all destinations. (T) (2) Simultaneous transmission of data to more than one destination. (3) Contrast with *multicast*.

broadcast address. In communications, a station address (eight 1's) reserved as an address common to all stations on a link. Synonymous with *all-stations address*.

C

cache. (1) A special-purpose buffer storage, smaller and faster than main storage, used to hold a copy of instructions and data obtained from main storage and likely to be needed next by the processor. (T) (2) A buffer storage that contains frequently accessed instructions and data; it is used to reduce access time. (3) An optional part of the directory database in network nodes where frequently used directory information may be stored to speed directory searches. (4) To place, hide, or store in a cache.

call request packet. (1) A call supervision packet that a data terminal equipment (DTE) transmits to ask that a connection for a call be established throughout the network. (2) In X.25 communications, a call supervision packet transmitted by a DTE to ask for a call establishment through the network.

canonical address. In LANs, the IEEE 802.1 format for the transmission of medium access control (MAC) addresses for token-ring and Ethernet adapters. In canonical format, the least significant (rightmost) bit of each address byte is transmitted first. Contrast with *noncanonical address*.

carrier. An electric or electromagnetic wave or pulse train that may be varied by a signal bearing information to be transmitted over a communication system. (T)

carrier detect. Synonym for *received line signal detector (RLSD)*.

carrier sense. In a local area network, an ongoing activity of a data station to detect whether another station is transmitting. (T)

carrier sense multiple access with collision detection (CSMA/CD). A protocol that requires carrier sense and in which a transmitting data station that detects another signal while transmitting, stops sending, sends a jam signal, and then waits for a variable time before trying again. (T) (A)

channel. (1) A path along which signals can be sent, for example, data channel, output channel. (A) (2) A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment.

channel service unit (CSU). A unit that provides the interface to a digital network. The CSU provides line conditioning (or equalization) functions, which keep the signal's performance consistent across the channel bandwidth; signal reshaping, which constitutes the binary pulse stream; and loopback testing, which includes the transmission of test signals between the CSU and the network carrier's office channel unit. See also *data service unit (DSU)*.

checksum. (1) The sum of a group of data associated with the group and used for checking purposes. (T) (2) In error detection, a function of all bits in a block. If the written and calculated sums do not agree, an error is indicated. (3) On a diskette, data written in a sector for error detection purposes; a calculated checksum that does not match the checksum of data written in the sector indicates a bad sector. The data are either numeric or other character strings regarded as numeric for the purpose of calculating the checksum.

circuit switching. (1) A process that, on demand, connects two or more data terminal equipment (DTEs) and permits the exclusive use of a data circuit between them until the connection is released. (I) (A) (2) Synonymous with *line switching*.

class A network. In Internet communications, a network in which the high-order (most significant) bit of the IP address is set to 0 and the host ID occupies the three low-order octets.

class B network. In Internet communications, a network in which the two high-order (most significant and next-to-most significant) bits of the IP address are set to 1 and 0, respectively, and the host ID occupies the two low-order octets.

class of service (COS). A set of characteristics (such as route security, transmission priority, and bandwidth) used to construct a route between session partners. The class of service is derived from a mode name specified by the initiator of a session.

client. (1) A functional unit that receives shared services from a server. (T) (2) A user.

client/server. In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

clocking. (1) In binary synchronous communication, the use of clock pulses to control synchronization of data and control characters. (2) A method of controlling the number of data bits sent on a telecommunication line in a given time.

collision. An unwanted condition that results from concurrent transmissions on a channel. (T)

collision detection. In carrier sense multiple access with collision detection (CSMA/CD), a signal indicating that two or more stations are transmitting simultaneously.

Committed information rate. The maximum amount of data in bits that the network agrees to deliver.

community. In the Simple Network Management Protocol (SNMP), an administrative relationship between entities.

community name. In the Simple Network Management Protocol (SNMP), a string of octets identifying a community.

compression. (1) The process of eliminating gaps, empty fields, redundancies, and unnecessary data to shorten the length of records or blocks. (2) Any encoding to reduce the number of bits used to represent a given message or record.

configuration. (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (T) (2) The devices and programs that make up a system, subsystem, or network.

configuration file. A file that specifies the characteristics of a system device or network.

configuration parameter. A variable in a configuration definition, the values of which can characterize the relationship of a product to other products in the same network or can define characteristics of the product itself.

configuration report server (CRS). In the IBM Token-Ring Network Bridge Program, the server that accepts commands from the LAN Network Manager (LNM) to get station information, set station parameters, and remove stations on its ring. This server also collects and forwards configuration reports generated by stations on its ring. The configuration reports include the new active monitor reports and the nearest active upstream neighbor (NAUN) reports.

congestion. See *network congestion*.

control point (CP). (1) A component of an APPN or LEN node that manages the resources of that node. In an APPN node, the CP is capable of engaging in CP-CP sessions with other APPN nodes. In an APPN network node, the CP also provides services to adjacent end nodes in the APPN network. (2) A component of a node that manages resources of that node and optionally provides services to other nodes in the network. Examples are a system services control point (SSCP) in a type 5 subarea node, a network node control point (NNCP) in an APPN network node, and an end node control point (ENCP) in an APPN or LEN end node. An SSCP and an NNCP can provide services to other nodes.

control point management services (CPMS). A component of a control point, consisting of management services function sets, that provides facilities to assist in performing problem management, performance and accounting management, change management, and configuration management. Capabilities provided by the CPMS include sending requests to physical unit management services (PUMS) to test system resources, collecting statistical information (for example, error and performance data) from PUMS about the system resources, and analyzing and presenting test results and statistical information collected about the system resources. Analysis and presentation responsibilities for problem determination and performance monitoring can be distributed among multiple CPMSs.

control point management services unit (CP-MSU). The message unit that contains management services data and flows between management services function sets. This message unit is in general data stream (GDS) format. See also *management services unit (MSU)* and *network management vector transport (NMVT)*.

D

D-bit. Delivery-confirmation bit. In X.25 communications, the bit in a data packet or call-request packet that is set to 1 if end-to-end acknowledgment (delivery confirmation) is required from the recipient.

daemon. A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their task; others operate periodically.

data carrier detect (DCD). Synonym for *received line signal detector (RLSD)*.

data circuit. (1) A pair of associated transmit and receive channels that provide a means of two-way data communication. (I) (2) In SNA, synonym for *link connection*. (3) See also *physical circuit* and *virtual circuit*.

Notes:

1. Between data switching exchanges, the data circuit may include data circuit-terminating equipment (DCE), depending on the type of interface used at the data switching exchange.
2. Between a data station and a data switching exchange or data concentrator, the data circuit includes the data circuit-terminating equipment at the data station end, and may include equipment similar to a DCE at the data switching exchange or data concentrator location.

data circuit-terminating equipment (DCE). In a data station, the equipment that provides the signal conversion and coding between the data terminal equipment (DTE) and the line. (I)

Notes:

1. The DCE may be separate equipment or an integral part of the DTE or of the intermediate equipment.
2. A DCE may perform other functions that are usually performed at the network end of the line.

data link connection identifier (DLCI). The numeric identifier of a frame-relay subport or PVC segment in a frame-relay network. Each subport in a single frame-relay port has a unique DLCI. The following table, excerpted from the American National Standards Institute (ANSI) Standard T1.618 and the International Telegraph and Telephone Consultative Committee (ITU-T/CCITT) Standard Q.922, indicates the functions associated with certain DLCI values:

DLCI Values	Function
0	in-channel signaling
1–15	reserved
16–991	assigned using frame-relay connection procedures
992–1007	layer 2 management of frame-relay bearer service
1008–1022	reserved
1023	in-channel layer management

data link control (DLC). A set of rules used by nodes on a data link (such as an SDLC link or a token ring) to accomplish an orderly exchange of information.

data link control (DLC) layer. In SNA, the layer that consists of the link stations that schedule data transfer over a link between two nodes and perform error control for the link. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

Note: The DLC layer is usually independent of the physical transport mechanism and ensures the integrity of data that reaches the higher layers.

data link layer. In the Open Systems Interconnection reference model, the layer that provides services to transfer data between entities in the network layer over a communication link. The data link layer detects and possibly corrects errors that may occur in the physical layer. (T)

data link level. (1) In the hierarchical structure of a data station, the conceptual level of control or processing logic between high level logic and the data link that maintains control of the data link. The data link level performs such functions as inserting transmit bits and deleting receive bits; interpreting address and control fields; generating, transmitting, and interpreting commands and responses; and computing and interpreting frame check sequences. See also *packet level* and *physical level*. (2) In X.25 communications, synonym for *frame level*.

data link switching (DLSw). A method of transporting network protocols that use IEEE 802.2 logical link control (LLC) type 2. SNA and NetBIOS are examples of protocols that use LLC type 2. See also *encapsulation* and *spoofing*.

data packet. In X.25 communications, a packet used for the transmission of user data on a virtual circuit at the DTE/DCE interface.

data service unit (DSU). A device that provides a digital data service interface directly to the data terminal equipment. The DSU provides loop equalization, remote and local testing capabilities, and a standard EIA/CCITT interface.

data set ready (DSR). Synonym for *DCE ready*.

data switching exchange (DSE). The equipment installed at a single location to provide switching functions, such as circuit switching, message switching, and packet switching. (I)

data terminal equipment (DTE). That part of a data station that serves as a data source, data sink, or both. (I) (A)

data terminal ready (DTR). A signal to the modem used with the EIA 232 protocol.

data transfer rate. The average number of bits, characters, or blocks per unit time passing between corresponding equipment in a data transmission system. (I)

Notes:

1. The rate is expressed in bits, characters, or blocks per second, minute, or hour.
2. Corresponding equipment should be indicated; for example, modems, intermediate equipment, or source and sink.

datagram. (1) In packet switching, a self-contained packet, independent of other packets, that carries information sufficient for routing from the originating data terminal equipment (DTE) to the destination DTE without relying on earlier exchanges between the DTEs and the network. (I) (2) In TCP/IP, the basic unit of information passed across the Internet environment. A datagram contains a source and destination address along with the data. An Internet Protocol (IP) datagram consists of an IP header followed by the transport layer data. (3) See also *packet* and *segment*.

Datagram Delivery Protocol (DDP). In AppleTalk networks, a protocol that provides network connectivity by means of connectionless socket-to-socket delivery service on the internet layer.

DCE ready. In the EIA 232 standard, a signal that indicates to the data terminal equipment (DTE) that the local data circuit-terminating equipment (DCE) is connected to the communication channel and is ready to send data. Synonymous with *data set ready (DSR)*.

DECnet. A network architecture that defines the operation of a family of software modules, databases, and hardware components typically used to tie Digital Equipment Corporation systems together for resource sharing, distributed computation, or remote system configuration. DECnet network implementations follow the Digital Network Architecture (DNA) model.

default. Pertaining to an attribute, condition, value, or option that is assumed when none is explicitly specified. (I)

designated router. A router that informs end nodes of the existence and identity of other routers. The selection of the designated router is based upon the router with the highest priority. When several routers share the highest priority, the router with the highest station address is selected.

destination node. The node to which a request or data is sent.

destination port. The 8-port asynchronous adapter that serves as a connection point with a serial service.

destination service access point (DSAP). In SNA and TCP/IP, a logical address that allows a system to route data from a remote device to the appropriate communications support. Contrast with *source service access point (SSAP)*.

device. A mechanical, electrical, or electronic contrivance with a specific purpose.

digital. (1) Pertaining to data that consist of digits. (T)
(2) Pertaining to data in the form of digits. (A)
(3) Contrast with *analog*.

Digital Network Architecture (DNA). The model for all DECnet hardware and software implementations.

direct memory access (DMA). The system facility that allows a device on the Micro Channel bus to get direct access to the system or bus memory without the intervention of the system processor.

directory. A table of identifiers and references to the corresponding items of data. (I) (A)

directory service (DS). An application service element that translates the symbolic names used by application processes into the complete network addresses used in an OSI environment. (T)

directory services (DS). A control point component of an APPN node that maintains knowledge of the location of network resources.

disable. To make nonfunctional.

disabled. (1) Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. (2) Pertaining to the state in which a transmission control unit or audio response unit cannot accept incoming calls on a line.

domain. (1) That part of a computer network in which the data processing resources are under common control. (T) (2) In Open Systems Interconnection (OSI), a part of a distributed system or a set of managed objects to which a common policy applies. (3) See *Administrative Domain* and *domain name*.

domain name. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of a host system is `ralvm7.vnet.ibm.com`, each of the following is a domain name:

- `ralvm7.vnet.ibm.com`
- `vnet.ibm.com`
- `ibm.com`

domain name server. In the Internet suite of protocols, a server program that supplies name-to-address translation by mapping domain names to IP addresses. Synonymous with *name server*.

Domain Name System (DNS). In the Internet suite of protocols, the distributed database system used to map domain names to IP addresses.

dotted decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

dump. (1) Data that has been dumped. (T) (2) To copy the contents of all or part of virtual storage for the purpose of collecting error information.

dynamic reconfiguration (DR). The process of changing the network configuration (peripheral PUs and LUs) without regenerating complete configuration tables or deactivating the affected major node.

Dynamic Routing. Routing using learned routes rather than routes statically configured at initialization.

E

echo. In data communication, a reflected signal on a communications channel. For example, on a communications terminal, each signal is displayed twice, once when entered at the local terminal and again when returned over the communications link. This allows the signals to be checked for accuracy.

EIA 232. In data communication, a specification of the Electronic Industries Association (EIA) that defines the interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE), using serial binary data interchange.

Electronic Industries Association (EIA). An organization of electronics manufacturers that advances the technological growth of the industry, represents the views of its members, and develops industry standards.

encapsulation. (1) In communications, a technique used by layered protocols by which a layer adds control information to the protocol data unit (PDU) from the layer it supports. In this respect, the layer encapsulates the data from the supported layer. In the Internet suite of protocols, for example, a packet would contain control information from the physical layer, followed by control information from the network layer, followed by the application protocol data. (2) See also *data link switching*.

encode. To convert data by the use of a code in such a manner that reconversion to the original form is possible. (T)

end node (EN). (1) See *Advanced Peer-to-Peer Networking (APPN) end node* and *low-entry networking (LEN) end node*. (2) In communications, a node that is frequently attached to a single data link and cannot perform intermediate routing functions.

entry point (EP). In SNA, a type 2.0, type 2.1, type 4, or type 5 node that provides distributed network management support. It sends network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal-point initiated commands to manage and control its resources.

Ethernet. A 10-Mbps baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and delayed retransmission. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

exception. An abnormal condition such as an I/O error encountered in processing a data set or a file.

exception response (ER). In SNA, a protocol requested in the form-of-response-requested field of a request header that directs the receiver to return a response only if the request is unacceptable as received or cannot be processed; that is, a negative response, but not a positive response, can be returned. Contrast with *definite response* and *no response*.

exchange identification (XID). A specific type of basic link unit that is used to convey node and link characteristics between adjacent nodes. XIDs are exchanged between link stations before and during link activation to establish and negotiate link and node characteristics, and after link activation to communicate changes in these characteristics.

explicit route (ER). In SNA, a series of one or more transmission groups that connect two subarea nodes. An explicit route is identified by an origin subarea address, a destination subarea address, an explicit route number, and a reverse explicit route number. Contrast with *virtual route (VR)*.

explorer frame. See *explorer packet*.

explorer packet. In LANs, a packet that is generated by the source host and that traverses the entire source routing part of a LAN, gathering information on the possible paths available to the host.

exterior gateway. In Internet communications, a gateway on one autonomous system that communicates with another autonomous system. Contrast with *interior gateway*.

Exterior Gateway Protocol (EGP). In the Internet suite of protocols, a protocol, used between domains and autonomous systems, that enables network reachability information to be advertised and exchanged. IP network addresses in one autonomous system are advertised to another autonomous system by means of EGP-participating routers. An example of an EGP is the Border Gateway Protocol (BGP). Contrast with Interior Gateway Protocol (IGP).

F

File Transfer Protocol (FTP). In the Internet suite of protocols, an application layer protocol that uses TCP and Telnet services to transfer bulk-data files between machines or hosts.

flow control. (1) In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. (2) See also *pacing*.

fragment. See *fragmentation*.

fragmentation. (1) The process of dividing a datagram into smaller parts, or fragments, to match the capabilities of the physical medium over which it is to be transmitted. (2) See also *segmenting*.

frame. (1) In Open Systems Interconnection architecture, a data structure pertaining to a particular area of knowledge and consisting of slots that can accept the values of specific attributes and from which inferences can be drawn by appropriate procedural attachments. (T) (2) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (3) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

frame level. Synonymous with *data link level*. See *link level*.

frame relay. (1) An interface standard describing the boundary between a user's equipment and a fast-packet network. In frame-relay systems, flawed frames are discarded; recovery comes end-to-end rather than hop-by-hop. (2) A technique derived from the integrated services digital network (ISDN) D channel standard. It assumes that connections are reliable and dispenses with the overhead of error detection and control within the network.

G

gateway. (1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T) (2) In the IBM Token-Ring Network, a device and its associated software that connect a local area network to another local area network or a host that uses different logical link protocols. (3) In TCP/IP, synonym for *router*.

general data stream (GDS). The data stream used for conversations in LU 6.2 sessions.

general data stream (GDS) variable. A type of RU substructure that is preceded by an identifier and a length field and includes either application data, user control data, or SNA-defined control data.

H

header. (1) System-defined control information that precedes user data. (2) The portion of a message that contains control information for the message such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

heap memory. The amount of RAM used to dynamically allocate data structures.

Hello. A protocol used by a group of cooperating, trusting routers to allow them to discover minimal delay routes.

hello message. (1) A message sent periodically to establish and test reachability between routers or between routers and hosts. (2) In the Internet suite of protocols, a message defined by the Hello protocol as an Interior Gateway Protocol (IGP).

heuristic. Pertaining to exploratory methods of problem solving in which solutions are discovered by evaluation of the progress made toward the final result.

high-level data link control (HDLC). In data communication, the use of a specified series of bits to control data links in accordance with the International Standards for HDLC: ISO 3309 Frame Structure and ISO 4335 Elements of Procedures.

high-performance routing (HPR). An addition to the Advanced Peer-to-Peer Networking (APPN) architecture that enhances data routing performance and reliability, especially when using high-speed links.

hop. (1) In APPN, a portion of a route that has no intermediate nodes. It consists of only a single transmission group connecting adjacent nodes. (2) To the routing layer, the logical distance between two nodes in a network.

hop count. (1) A metric or measure of distance between two points. (2) In Internet communications, the number of routers that a datagram passes through on its way to its destination. (3) In SNA, a measure of the number of links to be traversed in a path to a destination.

host. In the Internet suite of protocols, an end system. The end system can be any workstation; it does not have to be a mainframe.

hysteresis. The amount the temperature must change past the set alert threshold before the alert condition is cleared.

I

I-frame. Information frame.

information (I) frame. A frame in I format used for numbered information transfer.

input/output channel. In a data processing system, a functional unit that handles transfer of data between internal and peripheral equipment. (I) (A)

integrated services digital network (ISDN). A digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

Note: ISDNs are used in public and private network architectures.

interface. (1) A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions. (T) (2) Hardware, software, or both, that links systems, programs, or devices.

interior gateway. In Internet communications, a gateway that communicates only with its own autonomous system. Contrast with *exterior gateway*.

Interior Gateway Protocol (IGP). In the Internet suite of protocols, a protocol used to propagate network reachability and routing information within an autonomous system. Examples of IGPs are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

intermediate node. A node that is at the end of more than one branch. (T)

intermediate session routing (ISR). A type of routing function within an APPN network node that provides session-level flow control and outage reporting for all sessions that pass through the node but whose end points are elsewhere.

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

International Telecommunication Union (ITU). The specialized telecommunication agency of the United Nations, established to provide standardized communication procedures and practices, including frequency allocation and radio regulations worldwide.

internet. A collection of networks interconnected by a set of routers that allow them to function as a single, large network. See also *Internet*.

Internet. The internet administered by the Internet Architecture Board (IAB), consisting of large national backbone networks and many regional and campus networks all over the world. The Internet uses the Internet suite of protocols.

Internet address. See *IP address*.

Internet Architecture Board (IAB). The technical body that oversees the development of the Internet suite of protocols that are known as TCP/IP.

Internet Control Message Protocol (ICMP). The protocol used to handle errors and control messages in the Internet Protocol (IP) layer. Reports of problems and incorrect datagram destinations are returned to the original datagram source. ICMP is part of the Internet Protocol.

Internet Control Protocol (ICP). The VIrtual NEtworking System (VINES) protocol that provides exception notifications, metric notifications, and PING support. See also *RouTing update Protocol (RTP)*.

Internet Engineering Task Force (IETF). The task force of the Internet Architecture Board (IAB) that is responsible for solving the short-term engineering needs of the Internet.

Internet Protocol (IP). A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery and flow control and does not guarantee the reliability of the physical network.

Internetwork Packet Exchange (IPX). (1) The network protocol used to connect Novell's servers, or any workstation or router that implements IPX, with other workstations. Although similar to the Internet Protocol (IP), IPX uses different packet formats and terminology. (2) See also *Xerox Network Systems (XNS)*.

interoperability. The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units. (T)

intra-area routing. In Internet communications, the routing of data within an area.

Inverse Address Resolution Protocol (InARP). In the Internet suite of protocols, the protocol used for locating a protocol address through the known hardware address. In a frame-relay context, the data link connection identifier (DLCI) is synonymous with the known hardware address.

IP address. The 32-bit address defined by the Internet Protocol, standard 5, Request for Comments (RFC) 791. It is usually represented in dotted decimal notation.

IP datagram. In the Internet suite of protocols, the fundamental unit of information transmitted through an internet. It contains source and destination addresses, user data, and control information such as the length of the datagram, the header checksum, and flags indicating whether the datagram can be or has been fragmented.

IP router. A device in an IP internet that is responsible for making decisions about the paths over which network traffic will flow. Routing protocols are used to gain information about the network and to determine the best route over which the datagram should be forwarded toward the final destination. The datagrams are routed based on IP destination addresses.

IPXWAN. A Novell protocol that is used to exchange router-to-router information before exchanging standard Internetwork Packet Exchange (IPX) routing information and traffic over wide area networks (WANs).

L

LAN bridge server (LBS). In the IBM Token-Ring Network Bridge Program, the server that keeps statistical information about frames forwarded between two or more rings (through a bridge). The LBS sends these statistics to the appropriate LAN managers through the LAN reporting mechanism (LRM).

LAN Emulation (LE). An ATM Forum standard that supports legacy LAN applications over ATM networks.

LAN Emulation Client (LEC). A LAN Emulation component that represents users of the Emulated LAN.

LAN Emulation Configuration Server (LECS). A LAN Emulation Service component that centralizes and disseminates configuration data.

LAN Emulation Server (LES). A LAN Emulation Service component that resolves LAN Destinations to ATM Addresses.

LAN Network Manager (LNM). An IBM licensed program that enables a user to manage and monitor LAN resources from a central workstation.

LAN segment. (1) Any portion of a LAN (for example, a bus or ring) that can operate independently, but that is connected to other parts of the network by means of bridges. (2) A ring or bus network without bridges.

layer. (1) In network architecture, a group of services that is complete from a conceptual point of view, that is one out of a set of hierarchically arranged groups, and that extends across all systems that conform to the network architecture. (T) (2) In the Open Systems Interconnection reference model, one of seven conceptually complete, hierarchically arranged groups of services, functions, and protocols, that extend across all open systems. (T) (3) In SNA, a grouping of related functions that are logically separate from the functions in other groups. Implementation of the functions in one layer can be changed without affecting functions in other layers.

LE. LAN Emulation. An ATM Forum standard that supports legacy LAN applications over ATM networks.

LEC. LAN Emulation Client. A LAN Emulation component that represents users of the Emulated LAN.

LECS. LAN Emulation Configuration Server. A LAN Emulation Service component that centralizes and disseminates configuration data.

LES. LAN Emulation Server. A LAN Emulation Service component that resolves LAN Destinations to ATM Addresses.

line switching. Synonym for *circuit switching*.

link. The combination of the link connection (the transmission medium) and two link stations, one at each end of the link connection. A link connection can be shared among multiple links in a multipoint or token-ring configuration.

link access protocol balanced (LAPB). A protocol used for accessing an X.25 network at the link level. LAPB is a duplex, asynchronous, symmetric protocol, used in point-to-point communication.

link-attached. (1) Pertaining to devices that are connected to a controlling unit by a data link.
(2) Contrast with *channel-attached*. (3) Synonymous with *remote*.

link connection. (1) The physical equipment providing two-way communication between one link station and one or more other link stations; for example, a telecommunication line and data circuit-terminating equipment (DCE). (2) In SNA, synonymous with *data circuit*.

link level. (1) A part of Recommendation X.25 that defines the link protocol used to get data into and out of the network across the full-duplex link connecting the subscriber's machine to the network node. LAP and

LAPB are the link access protocols recommended by the CCITT. (2) See *data link level*.

link-state. In routing protocols, the advertised information about the usable interfaces and reachable neighbors of a router or network. The protocol's topological database is formed from the collected link-state advertisements.

link station. (1) The hardware and software components within a node representing a connection to an adjacent node over a specific link. For example, if node A is the primary end of a multipoint line that connects to three adjacent nodes, node A will have three link stations representing the connections to the adjacent nodes. (2) See also *adjacent link station (ALS)*.

local. (1) Pertaining to a device accessed directly without use of a telecommunication line. (2) Contrast with *remote*. (3) Synonym for *channel-attached*.

local area network (LAN). (1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T) (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network. (3) See also *Ethernet* and *token ring*. (4) Contrast with *metropolitan area network (MAN)* and *wide area network (WAN)*.

local bridging. A function of a bridge program that allows a single bridge to connect multiple LAN segments without using a telecommunication link. Contrast with *remote bridging*.

local management interface (LMI). See *local management interface (LMI) protocol*.

local management interface (LMI) protocol. In NCP, a set of frame-relay network management procedures and messages used by adjacent frame-relay nodes to exchange line status information over DLCI X'00'. NCP supports both the American National Standards Institute (ANSI) and International Telegraph and Telephone Consultative Committee (ITU-T/CCITT) versions of LMI protocol. These standards refer to LMI protocol as *link integrity verification tests (LIVT)*.

locally administered address. In a local area network, an adapter address that the user can assign to override the universally administered address. Contrast with *universally administered address*.

logical channel. In packet mode operation, a sending channel and a receiving channel that together are used to send and receive data over a data link at the same time. Several logical channels can be established on the same data link by interleaving the transmission of packets.

logical link. A pair of link stations, one in each of two adjacent nodes, and their underlying link connection, providing a single link-layer connection between the two nodes. Multiple logical links can be distinguished while they share the use of the same physical media connecting two nodes. Examples are 802.2 logical links used on local area network (LAN) facilities and LAP E logical links on the same point-to-point physical link between two nodes. The term logical link also includes the multiple X.25 logical channels that share the use of the access link from a DTE to an X.25 network.

logical link control (LLC). The data link control (DLC) LAN sublayer that provides two types of DLC operation for the orderly exchange of information. The first type is connectionless service, which allows information to be sent and received without establishing a link. The LLC sublayer does not perform error recovery or flow control for connectionless service. The second type is connection-oriented service, which requires establishing a link prior to the exchange of information. Connection-oriented service provides sequenced information transfer, flow control, and error recovery.

logical link control (LLC) protocol. In a local area network, the protocol that governs the exchange of transmission frames between data stations independently of how the transmission medium is shared. (T) The LLC protocol was developed by the IEEE 802 committee and is common to all LAN standards.

logical link control (LLC) protocol data unit. A unit of information exchanged between link stations in different nodes. The LLC protocol data unit contains a destination service access point (DSAP), a source service access point (SSAP), a control field, and user data.

logical unit (LU). A type of network accessible unit that enables users to gain access to network resources and communicate with each other.

loopback test. A test in which signals from a tester are looped at a modem or other network element back to the tester for measurements that determine or verify the quality of the communications path.

low-entry networking (LEN). A capability of nodes to attach directly to one another using basic peer-to-peer protocols to support multiple and parallel sessions between logical units.

low-entry networking (LEN) end node. A LEN node receiving network services from an adjacent APPN network node.

low-entry networking (LEN) node. A node that provides a range of end-user services, attaches directly to other nodes using peer protocols, and derives network services implicitly from an adjacent APPN network node, that is, without the direct use of CP-CP sessions.

M

Management Information Base (MIB). (1) A collection of objects that can be accessed by means of a network management protocol. (2) A definition for management information that specifies the information available from a host or gateway and the operations allowed. (3) In OSI, the conceptual repository of management information within an open system.

management station. In Internet communications, the system responsible for managing all, or a portion of, a network. The management station communicates with network management agents that reside in the managed node by means of a network management protocol, such as the Simple Network Management Protocol (SNMP).

mapping. The process of converting data that is transmitted in one format by the sender into the data format that can be accepted by the receiver.

mask. (1) A pattern of characters used to control retention or elimination of portions of another pattern of characters. (I) (A) (2) To use a pattern of characters to control retention or elimination of portions of another pattern of characters. (I) (A)

maximum transmission unit (MTU). In LANs, the largest possible unit of data that can be sent on a given physical medium in a single frame. For example, the MTU for Ethernet is 1500 bytes.

medium access control (MAC). In LANs, the sublayer of the data link control layer that supports medium-dependent functions and uses the services of the physical layer to provide services to the logical link control (LLC) sublayer. The MAC sublayer includes the method of determining when a device has access to the transmission medium.

medium access control (MAC) protocol. In a local area network, the protocol that governs access to the transmission medium, taking into account the topological aspects of the network, in order to enable the exchange of data between data stations. (T)

medium access control (MAC) sublayer. In a local area network, the part of the data link layer that applies a medium access method. The MAC sublayer supports topology-dependent functions and uses the services of the physical layer to provide services to the logical link control sublayer. (T)

metric. In Internet communications, a value, associated with a route, which is used to discriminate between multiple exit or entry points to the same autonomous system. The route with the lowest metric is preferred.

metropolitan area network (MAN). A network formed by the interconnection of two or more networks which may operate at higher speed than those networks, may cross administrative boundaries, and may use multiple access methods. (T) Contrast with *local area network (LAN)* and *wide area network (WAN)*.

MIB object. Synonym for *MIB variable*.

MIB variable. In the Simple Network Management Protocol (SNMP), a specific instance of data defined in a MIB module. Synonymous with *MIB object*.

MIB view. In the Simple Network Management Protocol (SNMP), the collection of managed objects, known to the agent, that is visible to a particular community.

MILNET. The military network that was originally part of ARPANET. It was partitioned from ARPANET in 1984. MILNET provides a reliable network service for military installations.

modem (modulator/demodulator). (1) A functional unit that modulates and demodulates signals. One of the functions of a modem is to enable digital data to be transmitted over analog transmission facilities. (T) (A) (2) A device that converts digital data from a computer to an analog signal that can be transmitted on a telecommunication line, and converts the analog signal received to data for the computer.

modulo. (1) Pertaining to a modulus; for example, 9 is equivalent to 4 modulo 5. (2) See also *modulus*.

modulus. A number, such as a positive integer, in a relationship that divides the difference between two related numbers without leaving a remainder; for example, 9 and 4 have a modulus of 5 (9 - 4 = 5; 4 - 9 = -5; and 5 divides both 5 and -5 without leaving a remainder).

monitor. (1) A device that observes and records selected activities within a data processing system for analysis. Possible uses are to indicate significant departure from the norm, or to determine levels of utilization of particular functional units. (T)
(2) Software or hardware that observes, supervises, controls, or verifies operations of a system. (A)
(3) The function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations.

multicast. (1) Transmission of the same data to a selected group of destinations. (T) (2) A special form of broadcast in which copies of a packet are delivered to only a subset of all possible destinations.

multiple-domain support (MDS). A technique for transporting management services data between management services function sets over LU-LU and CP-CP sessions. See also *multiple-domain support message unit (MDS-MU)*.

multiple-domain support message unit (MDS-MU). The message unit that contains management services data and flows between management services function sets over the LU-LU and CP-CP sessions used by multiple-domain support. This message unit, as well as the actual management services data that it contains, is in general data stream (GDS) format. See also *control point management services unit (CP-MSU)*, *management services unit (MSU)*, and *network management vector transport (NMVT)*.

N

Name Binding Protocol (NBP). In AppleTalk networks, a protocol that provides name translation function from the AppleTalk entity (resource) name (character string) into an AppleTalk IP address (16-bit number) on the transport layer.

name resolution. In Internet communications, the process of mapping a machine name to the corresponding Internet Protocol (IP) address. See also *Domain Name System (DNS)*.

name server. In the Internet suite of protocols, synonym for *domain name server*.

nearest active upstream neighbor (NAUN). In the IBM Token-Ring Network, the station sending data directly to a given station on the ring.

neighbor. A router on a common subnetwork that has been designated by a network administrator to receive routing information.

NetBIOS. Network Basic Input/Output System. A standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not need to handle the details of LAN data link control (DLC) protocols.

network. (1) A configuration of data processing devices and software connected for information interchange. (2) A group of nodes and the links interconnecting them.

network accessible unit (NAU). A logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with *network addressable unit*.

network address. According to ISO 7498-3, a name, unambiguous within the OSI environment, that identifies a set of network service access points.

network addressable unit (NAU). Synonym for *network accessible unit*.

network architecture. The logical structure and operating principles of a computer network. (T)

Note: The operating principles of a network include those of services, functions, and protocols.

network congestion. An undesirable overload condition caused by traffic in excess of what a network can handle.

network identifier. (1) In TCP/IP, that part of the IP address that defines a network. The length of the network ID depends on the type of network class (A, B, or C). (2) A 1- to 8-byte customer-selected name or an 8-byte IBM-registered name that uniquely identifies a specific subnetwork.

Network Information Center (NIC). In Internet communications, local, regional, and national groups throughout the world who provide assistance, documentation, training, and other services to users.

network layer. In Open Systems Interconnection (OSI) architecture, the layer that is responsible for routing, switching, and link-layer access across the OSI environment.

network management. The process of planning, organizing, and controlling a communication-oriented data processing or information system.

network management station. In the Simple Network Management Protocol (SNMP), a station that executes management application programs that monitor and control network elements.

network management vector transport (NMVT). A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

network manager. A program or group of programs that is used to monitor, manage, and diagnose the problems of a network.

network node (NN). See *Advanced Peer-to-Peer Networking (APPN) network node*.

network user address (NUA). In X.25 communications, the X.121 address containing up to 15 binary code digits.

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (I) (2) Any device, attached to a network, that transmits and receives data.

noncanonical address. In LANs, a format for the transmission of medium access control (MAC) addresses for token-ring adapters. In noncanonical format, the most significant (leftmost) bit of each address byte is transmitted first. Contrast with *canonical address*.

nonseed router. In AppleTalk networks, a router that acquires network number range and zone list information from a seed router attached to the same network.

O

Open Shortest Path First (OSPF). In the Internet suite of protocols, a function that provides intradomain information transfer. An alternative to the Routing Information Protocol (RIP), OSPF allows the lowest-cost routing and handles routing in large regional or corporate networks.

Open Systems Interconnection (OSI). (1) The interconnection of open systems in accordance with standards of the International Organization for Standardization (ISO) for the exchange of information. (T) (A) (2) The use of standardized procedures to enable the interconnection of data processing systems.

Note: OSI architecture establishes a framework for coordinating the development of current and future standards for the interconnection of computer systems. Network functions are divided into seven layers. Each layer represents a group of related data processing and communication functions that can be carried out in a standard way to support different applications.

Open Systems Interconnection (OSI) architecture. Network architecture that adheres to that particular set of ISO standards that relates to Open Systems Interconnection. (T)

Open Systems Interconnection (OSI) reference model. A model that describes the general principles of the Open Systems Interconnection, as well as the purpose and the hierarchical arrangement of its seven layers. (T)

origin. An external logical unit (LU) or application program from which a message or other data originates. See also *destination*.

orphan circuit. A non-configured circuit whose availability is learned dynamically.

P

pacing. (1) A technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. (2) See also *flow control*, *receive pacing*, *send pacing*, *session-level pacing*, and *virtual route (VR) pacing*.

packet. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)

packet internet groper (PING). (1) In Internet communications, a program used in TCP/IP networks to test the ability to reach destinations by sending the destinations an Internet Control Message Protocol (ICMP) echo request and waiting for a reply. (2) In communications, a test of reachability.

packet mode operation. Synonym for *packet switching*.

packet switching. (1) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during transmission of a packet. On completion of the transmission, the channel is made available for transfer of other packets. (I) (2) Synonymous with *packet mode operation*. See also *circuit switching*.

parallel bridges. A pair of bridges connected to the same LAN segment, creating redundant paths to the segment.

parallel transmission groups. Multiple transmission groups between adjacent nodes, with each group having a distinct transmission group number.

path. (1) In a network, any route between any two nodes. A path may include more than one branch. (T) (2) The series of transport network components (path control and data link control) that are traversed by the information exchanged between two network accessible units. See also *explicit route (ER)*, *route extension*, and *virtual route (VR)*.

path control (PC). The function that routes message units between network accessible units in the network and provides the paths between them. It converts the basic information units (BIUs) from transmission control (possibly segmenting them) into path information units

(PIUs) and exchanges basic transmission units containing one or more PIUs with data link control. Path control differs by node type: some nodes (APPN nodes, for example) use locally generated session identifiers for routing, and others (subarea nodes) use network addresses for routing.

path cost. In link-state routing protocols, the sum of the link costs along the path between two nodes or networks.

path information unit (PIU). A message unit consisting of a transmission header (TH) alone, or a TH followed by a basic information unit (BIU) or a BIU segment.

pattern-matching character. A special character such as an asterisk (*) or a question mark (?) that can be used to represent one or more characters. Any character or set of characters can replace a pattern-matching character. Synonymous with *global character* and *wildcard character*.

permanent virtual circuit (PVC). In X.25 and frame-relay communications, a virtual circuit that has a logical channel permanently assigned to it at each data terminal equipment (DTE). Call-establishment protocols are not required. Contrast with *switched virtual circuit (SVC)*.

physical circuit. A circuit established without multiplexing. See also *data circuit*. Contrast with *virtual circuit*.

physical layer. In the Open Systems Interconnection reference model, the layer that provides the mechanical, electrical, functional, and procedural means to establish, maintain, and release physical connections over the transmission medium. (T)

physical unit (PU). (1) The component that manages and monitors the resources (such as attached links and adjacent link stations) associated with a node, as requested by an SSCP via an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. This term applies to type 2.0, type 4, and type 5 nodes only. (2) See also *peripheral PU* and *subarea PU*.

ping command. The command that sends an Internet Control Message Protocol (ICMP) echo-request packet to a gateway, router, or host with the expectation of receiving a reply.

Point-to-Point Protocol (PPP). A protocol that provides a method for encapsulating and transmitting packets over serial point-to-point links.

polling. (1) On a multipoint connection or a point-to-point connection, the process whereby data stations are invited, one at a time, to transmit. (I) (2) Interrogation of devices for such purposes as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (A)

port. (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. (3) The representation of a physical connection to the link hardware. A port is sometimes referred to as an adapter; however, there can be more than one port on an adapter. There may be one or more ports controlled by a single DLC process. (4) In the Internet suite of protocols, a 16-bit number used to communicate between TCP or the User Datagram Protocol (UDP) and a higher-level protocol or application. Some protocols, such as File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP), use the same well-known port number in all TCP/IP implementations. (5) An abstraction used by transport protocols to distinguish among multiple destinations within a host machine. (6) Synonymous with *socket*.

port number. In Internet communications, the identification of an application entity to the transport service.

problem determination. The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environmental failure such as a power loss, or user error.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM in a current unaltered release of the program.

protocol. (1) A set of semantic and syntactic rules that determine the behavior of functional units in achieving communication. (I) (2) In Open Systems Interconnection architecture, a set of semantic and syntactic rules that determine the behavior of entities in the same layer in performing communication functions. (T) (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. Synonymous with *line control discipline* and *line discipline*. See *bracket protocol* and *link protocol*.

protocol data unit (PDU). A unit of data specified in a protocol of a given layer and consisting of protocol control information of this layer, and possibly user data of this layer. (T)

R

Rapid Transport Protocol (RTP) connection. In high-performance routing (HPR), the connection established between the endpoints of the route to transport session traffic.

reachability. The ability of a node or a resource to communicate with another node or resource.

read-only memory (ROM). Memory in which stored data cannot be modified by the user except under special conditions.

reassembly. In communications, the process of putting segmented packets back together after they have been received.

receive not ready (RNR). In communications, a data link command or response that indicates a temporary condition of being unable to accept incoming frames.

receive not ready (RNR) packet. See *RNR packet*.

received line signal detector (RLSD). In the EIA 232 standard, a signal that indicates to the data terminal equipment (DTE) that it is receiving a signal from the remote data circuit-terminating equipment (DCE). Synonymous with *carrier detect* and *data carrier detect (DCD)*.

Recognized Private Operating Agency (RPOA). Any individual, company, or corporation, other than a government department or service, that operates a telecommunication service and is subject to the obligations undertaken in the Convention of the International Telecommunication Union and in the Regulations; for example, a communication common carrier.

reduced instruction-set computer (RISC). A computer that uses a small, simplified set of frequently used instructions for rapid execution.

remote. (1) Pertaining to a system, program, or device that is accessed through a telecommunication line. (2) Synonym for *link-attached*. (3) Contrast with *local*.

remote bridging. The function of a bridge that allows two bridges to connect multiple LANs using a telecommunication link. Contrast with *local bridging*.

Remote Execution Protocol (REXEC). A protocol that allows the execution of a command or program on any host in the network. The local host receives the results of the command execution.

Request for Comments (RFC). In Internet communications, the document series that describes a part of the Internet suite of protocols and related experiments. All Internet standards are documented as RFCs.

reset. On a virtual circuit, reinitialization of data flow control. At reset, all data in transit are eliminated.

reset request packet. In X.25 communications, a packet transmitted by the data terminal equipment (DTE) to the data circuit-terminating equipment (DCE) to request that a virtual call or a permanent virtual circuit be reset. The reason for the request can also be specified in the packet.

ring. See *ring network*.

ring network. (1) A network in which every node has exactly two branches connected to it and in which there are exactly two paths between any two nodes. (T) (2) A network configuration in which devices are connected by unidirectional transmission links to form a closed path.

ring segment. A section of a ring that can be isolated (by unplugging connectors) from the rest of the ring. See *LAN segment*.

rlogin (remote login). A service, offered by Berkeley UNIX-based systems, that allows authorized users of one machine to connect to other UNIX systems across an internet and interact as if their terminals were connected directly. The rlogin software passes information about the user's environment (for example, terminal type) to the remote machine.

RNR packet. A packet used by a data terminal equipment (DTE) or by a data circuit-terminating equipment (DCE) to indicate a temporary inability to accept additional packets for a virtual call or permanent virtual circuit.

root bridge. The bridge that is the root of a spanning tree formed between other active bridges in the bridging network. The root bridge originates and transmits bridge protocol data units (BPDUs) to other active bridges to maintain the spanning tree topology. It is the bridge with the highest priority in the network.

route. (1) An ordered sequence of nodes and transmission groups (TGs) that represent a path from an origin node to a destination node traversed by the traffic exchanged between them. (2) The path that network traffic uses to get from source to destination.

route bridge. A function of an IBM bridge program that allows two bridge computers to use a telecommunication link to connect two LANs. Each bridge computer is connected directly to one of the LANs, and the telecommunication link connects the two bridge computers.

route extension (REX). In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *explicit route (ER)*, *path*, and *virtual route (VR)*.

Route Selection control vector (RSCV). A control vector that describes a route within an APPN network. The RSCV consists of an ordered sequence of control vectors that identify the TGs and nodes that make up the path from an origin node to a destination node.

router. (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached. (4) In TCP/IP, synonymous with *gateway*. (5) Contrast with *bridge*.

routing. (1) The assignment of the path by which a message is to reach its destination. (2) In SNA, the forwarding of a message unit along a particular path through a network, as determined by parameters carried in the message unit, such as the destination network address in a transmission header.

routing domain. In Internet communications, a group of intermediate systems that use a routing protocol so that the representation of the overall network is the same within each intermediate system. Routing domains are connected to each other by exterior links.

Routing Information Protocol (RIP). In the Internet suite of protocols, an interior gateway protocol used to exchange intradomain routing information and to determine optimum routes between internet hosts. RIP determines optimum routes on the basis of route metrics, not link transmission speed.

routing loop. A situation that occurs when routers circulate information among themselves until convergence occurs or until the networks involved are considered unreachable.

routing protocol. A technique used by a router to find other routers and to remain up to date about the best way to get to reachable networks.

routing table. A collection of routes used to direct datagram forwarding or to establish a connection. The information is passed among routers to identify network topology and destination feasibility.

Routing Table Maintenance Protocol (RTMP). In AppleTalk networks, a protocol that provides routing information generation and maintenance on the transport layer by means of the AppleTalk routing table. The AppleTalk routing table directs packet transmission through the internet from source socket to destination socket.

RouTing update Protocol (RTP). The VIrtual NEtworking System (VINES) protocol that maintains the routing database and allows the exchange of routing information between VINES nodes. See also *Internet Control Protocol (ICP)*.

rsh. A variant of the rlogin command that invokes a command interpreter on a remote UNIX machine and passes the command-line arguments to the command interpreter, skipping the login step completely.

S

seed router. In AppleTalk networks, a router that maintains configuration data (network range numbers and zone lists, for example) for the network. Each network must have at least one seed router. The seed router must be initially set up using the configurator tool. Contrast with *nonseed router*.

segment. (1) A section of cable between components or devices. A segment may consist of a single patch cable, several patch cables that are connected, or a combination of building cable and patch cables that are connected. (2) In Internet communications, the unit of transfer between TCP functions in different machines. Each segment contains control and data fields; the current byte-stream position and actual data bytes are identified along with a checksum to validate received data.

segmenting. In OSI, a function performed by a layer to map one protocol data unit (PDU) from the layer it supports into multiple PDUs.

sequence number. In communications, a number assigned to a particular frame or packet to control the transmission flow and receipt of data.

server. A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server. (T)

service access point (SAP). (1) In Open Systems Interconnection (OSI) architecture, the point at which the services of a layer are provided by an entity of that layer to an entity of the next higher layer. (T) (2) A logical point made available by an adapter where information can be received and transmitted. A single service access point can have many links terminating in it.

Service Advertising Protocol (SAP). In Internetwork Packet Exchange (IPX), a protocol that provides the following:

- A mechanism that allows IPX servers on an internet to advertise their services by name and type. Servers using this protocol have their name, service type, and address recorded in all file servers running NetWare.
- A mechanism that allows a workstation to broadcast a query to discover the identities of all servers of all types, all servers of a specific type, or the nearest server of a specific type.
- A mechanism that allows a workstation to query any file server running NetWare to discover the names and addresses of all servers of a specific type.

session. (1) In network architecture, for the purpose of data communication between functional units, all the activities which take place during the establishment, maintenance, and release of the connection. (T) (2) A logical connection between two network accessible units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) accompanying any transmissions exchanged during the session.

Simple Network Management Protocol (SNMP). In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application's Management Information Base (MIB).

SNA management services (SNA/MS). The services provided to assist in management of SNA networks.

socket. An endpoint for communication between processes or application programs.

source route bridging. In LANs, a bridging method that uses the routing information field in the IEEE 802.5 medium access control (MAC) header of a frame to determine which rings or token-ring segments the frame must transit. The routing information field is inserted into the MAC header by the source node. The information in the routing information field is derived from explorer packets generated by the source host.

source routing. In LANs, a method by which the sending station determines the route the frame will follow and includes the routing information with the frame. Bridges then read the routing information to determine whether they should forward the frame.

source service access point (SSAP). In SNA and TCP/IP, a logical address that allows a system to send data to a remote device from the appropriate communications support. Contrast with *destination service access point (DSAP).*

spanning tree. In LAN contexts, the method by which bridges automatically develop a routing table and update that table in response to changing topology to ensure that there is only one route between any two LANs in the bridged network. This method prevents packet looping, where a packet returns in a circuitous route back to the sending router.

sphere of control (SOC). The set of control point domains served by a single management services focal point.

sphere of control (SOC) node. A node directly in the sphere of control of a focal point. A SOC node has exchanged management services capabilities with its focal point. An APPN end node can be a SOC node if it supports the function to exchange management services capabilities.

split horizon. A technique for minimizing the time to achieve network convergence. A router records the interface over which it received a particular route and does not propagate its information about the route back over the same interface.

spoofing. For data links, a technique in which a protocol initiated from an end station is acknowledged and processed by an intermediate node on behalf of the final destination. In IBM 6611 data link switching, for example, SNA frames are encapsulated into TCP/IP packets for transport across a non-SNA wide area network, unpacked by another IBM 6611, and passed to the final destination. A benefit of spoofing is the prevention of end-to-end session timeouts.

standard MIB. In the Simple Network Management Protocol (SNMP), a MIB module that is located under the management branch of the Structure of Management Information (SMI) and that is considered a standard by the Internet Engineering Task Force (IETF).

static route. The route between hosts, networks, or both that is manually entered into a routing table.

station. An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line.

StreetTalk. In the VIrtual NEtworking System (VINES), a unique network-wide naming and addressing system that allows users to locate and access any resource on the network without knowing the network topology. See also *Internet Control Protocol (ICP)* and *RouTing update Protocol (RTP).*

Structure of Management Information (SMI). (1) In the Simple Network Management Protocol (SNMP), the rules used to define the objects that can be accessed by means of a network management protocol. (2) In OSI, the set of standards relating to management information. The set includes the *Management Information Model* and the *Guidelines for the Definition of Managed Objects*.

subarea. A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all network accessible units (NAUs), links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subnet. (1) In TCP/IP, a part of a network that is identified by a portion of the IP address. (2) Synonym for *subnetwork*.

subnet address. In Internet communications, an extension to the basic IP addressing scheme where a portion of the host address is interpreted as the local network address.

subnet mask. Synonym for *address mask*.

subnetwork. (1) Any group of nodes that have a set of common characteristics, such as the same network ID. (2) Synonymous with *subnet*.

Subnetwork Access Protocol (SNAP). In LANs, a 5-byte protocol discriminator that identifies the non-IEEE standard protocol family to which a packet belongs. The SNAP value is used to differentiate between protocols that use `$AA` as their service access point (SAP) value.

subnetwork mask. Synonym for *address mask*.

subsystem. A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system. (T)

switched virtual circuit (SVC). An X.25 circuit that is dynamically established when needed. The X.25 equivalent of a switched line. Contrast with *permanent virtual circuit (PVC)*.

synchronous. (1) Pertaining to two or more processes that depend upon the occurrence of specific events such as common timing signals. (T) (2) Occurring with a regular or predictable time relationship.

Synchronous Data Link Control (SDLC). (1) A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. (I) (2) Contrast with *binary synchronous communication (BSC)*.

SYNTAX. In the Simple Network Management Protocol (SNMP), a clause in the MIB module that defines the abstract data structure that corresponds to a managed object.

system configuration. A process that specifies the devices and programs that form a particular data processing system.

system services control point (SSCP). A component within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory services and other session services for users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks. The layered structure of SNA allows the ultimate origins and destinations of information, that is, the users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

T

Telnet. In the Internet suite of protocols, a protocol that provides remote terminal connection service. It allows users of one host to log on to a remote host and interact as directly attached terminal users of that host.

threshold. (1) In IBM bridge programs, a value set for the maximum number of frames that are not forwarded across a bridge due to errors, before a “threshold exceeded” occurrence is counted and indicated to network management programs. (2) An initial value from which a counter is decremented to 0, or a value to which a counter is incremented or decremented from an initial value.

throughput class. In packet switching, the speed at which data terminal equipment (DTE) packets travel through the packet switching network.

time to live (TTL). A technique used by best-effort delivery protocols to inhibit endlessly looping packets. The packet is discarded if the TTL counter reaches 0.

timeout. (1) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event. (I) (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

token. (1) In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. Each data station has an opportunity to acquire and use the token to control the medium. A token is a particular message or bit pattern that signifies permission to transmit. (T) (2) In LANs, a sequence of bits passed from one device to another along the transmission medium. When the token has data appended to it, it becomes a frame.

token ring. (1) According to IEEE 802.5, network technology that controls media access by passing a token (special packet or frame) between media-attached stations. (2) A FDDI or IEEE 802.5 network with a ring topology that passes tokens from one attaching ring station (node) to another. (3) See also *local area network (LAN)*.

token-ring network. (1) A ring network that allows unidirectional data transmission between data stations, by a token passing procedure, such that the transmitted data return to the transmitting station. (T) (2) A network that uses a ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission.

topology. In communications, the physical or logical arrangement of nodes in a network, especially the relationships among nodes and the links between them.

topology database update (TDU). A message about a new or changed link or node that is broadcast among APPN network nodes to maintain the network topology database, which is fully replicated in each network node. A TDU contains information that identifies the following:

- The sending node
- The node and link characteristics of various resources in the network
- The sequence number of the most recent update for each of the resources described.

trace. (1) A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A) (2) For data links, a record of the frames and bytes transmitted or received.

transceiver (transmitter-receiver). In LANs, a physical device that connects a host interface to a local area network, such as Ethernet. Ethernet transceivers contain electronics that apply signals to the cable and that sense collisions.

Transmission Control Protocol (TCP). A communications protocol used in the Internet and in any network that follows the U.S. Department of Defense standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet Protocol (IP) as the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transmission group (TG). (1) A connection between adjacent nodes that is identified by a transmission group number. (2) In a subarea network, a single link or a group of links between adjacent nodes. When a transmission group consists of a group of links, the links are viewed as a single logical link, and the transmission group is called a *multilink transmission group (MLTG)*. A *mixed-media multilink transmission group (MMMLTG)* is one that contains links of different medium types (for example, token-ring, switched SDLC, nonswitched SDLC, and frame-relay links). (3) In an APPN network, a single link between adjacent nodes. (4) See also *parallel transmission groups*.

transmission header (TH). Control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit*.

transparent bridging. In LANs, a method for tying individual local area networks together through the medium access control (MAC) level. A transparent bridge stores the tables that contain MAC addresses so that frames seen by the bridge can be forwarded to another LAN if the tables indicate to do so.

transport layer. In the Open Systems Interconnection reference model, the layer that provides a reliable end-to-end data transfer service. There may be relay open systems in the path. (T) See also *Open Systems Interconnection reference model*.

trap. In the Simple Network Management Protocol (SNMP), a message sent by a managed node (agent function) to a management station to report an exception condition.

tunneling. To treat a transport network as though it were a single communication link or LAN. See also *encapsulation*.

T1. In the United States, a 1.544-Mbps public access line. It is available in twenty-four 64-Kbps channels. The European version (E1) transmits 2.048 Mbps.

U

universally administered address. In a local area network, the address permanently encoded in an adapter at the time of manufacture. All universally administered addresses are unique. Contrast with *locally administered address*.

User Datagram Protocol (UDP). In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

V

V.24. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE).

V.25. In data communication, a specification of the CCITT that defines the automatic answering equipment and parallel automatic calling equipment on the General Switched Telephone Network, including procedures for disabling of echo controlled devices for both manually and automatically established calls.

V.35. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) at various data rates.

V.36. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) at rates of 48, 56, 64, or 72 kilobits per second.

VINES. VIrtual NEtworking System.

virtual circuit. (1) In packet switching, the facilities provided by a network that give the appearance to the user of an actual connection. (T) See also *data circuit*. Contrast with *physical circuit*. (2) A logical connection established between two DTEs.

virtual link. In Open Shortest Path First (OSPF), a point-to-point interface that connects border routers that are separated by a non-backbone transit area. Because area routers are part of the OSPF backbone, the virtual link connects the backbone. The virtual links ensure that the OSPF backbone does not become discontinuous.

VIrtual NEtworking System (VINES). The network operating system and network software from Banyan Systems, Inc. In a VINES network, virtual linking allows all devices and services to appear to be directly connected to each other, when they may actually be thousands of miles apart. See also *StreetTalk*.

virtual route (VR). (1) In SNA, either (a) a logical connection between two subarea nodes that is physically realized as a particular explicit route or (b) a logical connection that is contained wholly within a subarea node for intranode sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual route pacing, and provides data integrity through sequence numbering of path information units (PIUs). (2) Contrast with *explicit route (ER)*. See also *path* and *route extension (REX).*

W

wide area network (WAN). (1) A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities. (T) (2) A data communication network designed to serve an area of hundreds or thousands of miles; for example, public and private packet-switching networks, and national telephone networks. (3) Contrast with *local area network (LAN)* and *metropolitan area network (MAN).*

wildcard character. Synonym for *pattern-matching character*.

X

X.21. An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for a general-purpose interface between data terminal equipment and data circuit-terminating equipment for synchronous operations on a public data network.

X.25. (1) An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. (2) See also *packet switching*.

Xerox Network Systems (XNS). The suite of internet protocols developed by the Xerox Corporation.

Although similar to TCP/IP protocols, XNS uses different packet formats and terminology. See also *Internetwork Packet Exchange (IPX)*.

Z

zone. In AppleTalk networks, a subset of nodes within an internet.

Zone Information Protocol (ZIP). In AppleTalk networks, a protocol that provides zone management service by maintaining a mapping of the zone names and network numbers across the internet on the session layer.

zone information table (ZIT). A listing of network numbers and their associated zone name mappings in the internet. This listing is maintained by each internet router in an AppleTalk internet.

Index

Special Characters

?(Help)
- add 28-2
- ARP configuration command 20-11
- ARP console command 21-2
- ARP over ATM console command 21-7
- ASRT Bridge configuration command 7-3
- ASRT Bridge console command 8-2
- BAN console command 8-19
- CIP console command 21-7
- DLSw configuration command 27-38
- IP configuration command 16-16
- IP console command 17-2
- IPX configuration command 18-16
- IPX console command 19-2
- IPX filter configuration command 18-31
- IPX over ATM console command 21-7
- NetBIOS Filtering configuration command 10-2
- NetBIOS Filtering console command 11-1
- OSPF configuration command 23-17
- OSPF console command 24-2
- See help 19-2
- SNMP configuration command 25-3
- SNMP console command 26-2
- TCP/IP Host Services configuration command 13-2
- TCP/IP Host Services console command 14-2
- Tunnel configuration command 7-35

Numerics

8209 bridges 3-7

A

access control
- IP filtering 16-7

Access controls
- IP console command 17-2
- IPX console command 19-2

Adaptive Source Routing Transparent Bridge (ASRT)
- basic configuration procedures 5-1
- bit ordering in STB and SRB bridges 2-29
- bridge-only management 3-1, 3-3
- bridging basics 1-1
- bridging tunnel 3-1
  - encapsulation and OSPF 3-2
- configuration matrix 2-30
- configuring 2-29, 5-1, 7-1
- description of 2-20
- eliminating packet size problems 2-28
- Ethernet packet format translation 2-5
- hardware address filtering 2-28

Adaptive Source Routing Transparent Bridge (ASRT) *(continued)*
- MIB support 3-1, 3-3
- monitoring 8-1
- overview 1-1
  - complex bridges 1-4
  - CSMA/CD MAC frames 1-7
  - local bridges 1-4
  - MAC bridge frame formats 1-1, 1-7
  - operation and protocol architecture 1-5
  - point-to-point links 1-5
  - remote bridges 1-4
  - simple bridge 1-3, 1-5
  - token-ring MAC frames 1-8
- protocol filtering 2-15
- source routing bridge (SRB) 2-10
  - operation 2-11
  - source routing frames 2-11
  - spanning tree explore option 2-14
- spanning tree bridges 2-5
- spanning tree explore option
  - balancing traffic loads 2-14
  - simulating a network 2-14
- SR-TB bridging 2-24
- SR-TB conversion
  - description of 2-20
  - operation 2-21, 2-22
- SRB terminology and concepts
  - bridge instance 2-16
  - bridge number 2-16
  - explorer frames 2-16
  - interface number 2-16
  - overview 2-16
  - route 2-17
  - route discovery 2-17
  - segment number 2-17
  - source routing 2-17
- TCP/IP host services 3-1, 3-3
- terminology and concepts
  - aging time 2-6
  - all routes broadcast 2-27
  - all stations broadcast 2-27
  - bridge 2-7, 2-27
  - bridge address 2-7
  - bridge hello time 2-7
  - bridge identifier 2-7
  - bridge maximum age 2-7
  - bridge number 2-27
  - bridge priority 2-8
  - designated port 2-8
  - destination bridge 2-8
  - explorer frames 2-27
  - filtering and permanent databases 2-8

Adaptive Source Routing Transparent Bridge (ASRT) *(continued)*
terminology and concepts *(continued)*
parallel bridges 2-9
path cost 2-9
port 2-9
port ID 2-9
port number 2-9
port priority 2-9
resolution 2-9
ring number 2-27
root bridge 2-9
root port 2-10
route 2-27
route designator 2-27
route discovery 2-27
segment number 2-28
single route broadcasting 2-28
source routing bridging 2-28
spanning tree 2-10, 2-28
transparent bridging 2-28
transparent bridge (STB)
network requirements 2-2
operation of 2-2
overview 2-1
routers and transparent bridges 2-2
shaping the spanning tree 2-3
transparent-source routing compatibility 2-28
Adaptive Source Routing Transparent Bridge (ASRTB)
SR-TB conversion
general description 2-21
Add
ARP over ATM configuration command 20-17
ASRT Bridge configuration command 7-3
ASRT Bridge console command 8-2
CIP configuration command 20-17
DLSw configuration command 27-38
IP configuration command 16-17
IPX configuration command 18-16
IPX over ATM configuration command 20-17
OSPF configuration command 23-18
SNMP configuration command 25-3
SNMP console command 26-2
TCP/IP Host Services configuration command 13-2
Tunnel configuration command 7-36
XTP configuration command 22-13
XTP monitoring command 22-19
Add Entry
ARP configuration command 20-12
Address entries
dynamic 7-27, 8-4, 8-10
free 7-28, 8-4
permanent 7-27, 8-4, 8-10
registered 7-27, 8-4, 8-10
reserved 7-27
static 7-27
address wildcards, DTE 22-3
addresses and components of Classical IP 15-3
addresses, entering
CIP 20-8
Advertisement Expansion
OSPF console command 24-2
AppleTalk
split-horizon routing 18-13
APPN
interface support 27-15
Area Summary
OSPF console command 24-5
ARP
configuring 20-1
displaying statistics 21-4
exiting to CONFIG prompt 20-15, 21-5
monitoring 21-1
translation cache 20-2
with AppleTalk threading 3-9
with IP threading 3-8
ARP configuration commands
?(Help) 20-11
add entry 20-12
change entry 20-12
delete entry 20-13
disable auto-refresh 20-13
enable auto-refresh 20-13
exit 20-15
list 20-14
set 20-15
summary of 20-11
ARP console commands
?(Help) 21-2
accessing 21-1
clear 21-2
dump 21-2
exit 21-5
hardware 21-3
protocol 21-4
statistics 21-4
summary of 21-2
ARP over ATM
? (Help) 20-16
add 20-17
Classical IP, description 20-4
configuration commands, summary 20-16
effect on ARP table 20-16
IPX and ARP over ATM, description 20-10
ARP over ATM configuration command
add 20-23
delete 20-25
exit 20-29
list 20-27
ARP over ATM configuration commands
accessing 20-10

ARP over ATM console commands
  ?(Help) 21-7
  delete 21-7
  display 21-7
  dump 21-8
  hardware 21-9
  ping 21-9
  protocol 21-9
  statistics 21-10
  summary of 21-6
ARP table
  CIP 20-16
  IPX over ATM 20-16
AS boundary routing, OSPF 23-13
AS-External Advertisements
  OSPF console command 24-6
ASRT
  See Adaptive Source Routing Transparent Bridge 1-1, 2-1, 3-1
ASRT Bridge configuration commands
  ?(Help) 7-3
  add 7-3
  and IP tunnel 7-34
  ASRT Bridge configuration command 7-28
  ban 7-12
  change 7-12
  delete 7-13
  disable 7-15
  enable 7-18
  exit 7-34
  functional address to group address mapping 7-7
  IP tunnel commands 7-34
  list 7-21
  NetBIOS filtering commands
    summary 10-1
  NetBIOS filtering concepts 3-1, 3-4
  NetBIOS filtering configuration commands
    ?(Help) 10-2
    create 10-2
    delete 10-2
    disable 10-3
    enable 10-3
    exit 10-10
    filter-on 10-3
    list 10-5
    update 10-5
  port maps explained 7-5
  set 7-28
  summary of 7-1
  tunnel 7-34
  tunnel configuration commands
    ?(Help) 7-35
    add 7-36
    delete 7-36
    exit 7-38
    join 7-36
    list 7-38
ASRT Bridge console commands
  ?(Help) 8-2
  add 8-2
  ban 8-3
  BAN console commands
    ?(Help) 8-19
    discussion 8-19
    list 8-20
  cache 8-3
  delete 8-4
  exit 8-18
  flip 8-4
  list 8-5
  NetBIOS 8-18
  NetBIOS filtering console commands
    ?(Help) 11-1
    list 11-2
    summary 11-1
  summary 8-1
ASRT Bridge NetBIOS feature
  prompt 7-1, 8-1
ASRT Bridge NetBIOS-Filtering feature
  prompt 7-1, 8-1
ASRT Bridge tunnel feature
  prompt 7-1
ASRT bridging 6-2
ASRT configuration commands
  list
    filtering 7-24
  netbios 7-28
ATM addresses
  CIP 20-7
ATM addresses of Classical IP components 15-3
ATM addressing in IPX
  ESI 6-4
  selector 6-4
Attach
  IPX filter configuration command 18-31
Auto-refresh
  disabling 20-13
  enabling 20-13

B

backup peer function, XTP 22-4
balancing SNA and NetBIOS traffic 27-17
BAN
  ASRT Bridge configuration command 7-12
  ASRT Bridge console command 8-3
  opening service access points 4-7
BAN configuration command
BAN Console Commands
  accessing 8-19
  discussion 8-19
  Exit 8-20
  List 8-20

BAN filtering console commands
?(Help) 8-19
ban monitoring 28-3
BOOTP
enabling/disabling 16-11
server 16-12
Bootstrap monitor
forwarding process 16-10
Bootstrap protocol 16-10
Boundary Access Node (BAN)
configuring 4-1
using 4-1
Boundary routing, OSPF 23-13
Bridge
MAC frame formats 1-1, 1-7
point-to-point links 1-5
bridge and router 2-2
Bridges
basic operation 1-5
overview 1-1
types 1-3
versus routers 1-2
bridging 6-1
See also routing and bridging
bridging behaviors 6-2
bridging features 3-1
bridging overview 6-1, 6-2
Bridging tunnel
description of 3-1
encapsulation and OSPF 3-2

C

Cache
ASRT Bridge console command 8-3
IP console command 17-3
IPX console command 19-3
TCP/IP Host Services console command 14-3
Change
ARP over ATM configuration command 20-23
ASRT Bridge configuration command 7-12
CIP configuration command 20-23
IP configuration command 16-22
IPX over ATM configuration command 20-23
XTP configuration command 22-15
Change Entry
ARP configuration command 20-12
CIP
See also Classical IP over ATM
ATM addresses 20-7
components 20-5
configuration commands, summary 20-16
configuring 20-1
description 20-4
differences from IPX over ATM 20-16
effect on ARP table 20-16
CIP *(continued)*
how to enter addresses 20-8
IP addresses 20-6
key configuration parameters 20-8
logical IP subnets (LIS) 20-4
refresh 20-6
timeout 20-6
Virtual Channel Connection (VCC) 20-7
CIP configuration commands
? (Help) 20-16
accessing 20-10
add 20-17
change 20-23
delete 20-25
Exit 20-29
list 20-27
CIP console commands
?(Help) 21-7
delete 21-7
display 21-7
dump 21-8
hardware 21-9
ping 21-9
protocol 21-9
statistics 21-10
summary of 21-6
Classical IP ARP table, timeouts and refresh 15-2
Classical IP and ARP over ATM
See also Classical IP over ATM
description 20-4
Classical IP ATM addresses 15-3
Classical IP client 15-3
Classical IP over ATM
ATM addresses 15-3
components 15-2
IP addresses 15-3
key parameters 15-4
overview 15-1
Quality of Service (QOS) 15-4
timeouts and refresh of the ARP table 15-2
VCCs 15-3
Clear
ARP console command 21-2
IPX interface-based filter command 19-14
Close SAP
DLSw configuration command 27-43
Command summary
LNM 12-5
components of Classical IP over ATM 15-2
Config command 19-4
Configuration commands
DLSw 9-20
LNM 12-5
NetBIOS 9-20
XTP 22-13

Configuration environment
  accessing 9-21
Configuration parameters
  setting for ARP 20-15
connection request timer 22-4
console commands
  ARP over ATM 21-6
  CIP 21-6
  IPX over ATM 21-6
Counters
  IP console command 17-4
  IPX console command 19-5
Create
  IPX filter configuration command 18-32
Create command 10-2

D

Database
  permanent 8-4, 8-10
Database Summary
  OSPF console command 24-7
Default
  IPX filter configuration command 18-32
Delete
  ARP over ATM configuration command 20-25
  ARP over ATM console command 21-7
  ASRT Bridge configuration command 7-13
  ASRT Bridge console command 8-4
  CIP configuration command 20-25
  CIP console command 21-7
  DLSw configuration command 27-43
  IP configuration command 16-24
  IPX configuration command 18-19, 19-6
  IPX filter configuration command 18-32
  IPX over ATM configuration command 20-25
  IPX over ATM console command 21-7
  NetBIOS filtering configuration command 10-2
  OSPF configuration command 23-19
  SNMP configuration command 25-5
  SNMP console command 26-3
  TCP/IP Host Services configuration command 13-3
  Tunnel configuration command 7-36
  XTP configuration command 22-15
  XTP monitoring command 22-20
Delete Entry
  ARP configuration command 20-13
Detach
  IPX filter configuration command 18-33
Disable
  ASRT Bridge configuration command 7-15
  DLSw configuration command 27-44
  IP configuration command 16-26
  IPX configuration command 18-20, 19-6
  IPX filter configuration command 18-33
  IPX interface-based filter command 19-15
Disable *(continued)*
  LNM configuration command 12-6
  NetBIOS filtering configuration command 10-3
  OSPF configuration command 23-21
  SNMP configuration command 25-7, 25-8
  SNMP console command 26-3
  TCP/IP Host Services configuration command 13-3
  XTP configuration command 22-16
Disable Auto-Refresh
  ARP configuration command 20-13
display
  ARP over ATM console command 21-7
  CIP console command 21-7
  IPX over ATM console command 21-7
DLSw 9-21
  configuration environment 9-21
  configuration procedure 27-36
  configuration requirements 27-18
  configuring 27-1, 27-22
  configuring ASRT for DLSw 27-19
  configuring IP for DLSw 27-18, 27-20
  configuring NetBIOS for 9-21
  configuring the SDLC interface 27-21
  interoperability considerations B-1
  interoperability with IBM 6611
    bridge configuration B-1
    IP configuration considerations B-2
  monitoring 28-1
  multicast addresses 27-46
  overview 27-1
  summary of 27-37
  TCP interoperability considerations B-2
  X.25 requirement for QLLC 27-22
DLSw configuration commands
  ?(Help) 27-38
  Add 27-38
  BAN 27-42
  Close SAP 27-43
  Delete 27-43
  Disable 27-44
  Enable 27-45
  exit 27-56
  Join Group 27-46
  Leave Group 27-47
  List 27-48, 27-49
    priority 27-49
  netbios 27-51, 28-18
  Open SAP 27-51
  Set 27-52
DLSw console commands
  summary of 28-1
DLSw monitoring commands
  list
    dls sessions nb 28-9
    tcp capabilities 28-15
    tcp statistics 28-17

DLSw monitoring commands *(continued)*
- netbios 27-51, 28-18
- set
  - priority 28-19

DTE address wildcards 22-3
Dump
- ARP console command 21-2
- ARP over ATM console command 21-8
- CIP console command 21-8
- IPX console command 19-6
- IPX over ATM console command 21-8
- TCP/IP Host Services console command 14-2

Dump Routing Tables
- IP console command 17-5
- OSPF console command 24-8

E

Enable
- ASRT Bridge configuration command 7-18
- DLSw configuration command 27-45
- IP configuration command 16-28
- IPX configuration command 18-21, 19-7
- IPX filter configuration command 18-33
- IPX interface-based filter command 19-15
- LNM configuration command 12-6
- NetBIOS filtering configuration command 10-3
- OSPF configuration command 23-21
- TCP/IP Host Services configuration command 13-3
- XTP configuration command 22-16

Enable Auto-Refresh
- ARP configuration command 20-13

end system identifier (ESI) 6-4
Exit
- ARP configuration command 20-15
- ARP console command 21-5
- ARP over ATM configuration command 20-29
- ASRT Bridge configuration command 7-34
- ASRT Bridge console command 8-18
- BAN console command 8-20
- CIP configuration command 20-29
- IP configuration command 16-44
- IP console command 17-10
- IPX configuration command 18-30
- IPX console command 19-13
- IPX interface-based filter command 19-16
- IPX over ATM configuration command 20-29
- LNM configuration command 12-9
- LNM monitoring command 12-9
- NetBIOS Filtering configuration command 10-10
- NetBIOS Filtering console command 11-3
- OSPF configuration command 23-32
- OSPF console command 24-19
- SNMP configuration command 25-11
- SNMP console command 26-5
- TCP/IP Host Services configuration command 13-5

Exit *(continued)*
- TCP/IP Host Services console command 14-5
- Tunnel configuration command 7-38

F

Filter-lists
- IPX configuration command 18-22
- IPX console command 19-8

Filter-on command 10-3
Filters
- IPX console command 19-8

Flip
- ASRT Bridge console command 8-4

Forwarding process 16-11
Frame command 18-23
Frame size 9-23
- for NetBIOS 9-23

fully meshed network 6-3
Functional address to group address mapping 7-7

H

Hardware
- ARP console command 21-3
- ARP over ATM console command 21-9
- CIP console command 21-9
- IPX over ATM console command 21-9

Help
- ARP over ATM configuration command 20-16
- CIP configuration command 20-16
- IPX over ATM configuration command 20-16
- LNM configuration command 12-6
- LNM monitoring command 12-6

I

IGP (Interior Gateway Protocol) 23-1
Interface Addresses
- IP console command 17-6

Interface Summary
- OSPF console command 24-9

Inverse ARP
- configuration commands 20-11
- configuring 20-1
- overview 20-3

IP
- Adding UDP Broadcast destinations 16-13
- addressing network interfaces 16-1
- ARP net routing 16-6
- ARP subnet routing 16-6
- autonomous systems 23-1
- BootP/DHCP forwarding process 16-10
- configuring 16-1
- Disabling BOOTP forwarding 16-11
- Disabling UDP Forwarding 16-12

IP *(continued)*
dynamic routing 16-2
Enabling BOOTP forwarding 16-11
Enabling UDP Forwarding 16-12
interior gateway protocols 23-1
monitoring 17-1
OSPF and multicast routing 23-3
OSPF protocol 16-3, 23-1
RIP protocol 16-3, 23-1
sizes command 17-8
static routing 16-4
IP addresses
CIP 20-6
IP addresses in Classical IP 15-3
IP configuration commands
?(Help) 16-16
add 16-17
change 16-22
delete 16-24
disable 16-26
enable 16-28
exit 16-44
list 16-34
move 16-36
set 16-37
summary of 16-16
update 16-41
IP console commands
?(Help) 17-2
access controls 17-2
cache 17-3
counters 17-4
dump routing tables 17-5
exit 17-10
interface addresses 17-6
ping 17-7
route 17-8
static routes 17-8, 17-9
summary of 17-1
traceroute 17-9
udp-forwarding 17-10
IP filtering
access control 16-7
description 16-7
route filtering 16-10
IP monitor commands 17-7
IP multicast support
configuring the router 16-14
description 16-13
enrolling the router 16-15
IP routing 6-1
IP tunnel configuration commands 7-34
IP tunnel feature
ASRT bridge 7-1
IPX
addressing 18-1

IPX *(continued)*
configuring 18-1
description 18-1
monitoring 19-1
routing
update interval 18-3
IPX configuration commands 18-23
?(Help) 18-16
add 18-16
delete 18-19, 19-6
disable 18-20, 19-6
enable 18-21, 19-7
exit 18-30
Filter-lists 18-22
list 18-24
move 18-25
set 18-26
summary of 18-15
IPX console commands
?(Help) 19-2
access controls 19-2
cache 19-3
config 19-4
counters 19-5
dump routing tables 19-6
exit 19-13
Filter-lists 19-8
filters 19-8
ipxwan 19-8
ping 19-11
sizes 19-11
slist 19-12
summary of 19-1
IPX Filter configuration commands
?(Help) 18-31
attach 18-31
create 18-32
default 18-32
delete 18-32
detach 18-33
disable 18-33
enable 18-33
list 18-34
move 18-34
set-cache 18-35
update 18-35
Add 18-35
Add (IPX) 18-37
Add (RIP) 18-36
Add (Router) 18-35
Add (SAP) 18-36
Delete 18-40
Exit 18-41
Move 18-40
IPX Interface Filters
configuring 18-9

IPX monitoring commands
  Interface-based filter commands
    Clear 19-14
    Disable 19-15
    Enable 19-15
    Exit 19-16
    List 19-15
IPX over ATM
  configuration commands, summary 20-16
  description 20-10
  differences from CIP 20-16
  effect on ARP table 20-16
IPX over ATM commands
  ? (Help) 20-16
IPX over ATM configuration commands
  add 20-17
  change 20-23
  delete 20-25
  exit 20-29
  list 20-27
IPX routing 6-1, 6-4
IPX routing support by RFC 1483 6-3
ipxwan command 19-8

J

join
  OSPF configuration command 23-23
  OSPF console command 24-11
  Tunnel configuration command 7-36
Join Group
  DLSw configuration command 27-46

K

keepalive timer, setting for XTP 22-16
key parameters for Classical IP over ATM 15-4

L

LAN Network Manager
  See LNM 12-1
LE_ARP_REQUESTS, router-level 6-2
Leave
  OSPF configuration command 23-23
  OSPF console command 24-11
Leave Group
  DLSw configuration command 27-47
LIS 15-1
  See also logical IP subnets
List
  ARP configuration command 20-14
  ARP over ATM configuration command 20-27
  ASRT Bridge configuration command 7-21
  ASRT Bridge console command 8-5
  BAN console command 8-20

List *(continued)*
  CIP configuration command 20-27
  DLSw configuration command 27-48
  IP configuration command 16-34
  IPX configuration command 18-24
  IPX filtering configuration command 18-34
  IPX interface-based filter command 19-15
  IPX over ATM configuration command 20-27
  LNM configuration command 12-7
  NetBIOS Filtering configuration command 10-5
  NetBIOS Filtering console command 11-2
  OSPF configuration command 23-23
  SNMP configuration command 25-9
  SNMP console command 26-3
  TCP/IP Host Services configuration command 13-4
  Tunnel configuration command 7-38
  XTP configuration command 22-16
  XTP monitoring command 22-20
List Devices command 20-11
LLC device support 27-6
LNM
  agents and functions 12-1
  and LLC2 support 12-4
  configuration commands 12-5
  configuration restrictions 12-3
  configuring 12-4
  overview 12-1
LNM configuration commands
  disable 12-6
    agent port# 12-6
  enable 12-6
    agent port# 12-7
    configuration 12-7
    lnm port# 12-7
  exit 12-9
  help 12-6
  list 12-7
    password 12-7
    port port# 12-7
  set 12-9
LNM monitoring commands
  exit 12-9
  help 12-6
  list 12-8
    bridge 12-8
    lnm ports 12-8
    source 12-8
logical IP subnet (LIS) 15-1
logical IP subnets
  description 20-4

M

MAC addresses 7-29
MAC frames
  CSMA/CD 1-7

MAC frames *(continued)*
  token-ring 1-8
Mcache
  OSPF console command 24-11
Memory allocation 9-23
  for NetBIOS UI frames 9-23
meshed networks 6-3
Metric, using to determine OSPF costs 23-13
Mgroups
  OSPF console command 24-12
monitoring
  ARP over ATM console commands 21-6
  CIP console commands 21-6
  IPX over ATM console commands 21-6
Monitoring commands
  DLSw 9-20
  LNM 12-5
  NetBIOS 9-20
Move
  IP configuration command 16-36
  IPX configuration command 18-25
  IPX filter configuration commands 18-34
Mstats
  OSPF console command 24-13
multicast exploration 27-4
Multiple spanning trees, problems with 3-7

N
name lists
  committing changes 9-11
  configuring 9-9
  configuring and monitoring 9-24
  overview 9-9
  using 9-11
national personality, setting 22-8
neighbor discovery 27-4
neighbor priority 27-16
Neighbor Summary
  OSPF console command 24-14
NetBIOS 9-21, 9-22, 9-23
  ASRT bridge 7-1
  ASRT Bridge console command 8-18
  balancing traffic with SNA 27-17
  committing name list changes 9-11
  configuring for DLSw 9-21
  configuring name lists 9-9
  frame size 9-23
  memory allocation 9-23
    for UI frames 9-23
  name lists overview 9-9
  opening NetBIOS SAPS for DLSw 9-22
  session priority 9-22
  using name lists 9-11
NetBIOS Filtering
  basic configuration procedures 9-15
NetBIOS Filtering *(continued)*
  building a filter 3-6
  concepts 3-1, 3-4
  prompt 7-1
  simple and complex filters 3-6
  using bytes 3-5
  using host names 3-5
NetBIOS filtering commands
  add 9-24
  commands
    summary 9-24
  configuration 9-24, 9-26, 9-27, 9-28, 9-29, 9-42
  delete 9-26
  disable 9-27
  enable 9-28
  exit 9-42
  list 9-29, 9-31
  monitoring
    summary 9-24
  test 9-41
  update 9-37
NetBIOS filtering configuration commands
  ?(Help) 10-2
  create 10-2
  delete 10-2
  disable 10-3
  enable 10-3
  exit 10-10
  filter-on 10-3
  list 10-5
  summary of 10-1
  update 10-5
NetBIOS filtering console commands
  ?(Help) 11-1
  exit 11-3
  list 11-2
  summary of 11-1
NetBIOS Name Caching
  description 3-3
NetBIOS prompt 7-1, 8-1
NetBIOS-filtering prompt 8-1
Network hardware
  displaying ARP-registered 21-3
Network interface
  clearing 21-2
  console process 19-2
Non-volatile configuration memory
  configuring 9-20

O
Open SAP
  DLSw configuration command 27-51
operational software files 6-3
OSPF
  advantages over RIP 23-1

OSPF *(continued)*
- areas 23-5
- AS boundary routing 23-13
- configuring 23-1
- configuring over ATM 23-4
- converting from RIP 23-15
- description of 23-1
- designated router 23-2
- enabling 16-3, 23-5
- interface costs 23-16
- IP multicast routing 23-3
- IP multicast routing, sort string 23-11
- monitoring 24-1
- network interface parameters 23-9
- non-broadcast network interface parameters 23-11
- parameters for attached areas 23-5
- RIP comparison 23-14
- router IDs 23-5
- routing explained 23-1
- Sort String IP multicast routing 23-11
- virtual links 23-13

OSPF configuration commands
- ?(Help) 23-17
- add 23-18
- delete 23-19
- disable 23-21
- enable 23-21
- exit 23-32
- join 23-23
- leave 23-23
- list 23-23
- set 23-27
- summary of 23-17

OSPF console commands
- ?(Help) 24-2
- advertisement expansion 24-2
- area summary 24-5
- AS-external advertisements 24-6
- database summary 24-7
- dump routing tables 24-8
- exit 24-19
- interface summary 24-9
- join 24-11
- leave 24-11
- mcache 24-11
- Mgroups 24-12
- Mstats 24-13
- neighbor summary 24-14
- ping 24-16
- Routers 24-16
- size 24-17
- statistics 24-17
- summary of 24-1
- traceroute 24-16
- weight 24-19

overview of Classical IP over ATM 15-1
overview of routing and bridging 6-1

P

packet-filter 17-7
parameters, key configuration
- for Classical IP over ATM 15-4

partially meshed network 6-3
permanent virtual circuits 6-4
Ping
- ARP over ATM console command 21-9
- CIP console command 21-9
- IP console command 17-7
- IPX console command 19-11
- IPX over ATM console command 21-9
- OSPF console command 24-16
- TCP/IP Host Services console command 14-3

Port map 7-28, 8-4
Protocol
- ARP console command 21-4
- ARP over ATM console command 21-9
- CIP console command 21-9
- IPX over ATM console command 21-9

Protocol filters
- Ethernet Type 7-9, 7-14
- SNAP packets 7-9, 7-14

Protocols
- Adaptive Source Routing Transparent Bridge (ASRT) 5-1, 7-1, 8-1
- ARP 20-1, 21-1
- classical IP and ARP over ATM 20-1, 21-1
- displaying ARP-registered 21-4
- inverse arp 20-1
- IP 16-1, 17-1
- IPX 18-1, 19-1
- IPX and ARP over ATM 21-1
- LAN and Internetworking
  - IPX 18-1
  - OSPF 23-1
- OSPF 23-1, 24-1
- RIP 16-3, 16-32
- SNMP 25-1, 26-1
- TCP/IP Host Services 13-1, 14-1

PVCs 6-4

Q

QLLC
- ?(Help) 28-2
- configuring 27-37
- device support 27-10
- monitoring 28-1
- X.25 requirement for DLSw 27-22

Quality of Service (QOS) in Classical IP 15-4

R
refresh
CIP 20-6
Refresh timer
setting 20-15
remote DTE, searching for 22-4
resolving LAN destination addresses, router-level 6-2
Revert
SNMP console command 26-4
RFC 1483
overview 6-3
support for IPX routing 6-3
RFCs 6-3
RIP
converting to OSPF 23-15
enabling 16-3
OSPF routes 23-13
processing 16-32
RIP/SAP
disable/enable 16-26
monitoring 19-4
Route
IP console command 17-8
route filtering
IP filtering 16-10
Router
displaying ARP configuration of 20-14
Routers
OSPF console command 24-16
TCP/IP Host Services console command 14-5
Routing
OSPF 23-13
routing and bridging
adaptive source route transparent (ASRT) bridging 6-2
overview 6-1
overview of RFC 1483 support 6-3
pure source routing (SR) 6-2
pure transparent (TB) 6-2
RFC 1483 support for IPX routing 6-3
router-level bridging behavior 6-2
source route transparent (SRT) bridging 6-2
source routing (SR) and transparent (TB) 6-2
source routing (SR) to transparent (TB) 6-2
support for IPX routing 6-4
support of PVCs and SVCs 6-4
routing overview 6-1

S
SAPS 9-22
opening NetBIOS SAPs for DLSw 9-22
Save
SNMP console command 26-5
SDLC
device support 27-6
selector 6-4
serial port
service access points
opening 4-7
Session priority 9-22
for NetBIOS and DLSw 9-22
set
ARP configuration command 20-15
DLSw configuration command 27-52
IP configuration command 16-37
IPX configuration command 18-26
LNM configuration command 12-9
OSPF configuration command 23-27
SNMP configuration command 25-11
TCP/IP Host Services configuration command 13-5
XTP configuration command 22-16
Set-cache
IPX filter configuration command 18-35
Size
OSPF console command 24-17
Sizes
IPX console command 19-11
Slist
IPX console command 19-12
SNA
balancing traffic with NetBIOS 27-17
DLSw 27-1
SNMP
configuring 25-1
MIB II objects A-8
monitoring 26-1
SNMP configuration commands
?(Help) 25-3
add 25-3
delete 25-5
disable 25-7, 25-8
exit 25-11
list 25-9
set 25-11
summary of 25-1
SNMP console commands
?(Help) 26-2
add 26-2
delete 26-3
disable 26-3
exit 26-5
list 26-3
revert 26-4
save 26-5
statistics 26-5
summary of 26-1
SNMP Traps A-17
source route (SR) and transparent bridging (TB) 6-2

source route (SR) to transparent bridging (TB) 6-2
source route bridging 6-2
source route transparent (SRT) bridging 6-2
Source routing
  terminology and concepts
    all routes broadcast 2-27
    all stations broadcast 2-27
    bridge 2-27
    bridge number 2-27
    explorer frames 2-27
    ring number 2-27
    route 2-27
    route designator 2-27
    route discovery 2-27
    segment number 2-28
    single route broadcasting 2-28
    source routing bridging 2-28
    spanning tree 2-28
    transparent bridging 2-28
  threading 3-1, 3-8
Source Routing Bridge
  description of 2-10
  frame types 2-11, 2-14
  operation of 2-11
  Routing Information Field 2-12
  spanning tree explorer frame 2-13
  terminology and concepts
    bridge instance 2-16
    bridge number 2-16
    explorer frames 2-16
    interface number 2-16
    route 2-17
    route discovery 2-17
    segment number 2-17
    source routing 2-17
Source Routing Transparent Bridge
  architecture 2-18
  description of 2-17
  general description 2-18
  operation of 2-18
  terminology
    explorer frames 2-19
    routing information field (RIF) 2-19
    routing information indicator (RII) 2-19
    source routing 2-19
    spanning tree 2-19
    transparent bridging 2-20
Spanning Tree Bridge 2-2
  explore option 2-14
Spanning tree network
  balancing traffic loads 2-14
  simulation of 2-14
Spanning Tree protocol
  with 8209 bridges 3-7
Split-horizon routing
  for AppleTalk 18-13
SR and TB bridging 6-2
SR to TB bridging 6-2
SRT bridging 6-2
Static Routes
  IP console command 17-8, 17-9
Static routing
  default gateway 16-4
  default subnet gateways 16-4, 16-5
  static network/subnet routes 16-4, 16-5
Statistics
  ARP console command 21-4
  ARP over ATM console command 21-10
  CIP console command 21-10
  IPX over ATM console command 21-10
  OSPF console command 24-17
  SNMP console command 26-5
SVCs 6-4

T

Talk
  OPCON command 20-10, 21-1
TCP
  interoperability considerations with DLSw B-2
TCP connections 27-4
TCP/IP Host Services
  basic configuration procedures 13-1
  configuring 13-1
  monitoring 14-1
TCP/IP Host Services configuration commands
  ?(Help) 13-2
  add 13-2
  delete 13-3
  disable 13-3
  enable 13-3
  exit 13-5
  list 13-4
  set 13-5
  summary of 13-2
TCP/IP Host Services console commands
  ?(Help) 14-2
  dump 14-2
  exit 14-5
  interface 14-3
  ping 14-3
  routers 14-5
  summary of 14-1
  traceroute 14-4
TCP/IP, transporting X.25 traffic over 22-1
Threading
  AppleTalk end stations 3-9
  IP end-stations 3-8
  IPX end stations 3-9
timeout
  CIP 20-6

timeouts and refresh of Classical IP ARP table entries 15-2
Timer
  refresh 20-15
Traceroute
  IP console command 17-9
  OSPF console command 24-16
  TCP/IP Host Services console command 14-4
Translation cache
  clearing 21-2
  displaying 21-2
Transparent Bridge (STB)
  bridge ID 2-3
  description of 2-1
  Ethernet packet format translation 2-5
  network requirements 2-2
  operation of 2-2
  port ID 2-3
  root bridge ID 2-2
  routers and bridges 2-2
  shaping the spanning tree 2-3
  spanning tree bridges 2-5
  terminology and concepts
    aging time 2-6
    bridge 2-7
    bridge address 2-7
    bridge hello time 2-7
    bridge identifier 2-7
    bridge maximum age 2-7
    bridge priority 2-8
    designated bridge 2-8
    designated port 2-8
    filtering and permanent databases 2-8
    parallel bridges 2-9
    path cost 2-9
    port 2-9
    port ID 2-9
    port number 2-9
    port priority 2-9
    resolution 2-9
    root bridge 2-9
    root port 2-10
    spanning tree 2-10
transparent bridging 6-2
Traps, SNMP A-17
Tunnel
  ASRT Bridge configuration command 7-34
Tunnel configuration commands
  ?(Help) 7-35
  add 7-36
  delete 7-36
  exit 7-38
  join 7-36
  list 7-38
Tunnel feature
  prompt 7-1
Tunneling
  bridge tunnel 2-10

U

UDP Destination
  adding 16-13
UDP forwarding
  enabling/disabling 16-12
udp-forwarding
  IP console command 17-10
Update
  IP configuration command 16-41
  IPX filter configuration commands 18-35
  NetBIOS filtering configuration command 10-5

V

VCCs in Classical IP 15-3
Virtual Channel Connection (VCC)
  CIP 20-7
Virtual Channel Connections in Classical IP 15-3

W

Weight
  OSPF console command 24-19
wildcards, DTE address 22-3

X

X.25 Transport Protocol (XTP) 22-1
XTP
  backup peer function 22-4
  configuration commands
    Add 22-13
    Change 22-15
    Delete 22-15
    Disable 22-16
    Enable 22-16
    Exit 22-18
    Help (?) 22-13
    List 22-16
    Set 22-16
  configuration procedures 22-5
  monitoring commands
    Add 22-19
    Delete 22-20
    Exit 22-23
    Help (?) 22-19
    List 22-20
  setting keepalive timer 22-16
  using 22-1

Tell Us What You Think!

Nways Multiprotocol Access Services
Protocol Configuration and Monitoring Reference Volume 1
Version 1 Release 1

Publication No. SC30-3884-00

We hope you find this publication useful, readable, and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form. If you are in the U.S.A., you can mail this form postage free or fax it to us at 1-800-253-3520. Elsewhere, your local IBM branch office or representative will forward your comments or you may mail them directly to us.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	□	□

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	□	□
Complete	□	□
Easy to find	□	□
Easy to understand	□	□
Well organized	□	□
Applicable to your task	□	□

Specific comments or problems:

____

____

____

Please tell us how we can improve this book:

____

____

____

Thank you for your comments. If you would like a reply, provide the necessary information below.

Name	Address
Company or Organization	
Phone No.	

Tell Us What You Think!
SC30-3884-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Design & Information Development
Dept. CGF/Bldg. 656
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

Fold and Tape **Please do not staple** Fold and Tape

SC30-3884-00

Cut or Fold
Along Line

Part Number: 85H7919

Printed in U.S.A.

85H7919

SC30-3884-00