Nways Multiprotocol Routing Services

Protocol Configuration and Monitoring Reference Volume 1
Version 3.4

SC30-3680-10

Nways Multiprotocol Routing Services

Protocol Configuration and Monitoring Reference Volume 1
Version 3.4

SC30-3680-10

Note

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

Eleventh Edition (October 1999)

This edition applies to Version 3 Release 4 of the IBM Nways Multiprotocol Routing 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 mail your comments to:

Department CGFA Design & Information Development
IBM Corporation
PO Box 12195
Research Triangle Park NC 27709
U.S.A.

In addition, you can enter comments about documentation as well as other product support on the Web at:

http://www.networking.ibm.com/support/docs.nsf

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

Contents

Figures

Tables

Notices

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering the 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

Advanced Peer-to-Peer Networking
APPN
eNetwork
IBM
OS/2
SecureWay
VTAM

Microsoft, Windows, Windows NT, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

NetView is a trademark of Tivoli Systems, Inc. in the United States or other countries or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

This manual belongs to the product library described in “IBM 2210 Nways Multiprotocol Router Publications” on page xxviii and describes a group of protocols supported by the 2210. A specific 2210 might not support all of the features and functions described in these manuals. If a feature or function is device-specific, that restriction is indicated in the relevant manual.

This manual refers to the 2210 as either “the router” or “the device.” The examples in the library represent the configuration of a 2210, 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.

Getting 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 the CD-ROM. You can view the file with an ASCII text editor.

About the Software

IBM Nways Multiprotocol Routing Services is the software that supports the IBM 2210 (licensed program number 5801-ARR). 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 Multiprotocol Routing 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 2210.
- The Configuration Program for IBM Nways Multiprotocol Routing Services (referred to in this book as the *Configuration Program*) is a graphical user interface that enables 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 pre-loaded at the factory; it is shipped separately from the device as part of the software order.

  You can also obtain the Configuration Program for IBM Nways Multiprotocol Routing Services from the IBM Networking Technical Support home page. See *Configuration Program User’s Guide for Nways Multiprotocol and Access Services,*, 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 underlined as shown in the following example:

   `reload`

   In this example, you can enter either the whole command (reload) or its abbreviation (rel).
2. Keyword choices for a parameter are enclosed in brackets and separated by the word or. For example:

   ```
   command [keyword1 or keyword2]
   ```

   Choose one of the keywords as a value for the parameter.
3. 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.
4. 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.
5. Keyboard key combinations are indicated in text in the following ways:
   - **Ctrl-P**
   - **Ctrl -**

   The key combination **Ctrl -** indicates that you should press the Ctrl key and the hyphen simultaneously. In certain circumstances, this key combination changes the command line prompt.
6. Names of keyboard keys that you press are indicated like this: **Enter**
7. Variables (that is, names used to represent data that you define) are denoted by italics. For example:

   `File Name: filename.ext`

IBM 2210 Nways Multiprotocol Router Publications

Library reorganization: Beginning with Version 3.2, the following changes to the organization of the library took place:

- The information in the *Software User's Guide* titled **Understanding, Using and Configuring Features** was moved into a new manual, *Using and Configuring Features*.
- The chapters on using, configuring, and monitoring the DIALs feature were moved into the *Using and Configuring Features* book.

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 networking home page at:

`http://www.networking.ibm.com`

The following list shows the books that support the IBM 2210.

Operations and Network Management

SC30-3681
Software User's Guide

This book explains how to:

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

SC30-3992
Using and Configuring Features

SC30-3680
Protocol Configuration and Monitoring Reference Volume 1

SC30-3865
Protocol Configuration and Monitoring Reference Volume 2

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

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

SC30-3682
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

Online help
The help panels for the Configuration Program assist the user in understanding the program functions, panels, configuration parameters, and navigation keys.

GC30-3830
Configuration Program User's Guide for Nways Multiprotocol and Access Services

This book discusses how to use the Configuration Program.

GG24-4446
IBM 2210 Nways Multiprotocol Router Description and Configuration Scenarios

This book contains examples of how to configure protocols using IBM Nways Multiprotocol Routing Services.

Safety

SD21-0030
Caution: Safety Information - Read This First

This book provides translations of caution and danger notices applicable to the installation and maintenance of an IBM 2210.

The following list shows the books in the IBM 2210 Nways Multiprotocol Router library, arranged according to tasks.

Planning and Installation

GA27-4068

IBM 2210 Introduction and Planning Guide

GC30-3867

IBM 2210 Nways Multiprotocol Router Installation and Initial Configuration Guide

These books are shipped with the 2210. They explain how to prepare for installation, install the 2210, perform an initial configuration, and verify that the installation is successful.

These books provide translations of danger notices and other safety information.

Diagnostics and Maintenance

SY27-0345

IBM 2210 Nways Multiprotocol Router Service and Maintenance Manual

This book is shipped with the 2210. It provides instructions for diagnosing problems with and repairing the 2210.

Summary of Changes for the IBM 2210 Software Library

The following list applies to the changes in the software that were made in Version 3 Release 4:

- Frame Relay enhancements:
  - New Frame Handler (FH) support
  - PU throttling to handle bursts of traffic in support of 3745 controllers
  - New interface type (Frame Relay subinterface) to allow virtual interfaces on the same physical interface
  - Unnumbered IP support
- VPN enhancements:
  - CPE enhancements:
    - Policy information from LDAP servers is locally stored.
    - Policy quick configuration.
    - Policy consistency checking.
    - Policy information may now be retrieved from LDAP servers within an administrative domain.
    - IPSec tunnel ping.
  - IP enhancements:
    - Voice routing enhancements:
      - IP Header Compression on PPP (RFCs 2507, 2508, 2509)
      - Interleaving voice traffic between fragmented data packets on multi-link PPP
      - Interleaving voice traffic between fragmented data packets on Frame Relay
      - Bypassing PPP or Frame Relay packet compression and encryption for voice traffic
    - IP loopback address

      This support allows users to define IP addresses on a special interface to support TN3270 Gateway, Network Dispatcher, and IPSec requirements.
    - IPv6

    - An inter-domain routing function (BGP4+) is provided for IPv6 that supports IPv6 routing and addressing information and uses TCP6 for transport.
    - IPv6 traffic is supported over ATM ethernet LAN emulation without encapsulation or tunneling.
  - Multiple forwarding paths

    IP routing can use up to four equal-cost static routes to support multiple parallel links to a given address and mask.
  - IP route aggregation
  - Multicast enhancements:
    - Protocol Independent Multicast-Dense Mode (PIM-DM) for IPv4.
    - Network administrators can now control the flow of IP multicast data into and out of their networks by using inbound and outbound traffic filters.
  - Not-so-stubby area (NSSA)

    OSPF supports not-so-stubby area (NSSA) as defined in RFC 1587 and the latest Internet draft is now supported.
  - Random Early Detection (RED)
  - Differential services policing enhancements
  - VRRP enhancements:
    - The hardware MAC address may be used instead of a virtual MAC address to identify a redundant gateway; this can offer a performance improvement.
    - When more than one backup candidate is available, preempt options can be configured.
    - For selecting the master IP router, additional criteria, such as available route or network interface, can be used to support non-IP functions.
- Dial-on-demand alternate interface for WAN reroute
- TN3270 enhancements
  - LU capping
  - LU-pool load balancing
  - Talk 5 disconnect of TN3270 sessions
  - Additional reporting information
  - Support of addresses 1 and 255
- Network Dispatcher enhancements
  - Advertising of network dispatcher cluster addresses by routing protocols
  - A new SSL Advisor
- DLSw SDLC PU1 support
- Ethernet encapsulation support for both ethernet type II (default) and 802.3 simultaneously on the same interface
- DHCP enhancements:
  - Hardfile backup for lease information
  - Multiple IP address support for DHCP interfaces
  - Short lease support
- RADIUS enhancements
  - Radius scalability
  - Login of Last Resort
- L2TP Scalability
- Thin Server enhancement

  Connection to an alternate or back-up master server

Summary of Changes

- Service file retrieval enhancements

Clarifications and corrections

In hard copy and PDF, the technical changes and additions are indicated by a vertical line (|) to the left of the change.

Getting Help

At the command prompts, you can obtain help in the form of a listing of the commands available at that level. To do this, type **?** (the **help** command), and then press **Enter**. Use **?** to list the commands that are available from the current level. You can usually enter a **?** after a specific command name to list its options.

Exiting a Lower Level Environment

The multiple-level nature of the software places you in secondary, tertiary, and even lower level environments as you configure or operate the 2210. To return to the next higher level, enter the **exit** command. To get to the secondary level, continue entering **exit** until you receive the secondary level prompt (either `Config>` or `+`).

For example, to exit the ASRT protocol configuration process:

```
ASRT config> exit
Config>
```

If you need to get to the primary level (OPCON), enter the intercept character (**Ctrl-P** by default).

Part 1. Configuring and Monitoring Bridge Functions

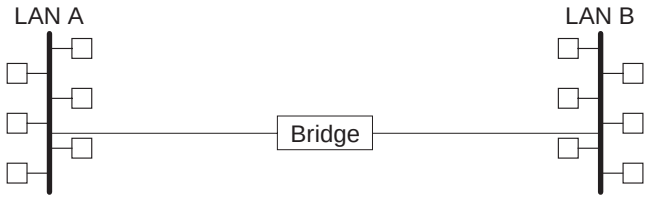
Chapter 1. Bridging Basics

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

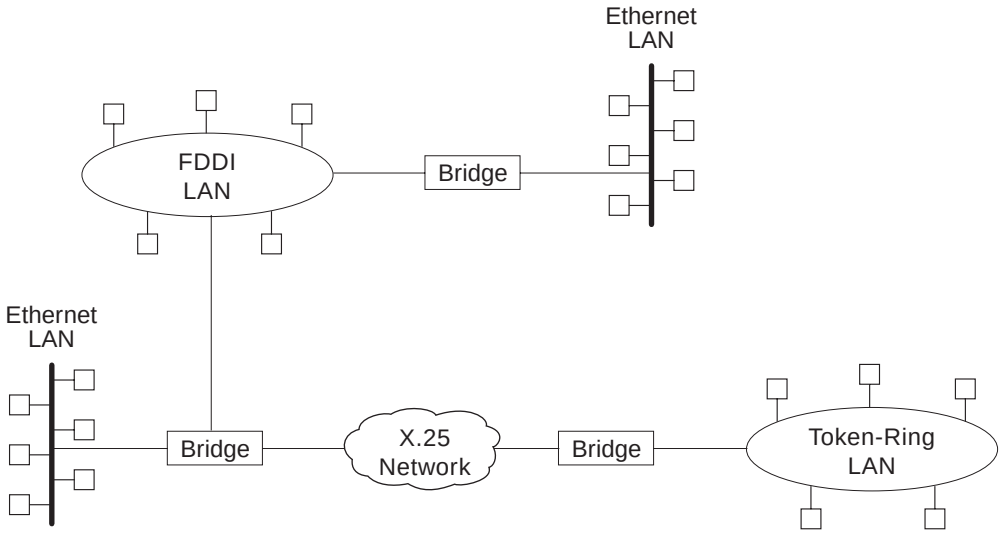
Bridging Overview

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 medium access control (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 on page 4 illustrates examples of simple and complex bridging configurations.



Simple Bridge Connecting Two Homogeneous (Ethernet) LANs



Complex Bridging Configuration Connecting Different LAN Technologies

Figure 1. Simple and Complex Bridging Configuration

Bridging and Routing

The 2210 can perform both bridging and routing. Protocol filtering is the process that determines whether the incoming data is routed or bridged.

Protocol Filtering

When processing an incoming data packet, 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 medium access control (MAC) address.

Table 1 shows how the “Bridge or Route?” question is answered based on the destination address contents.

Table 1. 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 frame is routed. Otherwise, the frame is bridged.
Unicast	The frame is bridged.

Routing and Bridging on a Per-Interface Basis

For IP, IPX, and AppleTalk, the following rules apply for routing or bridging over a specific interface:

- Packets are routed if a specific protocol is configured for the receiving interface
- Packets are filtered if you configure specific protocol filters on the receiving interface
- Packets that are not routed or filtered are candidates for bridging, depending on the destination medium access control (MAC) address.

Router Connections

Connecting at Layer 3 with a router enables 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 2 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” on page 11 for more information on source routing and transparent bridges.

The Link Layer maintains physical addressing schemes (versus 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. Because 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 enable 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.

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).

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 on page 4). 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 can 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

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 enable 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 2210’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 connect the 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 in the speed at which data is transmitted. WAN connections may be slower than LAN connections. This difference in speed can be significant when running time-sensitive applications. Another difference is 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 Logical Link Control (LLC) 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 2 on page 8 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.

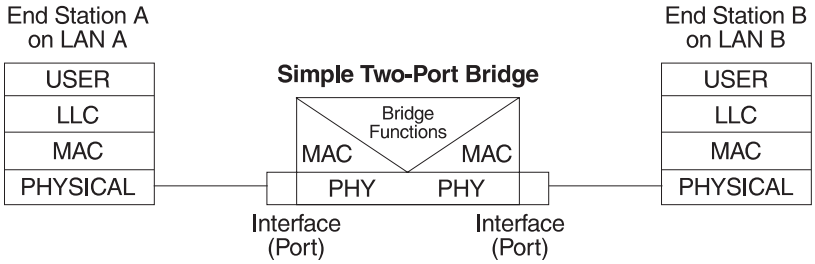


Figure 2. Two-Port Bridge Connecting Two LANs

Operation Example 2: Remote Bridging over a Serial Link

Figure 3 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.

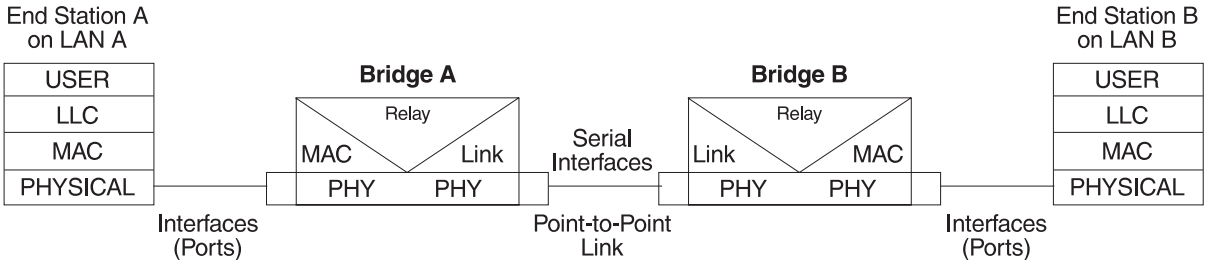


Figure 3. Bridging Over a Point-to-Point Link

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

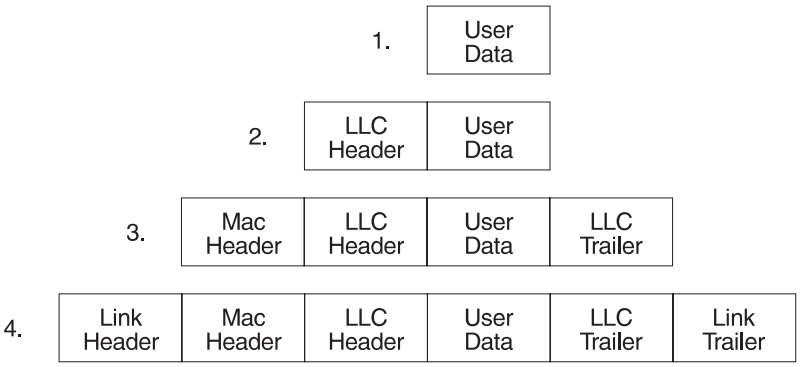


Figure 4. Data Encapsulation Over a Point-to-Point Link

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 5 shows the MAC frame formats supported by the bridge. 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.

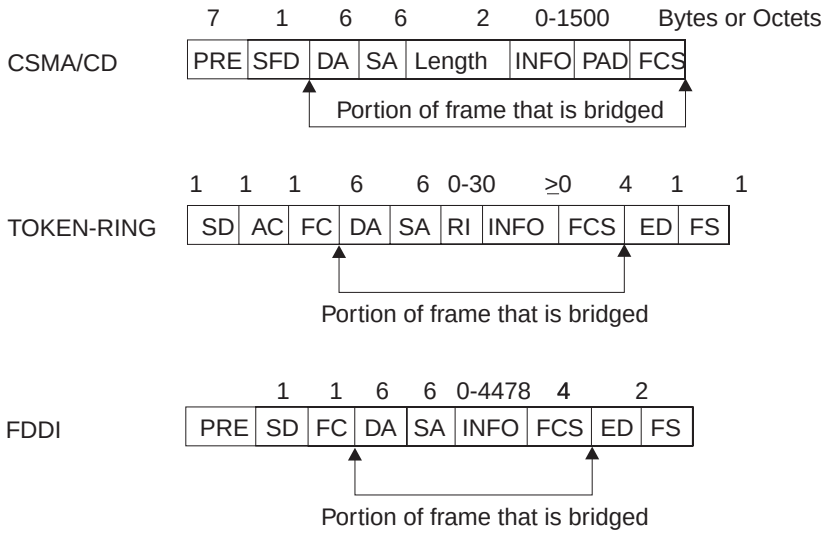


Figure 5. Examples of MAC Frame Formats

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), 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 originated the frame. This field can 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 Route Bridging (SRB)” on page 19.
- *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 Route Bridging (SRB)” on page 19.
- *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

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 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. When the new bridge is added, 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. The features that spanning tree provides include:

- *Loop detection*. Detects and eliminates physical data link loops in extended LAN configurations.
- *Automatic backup of data 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 medium access control (MAC) 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 6 on page 14 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. Spanning Tree Default Values

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

Table 2. Spanning Tree Default Values (continued)

Bridge 1	Bridge 2	Bridge 3
Port 2 Priority: 128 Path Cost: 17857	**Port 2** Priority: 128 Path Cost: 17857	**Port 2** Priority: 128 Path Cost: 17857
Port 3 Priority: 128 Path Cost: 17857	**Port 3** Priority: 128 Path Cost: 17857	**Port 3** Priority: 128 Path Cost: 17857

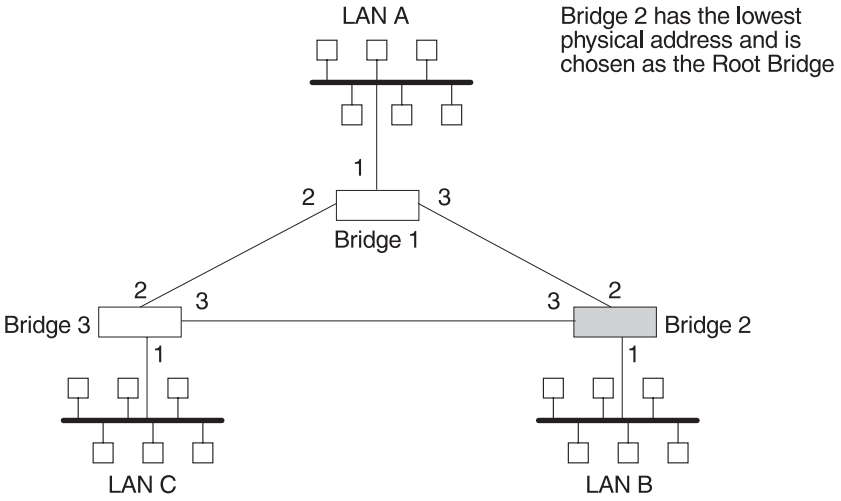


Figure 6. Networked LANs Before Spanning Tree

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 on page 13 is shown in Figure 7 on page 15 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.

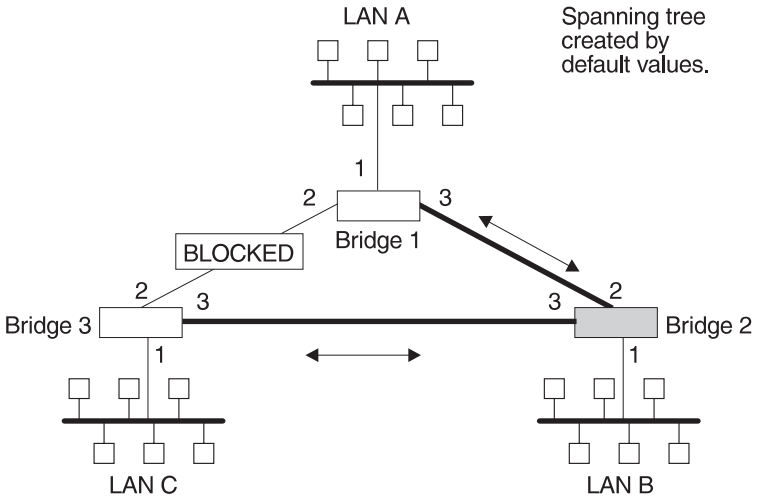


Figure 7. Spanning Tree Created with Default Values

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 8 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.

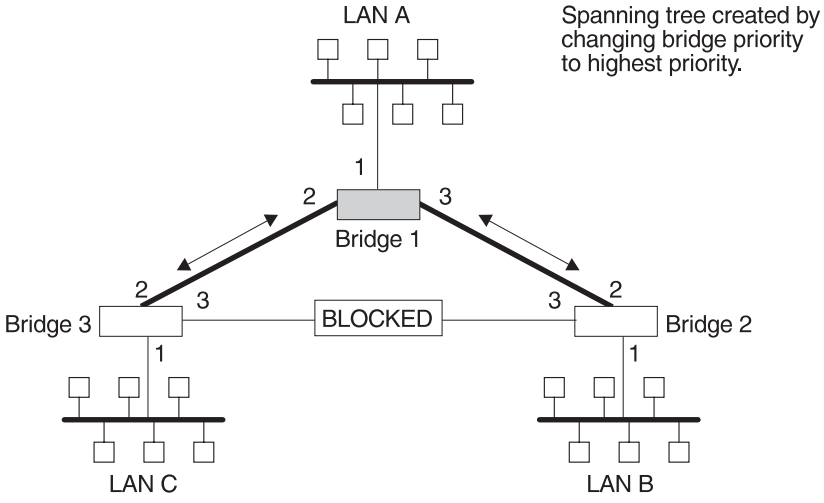


Figure 8. User-Adjusted Spanning Tree

Spanning Tree Bridges and Ethernet Packet Format Translation

The 2210 Spanning Tree Bridge protocol provides packet forwarding for the bridging devices in accordance with 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 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 that bridging is enabled on the circuit. IP tunneling does not have to be used.

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

Transparent Bridging and ATM

The ATM interface forwards transparent frames from Ethernet and Token-Ring networks, provided bridging is enabled on the virtual channel connection (VCC). IP tunneling does not have to be used.

Hello BPDUs are generated and transmitted for each VCC configured for transparent bridging. The spanning tree protocol causes ATM VCCs 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 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 protocol-independent device that connects local area networks (LANs). These devices operate at the data link layer, storing and forwarding data packets between LANs.

Bridge Address
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 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 amount of 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
A unique identifier that the spanning tree algorithm uses 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 address of the lowest-numbered port. You can override the default address with the **set bridge** configuration command.

Bridge Maximum Age
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 can cause 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 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 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 port ID of the designated bridge attached to the LAN.

Filtering and Permanent Databases

Databases that 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 monitoring 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 monitoring 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. You 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 monitoring process.

Parallel Bridges

Two or more bridges connecting the same LANs.

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

Spanning Tree
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 Route 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.

Bridging Methods

The source routing bridge (SRB) provides local bridging over 4 and 16 Mbps token-rings, as shown in Figure 9. It can also connect remote LANs through a telecommunications link operating at speeds up to E1.

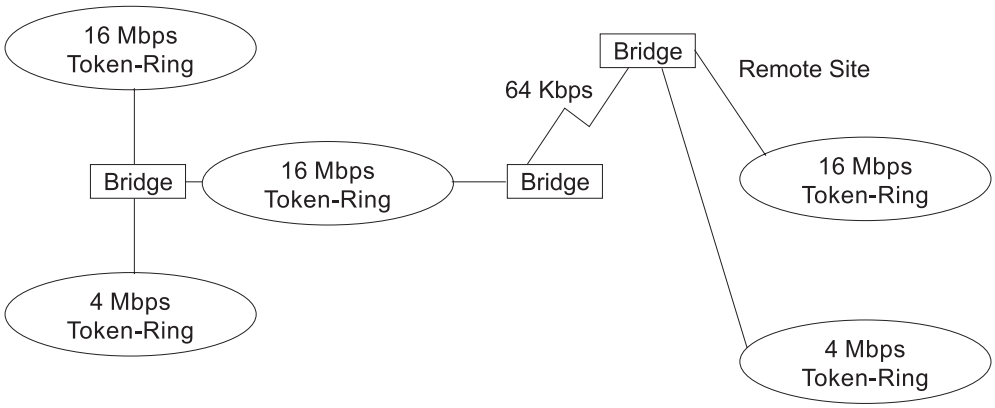


Figure 9. Example of Source Routing Bridge Connectivity

Among its features, the source routing bridge provides:

- *Bridge compatibility*. You can use the bridge to connect PC LANs running systems such as OS/2®, PC LAN Manager, and NetBIOS. The bridge can also carry 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 10 provides a closer look at the format of the source address field of a source routing frame.

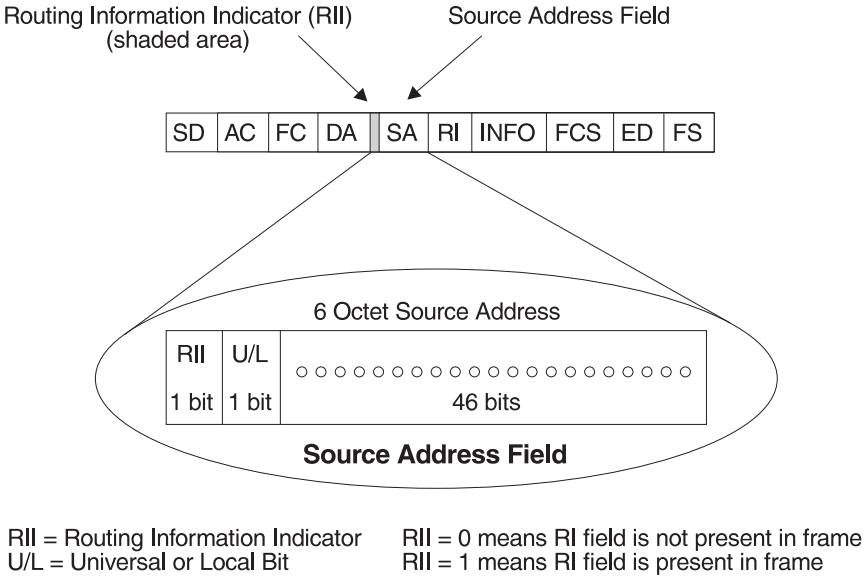


Figure 10. 802.5 Source Address Format

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 11 provides a closer look at the format of the Routing Information Field.
The following information describes each field in the RIF:

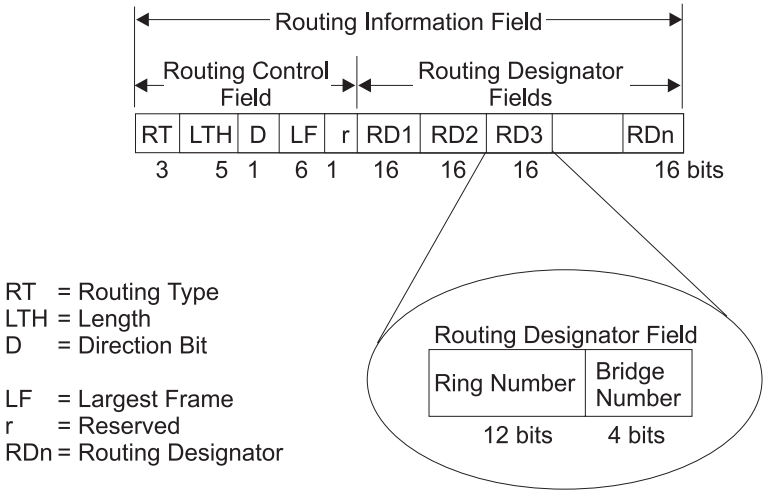


Figure 11. 802.5 Routing Information Field

- **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 from 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 cannot 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 that enables 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 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 enables 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 12 on page 24, 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.

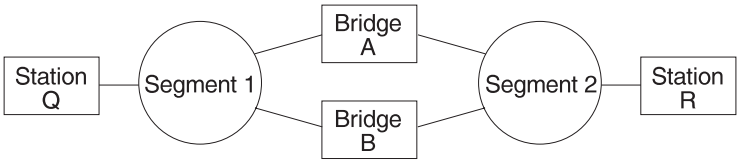


Figure 12. Example of Parallel Bridges

Balancing Traffic Loads

You can also use the spanning tree explore option for load balancing. For example, in Figure 13, 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 enables parallel bridges to share the traffic load.

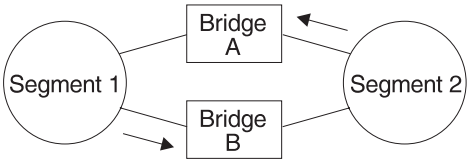


Figure 13. Using Spanning Tree Explore for Load Balancing

Note: For source routing to work, some end-node applications such as the PC LAN programs 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.

Source Routing Bridging and Frame Relay

If source routing bridging is enabled, source-routed frames are forwarded between the Frame Relay interface and the bridging forwarder. You can configure the bridge to treat each Frame Relay virtual circuit as a bridge port with a unique ring number. Additionally, Frame Relay virtual circuits that are not configured as bridge ports can be grouped together as a single multiaccess bridge port with a unique ring number. For more information, see “Understanding Multiaccess Bridge Ports” on page 48. Some virtual circuits that are not part of the active data path are BLOCKED in order to maintain the loop-free topology.

Source Routing Bridging and ATM

If source routing bridging is enabled on the virtual channel connection (VCC), source-routed frames are forwarded between the ATM interface and the bridging forwarder. A unique destination ring number is configured for each VCC. Some VCCs 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 14, 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 enables you to create two separate networks. These networks do not communicate unless they physically interconnect at some other point.

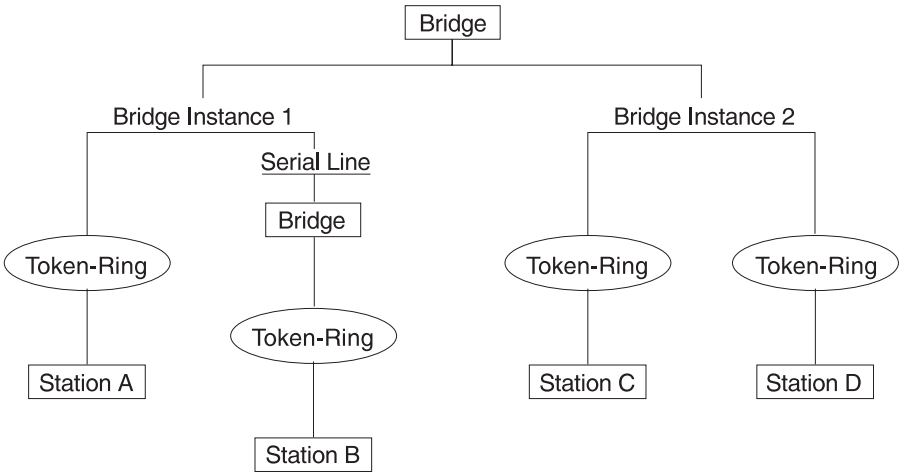


Figure 14. Bridge Instances within a Bridge

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 device 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 acknowledgment 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 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.

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 21 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 the 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 the 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 15. The arrows represent the data path.

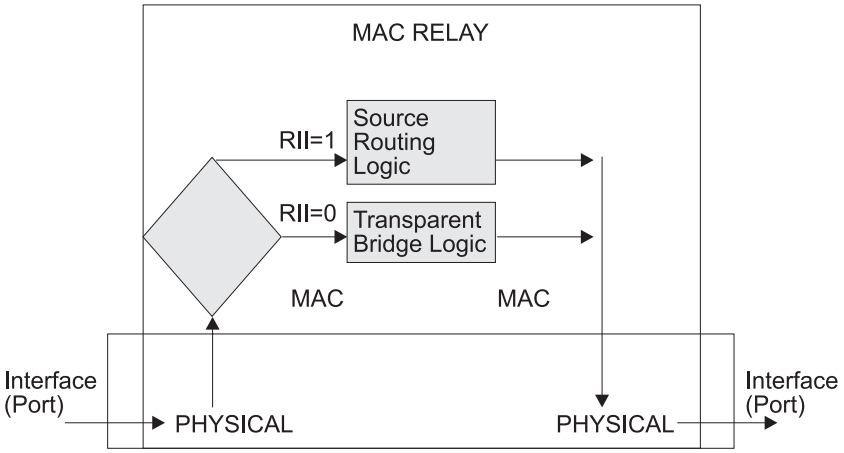


Figure 15. SRT Bridge Operation

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 forwards the packet.

Source Routing Transparent Bridging and Frame Relay

If SRT bridging is enabled on the circuit, source routed and transparent frames are forwarded between the Frame Relay interface and the bridging forwarder.

Source Routing Transparent Bridging and ATM

If SRT bridging is enabled on the virtual channel connection (VCC), source routed and transparent frames are forwarded between the ATM interface and the bridging forwarder.

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

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

Spanning Tree

A topology of bridges in which there is only one data route between any two end stations.

Transparent Bridging

A type of bridging that 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 a single ASRT bridge. This extended function enables 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 with 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.

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 conversely). 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 following 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 can create networks comprised of two types of domains (depending upon the bridging method):

- Source routing domains
- Transparent bridging domains

Figure 16 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 because the source routing spanning tree does not allow multiple paths between any two stations in the domain.

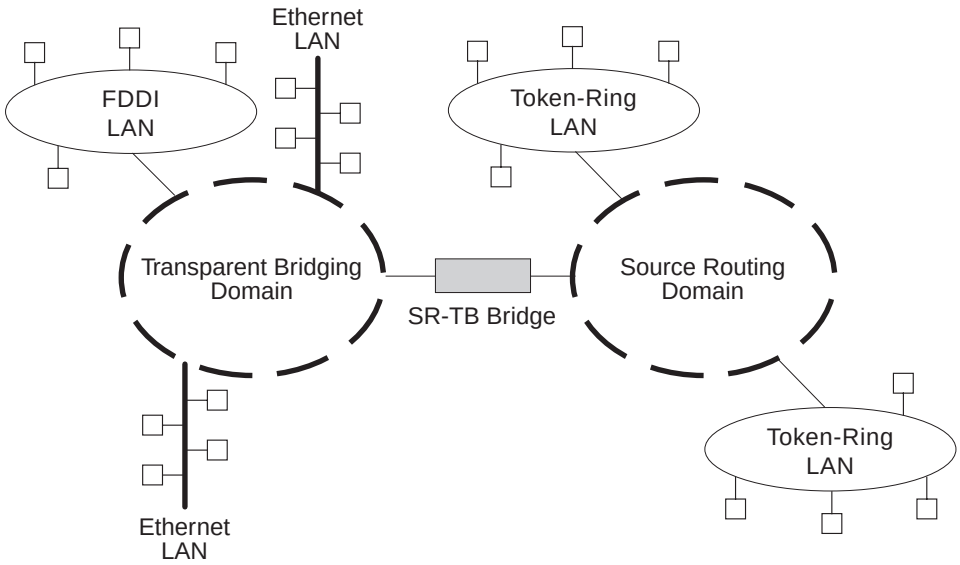


Figure 16. SR-TB Bridge Connecting Two Domains

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 because 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 3.

Table 3. 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.
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.
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 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.

Table 3. SR-TB Bridge Decision Table (continued)

Type of Frame Received	Action Taken by SR-TB Bridge
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, the bridge copies the 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, the bridge first determines if it has routing information associated with the destination address carried in the frame. If yes, the 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, the 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 unaware 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 17 on page 33 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

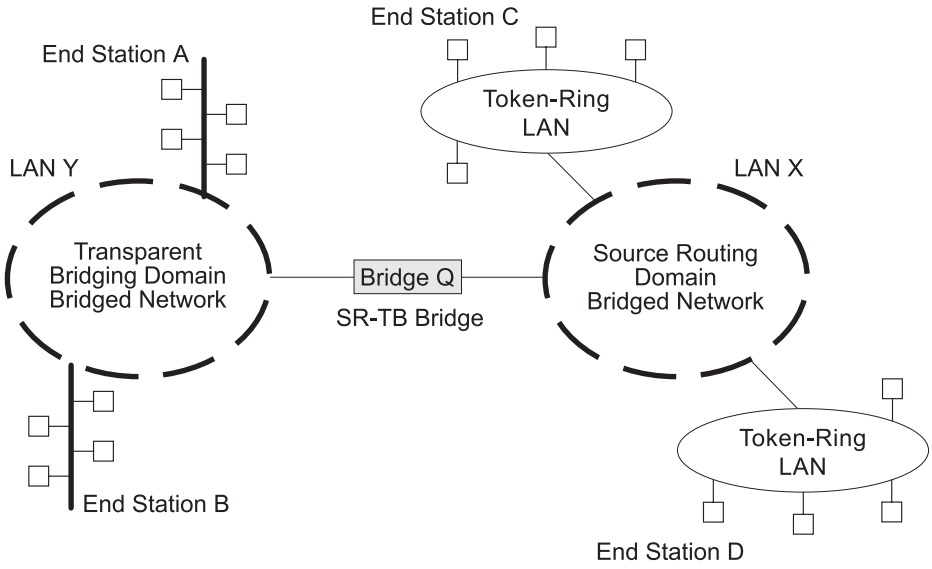


Figure 17. SR-TB Bridging Examples

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. Because 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 circuit.

SR-TB and ATM

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

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

This section describes 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 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 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.

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
A number that 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 Route Bridging
A bridging mechanism that routes frames through a multi-LAN network by specifying in the frame the route it will travel.

Spanning Tree
A topology of bridges such that there is only one data route between any two end stations.

Transparent Bridging

A type of bridging that 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 conversely. 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 enables 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 a problem 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 acknowledgment 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 and SRB Bridges

Because 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, 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 communication 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.

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

Bridging Methods

Bridge Personality	SR <-> TB Conversion Enabled?	Interface Type & Bridging Method Setting			
		Token Ring	Ethernet	Serial Line or Tunnel	ATM
ASRT	Yes	SR & TB	TB	SR & TB	SR & TB
ASRT	Yes	SR	TB	SR & TB	SR & TB
ASRT	Yes	SR or TB	TB	SR & 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 = Adaptive 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

The bridging tunnel (encapsulation) is another feature of the ASRT bridge software. By encapsulating packets in industry-standard TCP/IP packets, the bridging device 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 device, 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 18 on page 40 shows an example of an IP internetwork using the tunnel feature in its configuration.

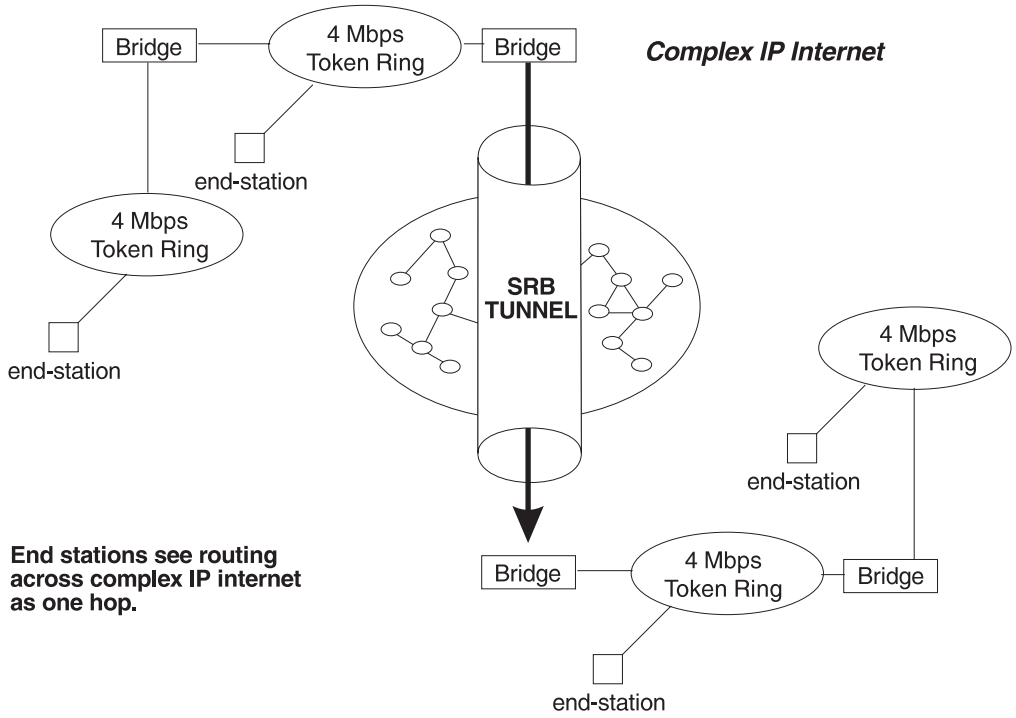


Figure 18. Example of the Bridge Tunnel Feature

The tunnel is transparent to the end stations. The bridging devices 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, enabling network administrators to distribute traffic over the least expensive route.
- *Dynamic Routing*. OSPF looks for the least-cost path, and also 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, 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 16. Using OSPF” on page 325.

TCP/IP Host Services (Bridge-Only Management)

The IBM 2210 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 configurations 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 monitoring interface, TCP/IP Host Services is handled as a new protocol having its own configuration and monitoring prompts. These prompts are accessed via the **protocol** command in `talk 6` and `talk 5`.

Bridge-only management function is activated by assigning an IP address to the bridge and enabling TCP/IP Host Services (see "Chapter 12. Configuring and Monitoring TCP/IP Host Services" on page 197). 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 are available whenever bridging is an option in the device software load.

Bridge-MIB Support

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

- dot1dStaticTable
- dot1dTpFdbTable
- dot1dPortPairTable

NetBIOS Name Caching

The NetBIOS name caching feature enables the bridging device to significantly reduce the number of Name-Query frames that leave an originating ring and are forwarded through a bridge. Configuring NetBIOS name caching is part of the NetBIOS configuration. Details are in "NetBIOS Name Caching and Route Caching" on page 143.

NetBIOS 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 136.

NetBIOS Name and Byte Filters

NetBIOS filtering is a feature that enables you to enhance the performance of ASRT Bridging. This feature lets you configure specific filters using the device 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. The 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 enable 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 43).

Configuration and monitoring of NetBIOS is described at “Chapter 8. Configuring and Monitoring NetBIOS” on page 153. 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 fields of the following NetBIOS packet types are examined:

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

The 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 device 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 ANDed with the bytes in the NetBIOS header before the device 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-item was specified.

When a match is found between a filter-item and a packet, the device:
- 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 device:
- 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 device 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 following 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 (TBs), 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 (SRBs) transmit spanning tree explorer (STE) frames across other SRBs 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 SRBs, 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 IBM 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 enables 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. 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 enable 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 because 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 the 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 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.
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.

SR-TB Duplicate MAC Address Feature

The duplicate MAC address (DMAC) feature enables you to attach an SR-TB bridge to a SR bridged network that has duplicate MAC addresses configured. The duplicate MAC address feature can be enabled with two options:

- **Duplicate MAC Feature without Load-Balance**

  This option enables you to enable duplicate MAC addresses without load-balancing. In this case, only one RIF is learned for the duplicate MAC address and aging is performed on this learned RIF. All stations from the TB domain will use this one RIF to communicate with that MAC address. When the entry for this RIF ages out, the next frame will be sent from the TB domain as a Spanning Tree Explorer (STE) frame.

- **Duplicate MAC Feature with Load-Balance**

  This option enables you to enable duplicate MAC addresses with load-balancing and can be enabled only after enabling DMAC without Load-Balance. In this case, two RIFs are learned and maintained for each duplicate MAC address. Each of the two RIFs will have its own aging timer. Whenever the bridge receives a frame with a particular RIF, the aging value corresponding to that RIF will be refreshed. The first time a station from the TB domain sends a frame to a duplicate MAC address, the bridge software decides which RIF will be used to send that frame. All subsequent frames from the sending station will be sent using that same RIF. The bridge will maintain primary and secondary RIFs for up to seven duplicate MAC addresses. If you specify separate age values for duplicate MAC addresses, the appropriate value will be used to age out entries corresponding to that duplicate MAC address, enabling you to tune the aging value for duplicate MAC addresses.

Bridging on ATM

The device supports bridging over native ATM (RFC 1483). When bridging over native ATM, multiple (virtual) ports may be configured for a single (physical) interface.

RFC 1483 Support for Bridging

RFC 1483 specifies an LLC value of 0xAA-AA-03 and an OUI value of 0x00-80-C2 for bridged protocols. The 2-octet PID portion of the SNAP header, in the case of bridged protocols, specifies the bridged media, and additionally, whether the original frame check sequence (FCS) is preserved within the original bridged PDU. The PID values for the different media are specified. Refer to RFC 1483 for further details.

The ATM interface will forward bridged MAC frames to and from Token Ring/802.5, Ethernet/802.3. One bridge port is used per VCC. While configuring a bridge port on an ATM interface, you must specify a VCC that is permanently tied to that port. Bridged frames received on a port/VCC are sent out on one or more ports/VCCs as per the bridging protocol being used and the bridging configuration. Once a bridge port is configured on an ATM interface and has a VCC associated with it, it functions as a normal bridging port on a legacy LAN. The association of the port with an ATM interface is transparent to the user and to the bridging function.

Bridging Features

When configuring a bridge port on an ATM interface, the user must specify whether PVC or SVC support should be used. For PVC support, you must provide the VPI and VCI for the PVC. For SVC support, you must provide the remote ATM address as well as the selector to be used for the local address.

Note: Unlike PVC support, using SVCs does not require any configuration at the intermediate switches.

Once a port has been added on an ATM interface, the bridging configuration commands that require a port number as a parameter can be used with this port number.

Refer to “Chapter 27. Using ARP” on page 571 and “Chapter 6. Configuring and Monitoring Bridging” on page 67 for additional information on configuring bridging over ATM.

Understanding Multiaccess Bridge Ports

A multiaccess bridge port is a bridge port that includes all Frame Relay virtual circuits that are not individually configured as bridge ports. The multiaccess bridge port is assigned a unique bridge segment number, which is used for source routing bridging.

A multiaccess bridge port has the following bridging characteristics:

- It supports source routing (SR) bridging only.
- Fully meshed configurations support any-to-any connectivity and may use the Spanning Tree protocol to prevent bridging loops.
- Non fully meshed configurations only support branch-to/from-datacenter connectivity because bridging between virtual circuits on the same multiaccess segment is not supported. This configuration cannot use the Spanning Tree protocol, so forwarding of STE frames must be enabled. By default, Spanning Tree protocol is disabled and forwarding of STE frames is enabled.

  Note: This is the preferred configuration because Spanning Tree protocol can consume considerable WAN bandwidth and most configurations are not fully meshed.
- It requires the 1-to-N bridge virtual segment.
- It provides protocol-independent connectivity between like end stations and limited connectivity between end stations on dissimilar media.
- It can provide effective data catchers for multiple IBM 2218 devices. (See “Interoperating with IBM 2218 Devices” on page 49.)

The Multiaccess Database

Each multiaccess bridge port maintains a multiaccess database that maps the next-hop segment number to the Frame Relay virtual circuit on which the frame was received. Database entries are built or updated as the segment receives ARE, STE, or specifically routed frames from the circuits. STE and ARE frames that are to be forwarded onto the multiaccess segment are flooded to all virtual circuits in the multiaccess segment. Specifically routed frames that are to be forwarded onto the multiaccess segment are only forwarded if there is a multiaccess database entry that maps the next-hop segment number to a virtual circuit.

The software “ages out” entries in the multiaccess database at a rate that you specify with the **multiaccess-age** command.

Configuring Multiaccess Bridge Ports

The following example illustrates how to configure multiaccess bridge ports on Frame Relay interfaces 1 and 4. Port 5 is the next available bridge port and this is the first time source routing is being enabled.

```
* talk 6
Config> prot asrt
ASRT Config> add multiaccess-port
ASRT Config> Interface number [0]? 1
ASRT Config> Port Number [5]?
ASRT Config> Segment Number for the port in hex (1 - FFF) [001]? 300
ASRT Config> Bridge Number in hex (0 - 9, A - F) [0]? 2
ASRT Config> Bridge Virtual Segment Number (1 - FFF) [001]? CCD
ASRT Config> add multiaccess-port
ASRT Config> Interface number [0]? 4
ASRT Config> Port Number [6]?
ASRT Config> Segment Number for the port in hex (1 - FFF) [001]? 400
```

Note: You are not prompted for bridge number and virtual segment number after configuring the first multiaccess bridge port.

Interoperating with IBM 2218 Devices

Using multiaccess ports with 2210/2212/2216 devices as data catchers can provide a high density, high availability topology for the 2218s in your network.

- High density occurs because many 2218 devices can connect to a datacenter bridge through a single multiaccess bridge port.
- High availability occurs by configuring a single 2218 to connect to primary and backup datacenter bridges through their multiaccess bridge ports. The 2218 can then switch between the primary and backup circuits as the 2218 detects problems in the Frame Relay network.

For the 2218 to switch between the primary and central bridges without losing LLC connections between itself and the central bridge you must:

- Configure the primary and backup datacenter bridges with the same bridge 1-to-N virtual segment number.
- Configure the primary and backup datacenter bridges with the same source route bridge number.
- Configure the primary and backup datacenter bridges with the same multiaccess segment number.

Note: This configuration only supports branch-to-datacenter connectivity.

Figure 19 on page 51 shows a typical network connection between 2210s and 2218s.

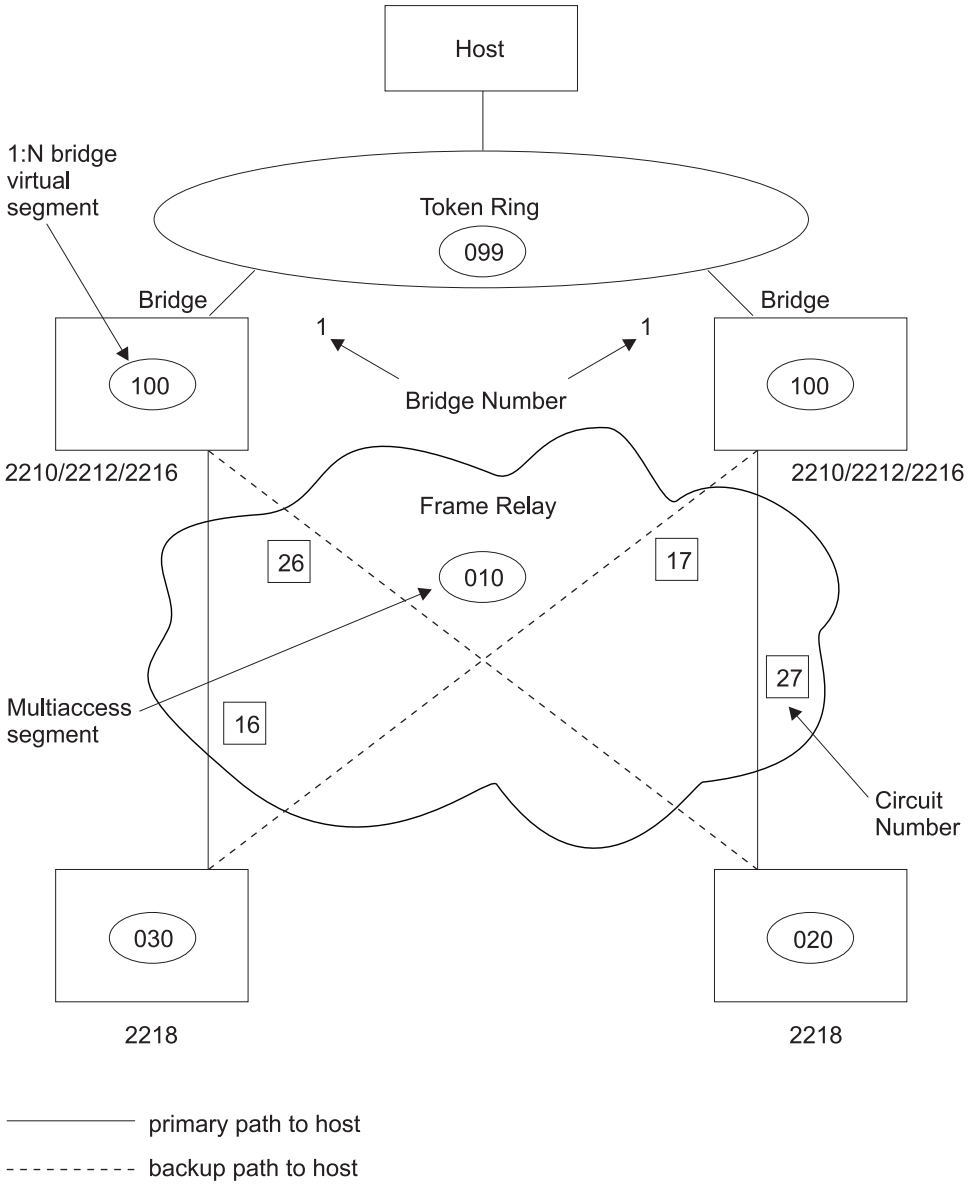


Figure 19. Sample Configuration with 2218 and Multiaccess Bridge Ports

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

This chapter describes the Boundary Access Node (BAN) feature on the 2210. 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

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 2210 software. This feature enables IBM Type 2.0 and 2.1 end stations connected to a 2210 to make a direct connection via Frame Relay to an SNA node supporting the RFC 1490/2427 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 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 20. Though their data actually passes through a 2210 and over a Frame Relay network, this is transparent to the end stations.

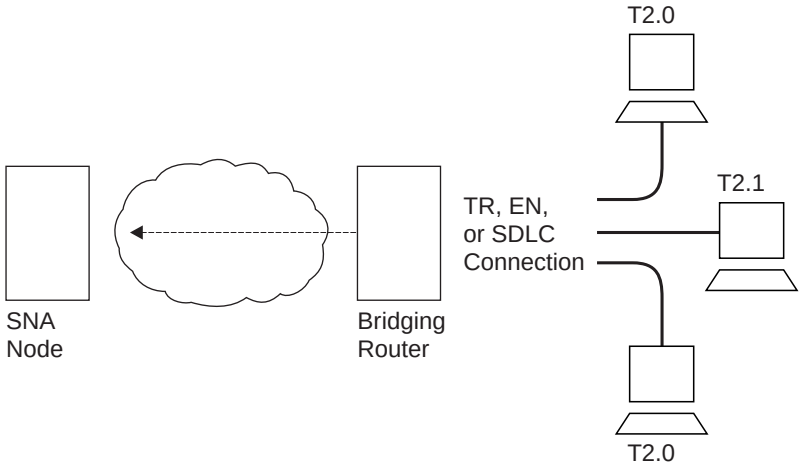


Figure 20. Direct Connection of End Stations to an SNA Node Using BAN

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. Ability to bridge Ethernet, Token-Ring, or FDDI 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.
3. 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.

4. 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 59 for more information.

There are two ways to use the BAN feature:

- Straight bridging using the 2210’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 21 on page 54 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 1490/2427) frames, and bridges directly to the SNA node.

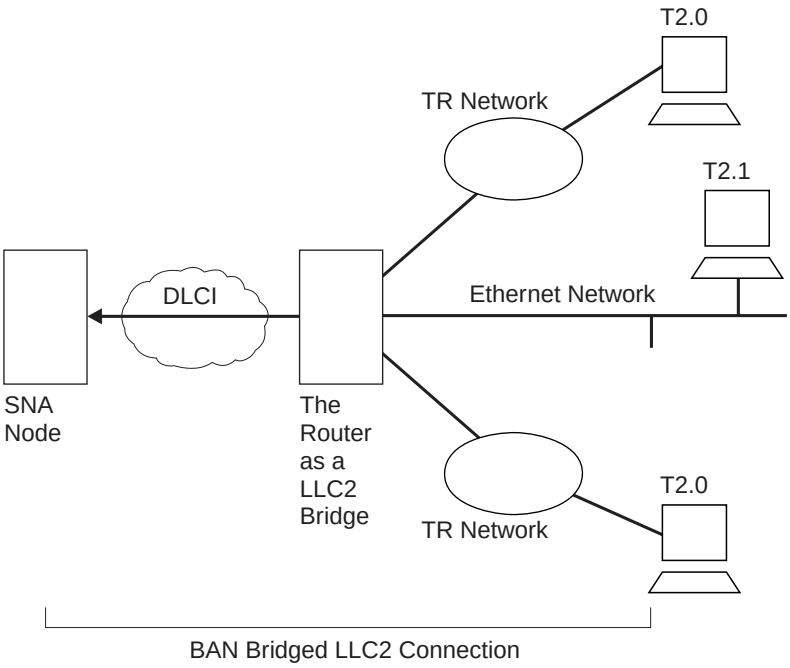


Figure 21. BAN Type 1: The Router as an LLC2 Bridge

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, or Ethernet format, provided the bridge is configured to support that type of frame.

Figure 22 on page 55 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.

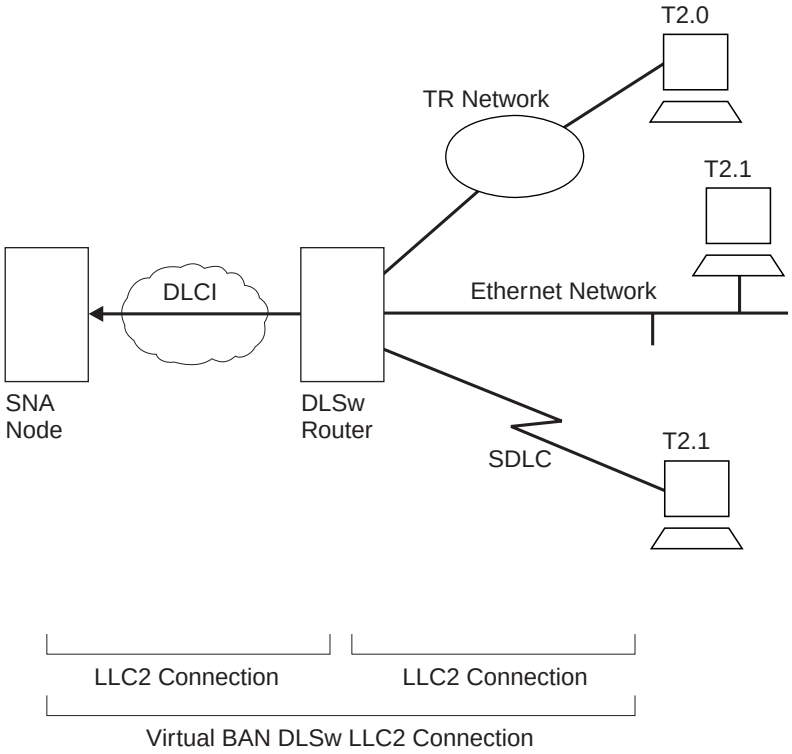


Figure 22. BAN Type 2: Local DLSw Conversion

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. Configure the router for DLSw (BAN type 2 only)

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 2210 must be identical to the BAN DLCI MAC address of the ISDN backup 2210. Also, the value of the internal bridge segment of the 2210 must be different from the value of the internal bridge segment of the backup 2210. See "Setting up Multiple DLCIs" on page 59 for more information.

Step 1: Configure the 2210 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 2210 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 59 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 you assign must be the same as the number used when configuring the device 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]?
ASRT config>
```

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

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

If BAN type 2 connections are being used, enable DLSw for bridging.

```
ASRT config>enable dls
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 21 on page 54 and Figure 22 on page 55). 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 2210 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: Configure the Router for DLSw (BAN Type 2 Only)

If BAN type 2 connections are being used, then DLSw must be configured. This involves enabling DLSw, setting the DLSw segment number, adding the local DLSw TCP partner, and opening the service access points (SAPs) associated with the FR

interface and the LAN interface. If you fail to perform this DLSw configuration, you will not be able to use BAN type 2 (DLS terminated) connections.

Enable DLSw, using the **enable dls** command from the `DLSw config>` prompt.

Set the DLSw segment number using the **set srb** command form the `DLSw config>` prompt.

To add a local DLSw TCP partner, do the following at the `DLSw config>` prompt:

```
DLSw config>add tcp
Enter the DLSw neighbor IP Address [0.0.0.0.]? 128.185.236.33
Neighbor Priority (H/M/L) [M]?
DLSw config>
```

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 23 on page 59.

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

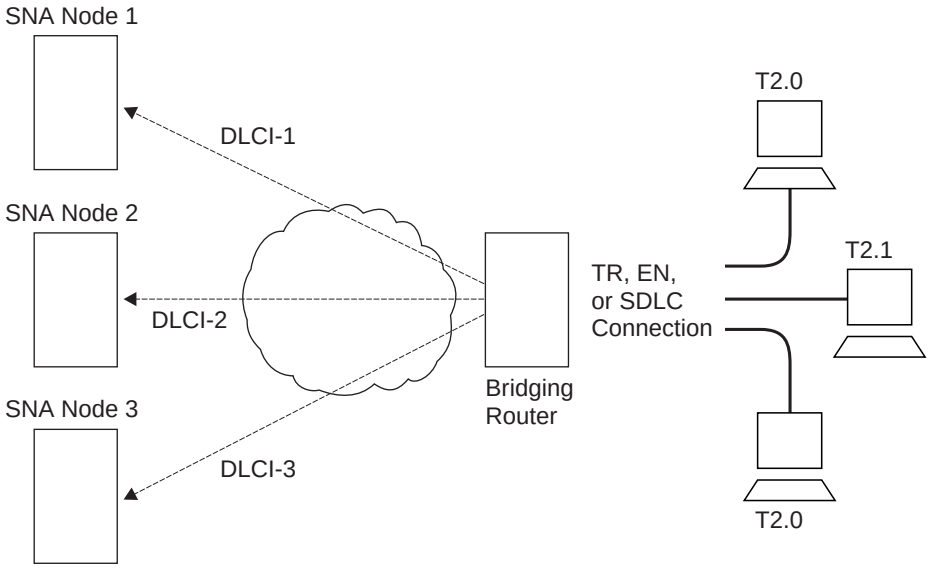


Figure 23. BAN Configuration with Multiple DLCIs to Different SNA Nodes

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 58), 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 57.

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. Using Bridging

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 “Basic Bridging Configuration Procedures”.

If you need more information about the ASRT bridge configuration commands, refer to the “Chapter 6. Configuring and Monitoring Bridging” on page 67.

For an introduction to modification of ASRT bridging, see “NetBIOS Name and Byte Filters” on page 42.

For examples of setting up NetBIOS filtering, see “NetBIOS Host Name and Byte Filtering Configuration Procedures” on page 147.

For information on how to access the ASRT configuration environment, see “Getting Started” in the *Software User’s Guide*.

Basic Bridging Configuration Procedures

The ASRT bridge enables 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 correctly 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 ASRT bridge supports bridging over combinations of one or more of the following interfaces:

- Ethernet
- Token-Ring
- Serial Line
- ATM

The Ethernet interface supports transparent bridging, while token-ring interfaces 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.

Using Bridging

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 are 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-route 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 you require, disable transparent bridging on the interfaces.

Note: You should be careful ***not*** to 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 bridging domain.

Using Bridging

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 6. Configuring and Monitoring Bridging” on page 67.

Chapter 6. Configuring and Monitoring 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

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

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, ATM interface features, and NetBIOS.

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

Note: The ASRT configuration commands are not effective immediately. They remain pending until you restart or reload the device.

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

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

ASRT Configuration Commands (Talk 6)

Table 4 shows the ASRT configuration commands.

Table 4. ASRT Configuration Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
Add	Adds station address entries to the permanent database, specific address mapping, LAN/WAN ports, multiaccess ports, protocol filters, duplicate MAC addresses, 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, duplicate MAC addresses, 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 • SR-TB conversion • Transparent (spanning tree) bridging function on a given port • Tunnel between bridges • Duplicate MAC address feature • Duplicate MAC load balancing • IPX Conversion
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 • SR-TB conversion • Transparent (spanning tree) bridging function on a given port • Tunnel between bridges • Duplicate MAC address feature • Duplicate MAC load balancing • IPX Conversion
List	Displays information about the complete bridge configuration or about selected configuration parameters.
NetBIOS	Displays the NetBIOS configuration prompt.

Table 4. ASRT Configuration Command Summary (continued)

Command	Function
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 • Aging value for duplicate MAC address Routing Information Fields (RIFs) • Aging value for multiaccess database entries • IPX Conversion Mode • Ethernet Preference
Tunnel	Allows access to the tunnel configuration prompt so that tunnel configuration commands can be entered.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Response to ASRT Configuration Commands

The ASRT configuration (Talk 6) commands are not effective immediately. They remain pending until you issue the **reload** or **restart** command.

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
- Multiaccess ports
- LAN/WAN ports
- Protocol filters that selectively filter packets based on their protocol type
- IP tunnel between end-stations and across IP network segments
- Up to 7 duplicate MAC addresses

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 . . .
dmac-addr
mapping . . .
multiaccess-port . . .
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

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

Adding an address

```
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 that 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. Answering *yes* maps the address to the specified port so that the port is included in that address entry's port map. Answering *no* 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 source address filtering (SAF). When SAF is applied (answer *yes* 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.

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
```

dmac-addr *addr-value*

Adds up to 7 duplicate MAC address entries to the database. The

addr-value is the MAC address of the desired entry. See “SR-TB Duplicate MAC Address Feature” on page 47 for additional information about the duplicate MAC address feature.

Valid Values: X’0000 0000 0000’ to X’FFFF FFFF FFFF’

Default Value: none

Example:

After adding the address, you can verify DMAC information by using the **list dmac** command.

```
ASRT config>add dmac-addr
Address (in 12-digit hex) []? 10005a777701


ASRT config>list dmac
Duplicate MAC address feature is    ENABLED
Load balance feature is    ENABLED
Age value for Duplicate MAC address :000000C8
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-02
10-00-5A-66-66-05
10-00-5A-77-77-01
```

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 conversely.

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 the range 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 the range 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) [ ]?
```

multiaccess-port *interface# port# segment# [bridge#] [virtual-segment#]*
Adds a multiaccess port to the bridging configuration. This command associates a port number with a Frame Relay interface and enables the port for source route bridging.

interface#
Specifies the Frame Relay interface on which you are configuring the multiaccess port.

Valid values: any existing Frame Relay interface number

Default value: 0

port# Specifies the bridge port number. This number must be unique among all configured bridge ports in the device.

Valid values: 1 to 254

Default value: next available port number

segment#

Specifies a 12-bit, hexadecimal, source routing segment number that represents the multiaccess segment. All bridges connected to the multiaccess segment must use the same segment number.

Valid values: X'001' to X'FFF'

Default value: X'001'

bridge#

Specifies a 4-bit, hexadecimal, source routing bridge number that represents this bridge on the multiaccess segment. This parameter is only required when enabling source routing for the first time. The bridge number should be unique among all bridges on the multiaccess segment.

Valid values: X'0' to X'F'

Default value: X'0'

virtual-segment#

Specifies a 12-bit, hexadecimal, source routing segment number. This parameter is only required when enabling source routing for the first time on more than two bridge ports or the first time you configure a multiaccess bridge port.

Valid values: X'001' to X'FFF'

Default value: X'001'

Example:

```
add multiaccess-port
Interface number [0]? 3
Port number [2]? 2
Segment number for the port in hex (1 - FFF) [001]? 200
Bridge number in hex (0-9, A-F) [0]? 1
Bridge Virtual Segment Number in hex (1-FFF) [001]? FFF
```

port *interface# port#*

Adds a LAN/WAN port 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: add a port

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

See "ATM Commands" on page 110 for information about adding ATM ports and "Frame Relay Commands" on page 109 for information about adding Frame Relay ports.

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:

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

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

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

Notes:

1. You cannot filter all SNAP format packets by adding a DSAP filter for type X'AA'. The encapsulated SNAP protocol(s) must be filtered individually. Consider using a sliding window filter. Refer to the chapter entitled “Using MAC Filtering” in *Using and Configuring Features*.
2. You cannot configure protocol filters for a protocol that is routed over a specific interface if the interface is also configured for bridging.

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:

ASRT config>**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. The bridge tunnel 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 device software encapsulate IBM traffic within industry-standard TCP/IP packets. The bridging device 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 device, 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 devices 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. You must use a *Port Number* that is not used for any other LAN port. Once a Port Number is assigned to the bridging tunnel, all other bridging commands that need a port number as a parameter can be used to configure the tunnel characteristics. For tunnel-specific configuration, such as the IP addresses of the endpoints, use the **tunnel** command (see “Tunnel” on page 104).

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 Configuration Commands” on page 104 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 number in the bridging configuration.

Example: `change bridge 3`

segment *old-segment# new-segment#*
Changes segment number 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 and multiaccess ports
- Protocol filters that selectively filter packets based on their protocol type
- Duplicate MAC addresses

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
	dmac-addr
	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`

dmac-addr *addr-value*
Deletes duplicate MAC address entries from the database. The *addr-value* is the MAC address of the desired entry you want to remove.

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

Default Value: none

Example:

```
ASRT>list gamic
Duplicate MAC address feature is   DISABLED
Load balance feature is   DISABLED
Age value for Duplicate MAC address :00000096
Duplicate MAC ADDRESSES CONFIGURED
================================
10-00-5A-66-66-00
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
```

```
10-00-5A-66-66-04
10-00-5A-66-66-05

ASRT config>delete dmac-address
Address (in 12-digit hex) []? 10005a666600
Address deleted

ASRT config>list dmac
Duplicate MAC address feature is   DISABLED
Load balance feature is   DISABLED
Age value for Duplicate MAC address :00000096
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05
```

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 the range 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 the range 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
- Duplicate frames
- Mapping between group and functional addresses
- Propagation of Spanning Tree Explorer Frames
- Source routing on a given port
- SR-TB conversion
- Transparent (spanning tree) bridging function on a given port
- Duplicate MAC address feature
- Duplicate MAC load balancing
- DLSw

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

Syntax:

disable

bridge
dls
duplicate . . .
dmac-addr
dmac-load-balance
ethertype-ibmrt-pc
fa-ga-mapping
ibm8209-spanning-tree
ipx-conversion . . .
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 device running DLSw appears as a bridge to the end stations.) See “Chapter 25. Using DLSw” on page 481 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. The bridge does not know 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]?
```

dmac-addr

Disables the duplicate MAC address feature.

Example: `disable dmac-addr`

```
ASRT>list dmac
Duplicate MAC address feature is    ENABLED
Load balance feature is    ENABLED
Age value for Duplicate MAC address :00000096
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-00
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05

ASRT config>disable dmac-addr

ASRT>list dmac
Duplicate MAC address feature is   DISABLED
Load balance feature is   DISABLED
Age value for Duplicate MAC address :00000096
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-00
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05
```

dmac-load-balance

Disables Duplicate MAC load balancing for the duplicate MAC address feature.

Example: `disable dmac-load-balance`

```
ASRT>list dmac
Duplicate MAC address feature is    ENABLED
Load balance feature is    ENABLED
Age value for Duplicate MAC address :00000096
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-00
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05
ASRT config>disable dmac-load-balance
ASRT>list dmac
Duplicate MAC address feature is   ENABLED
Load balance feature is   DISABLED
Age value for Duplicate MAC address :00000096
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-00
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05
```

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 conversely) 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`

ipx-conversion
Globally disables conversion of Novell IPX frames when bridging between Ethernet/802.3 and Token-Ring (802.5) bridge ports. When disabled, Novell IPX frames can be bridged between end stations on the same media type through a backbone LAN of a different media type, but cannot be bridged between end stations on different media.

Example: `disable ipx-conversion`

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 correctly. 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
- Duplicate frames
- Mapping between group and functional addresses
- Propagation of Spanning Tree Explorer Frames
- Source routing on a given port
- SR-TB conversion
- Transparent (Spanning Tree) bridging function on a given port
- Duplicate MAC address feature
- Duplicate MAC load balancing
- DLSw

Syntax:

enable	bridge . . .
	dls
	duplicate
	dmac-addr
	dmac-load-balance
	ethertype-ibmrt-pc
	fa-ga-mapping
	ibm8209-spanning-tree
	ipx-conversion . . .
	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 device. 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 device running DLSw appears as a bridge to the end stations. See “Chapter 25. Using DLSw” on page 481 from more information.

Example: **`enable dls`**

duplicate *frame-type*
Enables the generation of duplicate STE (spanning tree explorer) frames or TSFs (transparent spanning frames). This command is available to offset the **disable duplicate** command. Duplicate frame generation is enabled by default. The **enable duplicate** command can 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]?
```

dmac-addr
Enables the duplicate MAC address feature. See “SR-TB Duplicate MAC Address Feature” on page 47 for additional information about the duplicate MAC address feature.

Example with load-balancing:

```
ASRT config>enable dmac-addr

ASRT config>list dmac
Duplicate MAC address feature is    ENABLED
Load balance feature is   DISABLED
Age value for Duplicate MAC address :000000C8
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05

ASRT config>enable dmac-load-balance

ASRT config>li dmac
Duplicate MAC address feature is    ENABLED
Load balance feature is    ENABLED
Age value for Duplicate MAC address :000000C8
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05
```

Example (without load-balancing):

```
ASRT config>enable dmac-addr

ASRT config>list dmac
Duplicate MAC address feature is    ENABLED
Load balance feature is   DISABLED
Age value for Duplicate MAC address :000000C8
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05
```

dmac-load-balance
Enables Duplicate MAC load balancing for the duplicate MAC address

feature. See the discussion “SR-TB Duplicate MAC Address Feature” on page 47 for a description of Duplicate MAC load balancing.

Example :

```
ASRT config>enable dmac-addr

ASRT config>list dmac
Duplicate MAC address feature is    ENABLED
Load balance feature is   DISABLED
Age value for Duplicate MAC address :000000C8
Duplicate MAC ADDRESSES CONFIGURED
================================
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05

ASRT config>enable dmac-load-balance

ASRT config>li dmac
Duplicate MAC address feature is    ENABLED
Load balance feature is    ENABLED
Age value for Duplicate MAC address :000000C8
Duplicate MAC ADDRESSES CONFIGURED
================================
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05
```

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 conversely) 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`**

ipx-conversion

Globally enables conversion of Novell IPX frames when bridging between Ethernet/802.3 and Token-Ring (802.5) bridge ports. When enabled, Novell frames can be bridged between end stations on different media.

Example: **`enable ipx-conversion`**

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 required. 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 are 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

This option enables conversion of source routing to transparent bridging frame format and vice versa. It 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 correctly, you must give the transparent bridging domain 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 a 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
- dmac
- filtering . . .
- mapping . . .
- multiaccess
- 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 required entry. It can be an individual address, multicast address, or broadcast address. Permanent databases 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:        Default          Bridge Priority:     32768/0x8000
SRB Bridge Address:    Default          SRB 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
IPX Conversion:        Enabled
Conversion Mode:       Automatic
Ethernet Preference:   IEEE-802.3
                 +-------------------+
-----------------| 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
 Circuit number: 16
Port:   3      Interface:     2      Behavior:   STB & SRB   STB:  Enabled
Circuit number: 18
```

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

Segments
Indicates the number of source-routing bridge segments configured for the source-routing domain.

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

1:N SRB
Indicates the current state of 1:N Source Routing as ACTIVE or NOT ACTIVE.

Internal Segment
Displays the virtual segment number configured for 1:N SRB bridging.

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.

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.

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.

STP Participation
Displays the types of spanning tree protocols in which the bridge participates.

FA-GA conversion
Indicates whether FA-GA conversion is enabled or disabled.

UB Encapsulation
Indicates whether UB Encapsulation is enabled or disabled.

DLS for the bridge
Indicates if the Data Link Switch protocol is enabled or disabled in the bridge.

IPX Conversion
Indicates whether IPX Conversion is enabled or disabled.

Conversion Mode
Indicates the IPX Conversion Mode as either automatic or manual.

Ethernet Preference
Indicates the preferred Ethernet frame type used for IPX Conversion as IEEE-802.3 or Ethernet.

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, STB for transparent bridging and SRB for source route bridging.

VPI Specifies the VPI associated with the ATM port.

VCI Specifies the VCI associated with the ATM port.

Circuit Number
Specifies the DLCI associated with the Frame Relay port.

dmac Displays the configured options for the duplicate MAC address feature.

Example: `list dmac`

```
Duplicate MAC address feature is    ENABLED
Load balance feature is   DISABLED
Age value for Duplicate MAC address :000000C8
Duplicate MAC ADDRESSES CONFIGURED
```

```
=================================
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05
```

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:

Routed
Describes packets passed to routing forwarder to be forwarded.

Filtered
Describes packets that are administratively filtered setting protocol filters that you set.

Bridged and routed
This describes a protocol identifier for which there is a protocol entity within the system that 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 the range 1–FE (hexadecimal).
- Ethernet (Ether) protocol type is entered in the range of 5DD–FFFF (hexadecimal).
- Subnetwork Access Protocol (SNAP) protocol type is entered in 10-digit hexadecimal format.

multiaccess
Displays the aging time for entries in the multiaccess database and displays the multiaccess bridge ports. See the output of the **list port** command for a description of the bridge port parameters.

Example: `list multiaccess`

```
Aging time (in seconds): 300

Port ID (dec)    : 238:02, (hex): 80-02
Port State       : Enabled
STP Participation: Disabled
Port Supports    : Source Route Bridging Only
SRB: Segment Number: 0x003       MTU:  2040     STE: Enabled
Assoc Interface  : 1
Path Cost        : 0
```

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: 6, (hex): 80-06
Port State       : FORWARDING
STP Participation: Enabled
Port Supports: Source Routing Bridging Only
SRB: Segment Number:  Ox116     MTU:  1979
STE Forwarding:  Auto
Assoc Interface #/name : 1/FR/0   Circuit number 16
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Port Id (dec)    : 128: 7, (hex): 80-07
Port State       : FORWARDING
STP Participation: Enabled
Port Supports: Source Routing Bridging Only
SRB: Segment Number:  Ox117     MTU:  1979
STE Forwarding:  Auto
Assoc Interface #/name : 1/FR/0   Circuit number 17
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Port ID (dec)    : 128: 2, (hex): 80-02
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface  : 0  VPI  0  VCI:    78
Path Cost        : 0
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Port ID (dec)    : 128: 3, (hex): 80-03
```

```
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface  : 2
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 Port ID (dec)    : 128: 1, (hex): 80-01
 Port State       : Enabled
 STP Participation: Enabled
 Port Supports    : Transparent Bridging Only
 Assoc Interface  : 0  VPI:   0  VCI:   795
 Path Cost        : 0
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 Port ID (dec)    : 128: 4, (hex): 80-04
 Port State       : Enabled
 STP Participation: Enabled
 Port Supports    : Transparent Bridging Only
 Assoc Interface  : 0  Dest ATM Addr: 39112233445566778899001122334 4
                                                              5566778899
 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, and 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).

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. Also displays the VPI/VCI or the destination ATM address if the port exists on an ATM interface.

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 (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`

```
IEEE 802.1d Spanning Tree Configuration:
 Bridge Identifier                : 32768/000000000000 (using port address)
 Bridge-Max-Age (in seconds)      : 20
 Bridge-Hello-Time (in seconds)   : 2
 Bridge-Forward-Delay (in seconds): 15

SRB Spanning Tree Configuration:
 Bridge Identifier                : 32768/000000000000 (using port address)
 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 specify 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 you do not specify the stop index, 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 monitoring process that do not survive power on/offs or system resets and are ageless
- Dynamic - entries “learned” by the bridge “dynamically” that do not survive power on/offs or system resets and that 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 155 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:

- Aging time for dynamic address entries in the filtering database
- Bridge address
- IPX Conversion Mode and Ethernet Preference
- 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
- Aging time for RIFs associated with duplicate MAC addresses
- Aging time for entries in the multiaccess database

Syntax:

set

age

bridge

conversion-mode

ethernet-preference

dmac-age

filtering

lf-bit-interpretation . . .

maximum-packet-size . . .

multiaccess-age . . .

port

protocol bridge

protocol port . . .

route-descriptor-limit . . .

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 adaptive database in the case of an SR-TB bridge personality.

Enter the required value after each prompt and press **Return**.

Aging Time Valid Values: 10 to 1000000

Aging Time Default Value: 30

The resolution value specifies how often dynamic entries in the filtering database should be scanned to determine if they have exceeded 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 default address and enter your own unique address.

Enter *srb* or *tb* to specify whether the source routing bridge (srb) or transparent bridge (tb) bridge address is to be affected.

Note: Each bridge in the network must have a unique address for the spanning tree protocol to operate correctly.

Attention: 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 zeros.

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 correctly.

Default Value: 000000000000

Example: **`set bridge`**

```
Bridge Address (in 12-digit hex)[]?
```

conversion-mode *mode*

Specifies the IPX conversion mode as either automatic or manual.

mode When the conversion mode is set to automatic, the IPX frame type of each Ethernet/802.3 end-station is learned and stored in the filtering database and will be used on subsequent conversions to the Ethernet/802.3 MAC format. The Ethernet Preference parameter determines which IPX frame type to use when converting to the Ethernet /802.3 MAC format if one has not yet been learned.

When the Conversion Mode is set to manual, the value of the Ethernet Preference parameter specifies on which Ethernet/802.3 IPX frames the conversion is to be preformed.

Conversion-Mode Valid Values: automatic or manual

Conversion-Mode Default Value: automatic

Example: **`set conversion-mode manual`**

ethernet-preference *preference*

Specifies the preferred IPX frame type as either IEEE-802.3 or Ethernet.

preference

When the Conversion Mode is set to automatic, the IPX frame type of each Ethernet/802.3 end-station is learned and stored in the filtering database and will be used on subsequent conversions to the Ethernet/802.3 MAC format. The Ethernet Preference parameter

determines which IPX frame type to use when converting to the Ethernet /802.3 MAC format if one has not yet been learned.

When the Conversion Mode is set to manual, the value of the Ethernet Preference parameter specifies on which Ethernet/802.3 IPX frames the conversion is to be preformed.

Ethernet-Preference Valid Values: ieee-802.3 or ethernet

Ethernet-Preference Default Value: ieee-802.3

Example: `set ethernet-preference ethernet`

dmac-age *seconds*

Sets the time for aging out RIF entries in the RIF table for duplicate MAC addresses. This value will be used for only the learned duplicate MAC addresses. For all other addresses, the value from the **set age** command will be used for aging.

Enter the desired value after each prompt and press **Return**.

DMAC Aging Time Valid Values: 10 to 1000000

DMAC Aging Time Default Value: 300

Example: `set dmac-age`

```
seconds [300]? 200
ASRT config>list dmac
Duplicate MAC address feature is   DISABLED
Load balance feature is   DISABLED
Age value for Duplicate MAC address :000000C8
Duplicate MAC ADDRESSES CONFIGURED
==================================
10-00-5A-66-66-01
10-00-5A-66-66-02
10-00-5A-66-66-03
10-00-5A-66-66-04
10-00-5A-66-66-05
```

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 93.

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 device 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 16 to 65535

Default Value: packet size set for the port

Example: `set maximum-packet-size 1 4399`

multiaccess-age *seconds*

Sets the time for aging out entries in the multiaccess database. The database is scanned at the rate set by the *resolution* parameter of the **set age** command.

Valid values: 1 to 1 000 000

Default value: 300

Example: set multiaccess-age

```
seconds [300]? 500
```

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 device 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 device 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

Bridge Forward Delay

Time interval used before changing to another state (should this bridge become the root).

When this bridging device 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 used 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 device become the root bridge. Factor this value in when determining spanning tree routes between any two stations. A value of 0 instructs the bridging device 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. This is used 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*

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`

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. You will then be prompted for an RD-limit-value.

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. Tunnel configuration commands are entered at this prompt. See "Tunnel Configuration Commands" on page 105 for an explanation of each of these commands.

Syntax:

tunnel

BAN Configuration Commands

This section describes all of the BAN (boundary access node) configuration commands. These commands let you configure BAN as an added feature to ASRT bridging or to DLSw.

Note: The BAN configuration commands are not effective immediately. They remain pending until you restart or reload the device.

Configuration commands are entered at the `BAN config>` prompt. This prompt is accessed by entering the **ban** command at the `ASRT config>` or the `DLSw config` prompt. Table 5 shows the BAN configuration commands.

Table 5. BAN Configuration Commands

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
Add	Adds a BAN port.
Delete	Deletes a BAN port.
List	Displays all information concerning BAN ports.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

Response to BAN Configuration Commands

The BAN configuration (Talk 6) commands are not effective immediately. They remain pending until you issue the **reload** or **restart** command.

Add

Use the **add** command to add a BAN port to the BAN configuration. If a port number is not supplied with the command, you are prompted for the port number.

Syntax:

add *port#*

Example: `add`

```
Port Number [0]? 3.
Enter the BAN DLCI MAC Address []? 400012345678
Enter the Boundary Node Identifier MAC Address [4FFF00000000]?
Do you want the traffic bridged (b) or DLSw terminated (t) (b/t) [b]
```

Delete

Use the **delete** command to delete a BAN port from the BAN configuration. If a port number is not supplied with the command, you are prompted for the port number.

Syntax:

delete *port#*

Example: `delete 3`

List

Use the **list** command to list information about all BAN ports. The information that is displayed includes the BAN port number, the MAC address for the BAN DLCI, and whether the frames handled by the port are bridged or the LLC is terminated by DLSw.

Syntax: list

list

Example: `list`

```
bridge  BAN                Boundary           bridged or
port    DLCI MAC Address   Node Identifier    DLSw terminated
2       40:00:11:22:33:44  4F:FF:00:00:00:00  bridged
3       40:00:55:66:77:88  4F:FF:00:00:00:00  bridged
```

Tunnel Configuration Commands

This section describes the Tunnel configuration commands. The Tunnel configuration commands allow you to specify network parameters for a tunnel that transmits bridging frames over IP.

Note: The tunnel configuration commands are not effective immediately. You must restart or reload the device before they become effective.

Configuration commands for the tunnel are entered at the `TNL config>` prompt. This prompt is accessed by entering the **tunnel** command at the `ASRT config>` prompt. Table 6 shows the tunnel configuration commands.

Table 6. Tunnel Configuration Commands

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
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.

Table 6. Tunnel Configuration Commands (continued)

Command	Function
Join	Configures the device as a member of one or more multicast groups.
Leave	Removes the device 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 device.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Response to Tunnel Configuration Commands

The tunnel configuration (Talk 6) commands are not effective immediately. They remain pending until you issue the **reload** or **restart** command.

Tunneling and Multicast Packets

The bridging tunnel can be defined as either a unicast tunnel or a multicast tunnel. To define a unicast tunnel, use the **add** command to configure the IP address of the tunnel’s endpoint. To define a multicast tunnel, use the **set** and **join** commands. 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 some interfaces such as ATM, X.25 and Frame Relay. This means that when you define a multicast tunnel on the device (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 devices in the network do not like host addresses (this would happen in mixed vendor networks).

Add

Use the **add** command to add the IP address of end stations participating in a unicast 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 device 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.

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 device 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: 0 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: 0 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: 0 to 64

Default Value: 0

Example: `join server-group 7`

Leave

Use the **leave** command to remove the device 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: 0 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: 0 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: 0 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 device.

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).

Syntax:

set base-multicast-address

base-multicast-address
Sets the base multicast IP address for multicast tunneling.

Valid Values: any valid class D IP address with the last two bytes set to 0.

Default Value: 224.186.0.0

Example: `set base-multicast-address 224.10.0.0`

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. This is referred to as a Frame Relay point-to-point bridge port. You can also define a multiaccess bridge port associated with the Frame Relay interface itself. For further information, see “Configuring Multiaccess Bridge Ports” on page 49.

Once a bridge port is configured, all the functions associated with bridge ports, including protocol filtering and address filtering are available.

For each Frame Relay point-to-point bridge port, you must specify either PVC or SVC. For PVC support you must specify the associated DLCI number. For SVC support, you must provide the SVC circuit name.

Response to Frame Relay Configuration Commands

The Frame Relay configuration (Talk 6) commands are not effective immediately. They remain pending until you issue the **reload** or **restart** command.

At the `ASRT config>` prompt, use the following command to enable bridging for a Frame Relay circuit:

add port *interface# port# circuit-id*

interface#
The interface number of the Frame Relay interface.

port# The unique bridge-specific number associated with the circuit.

Valid Range: 1 to 254

Default Value: none

circuit-id
The DLCI number for the PVC on which bridging is being enabled or the circuit name of the SVC on which bridging is being enabled.

The command associates a port number with the Frame Relay PVC identified by the *circuit number* or the Frame Relay SVC identified by the circuit name and enables that circuit's participation in transparent bridging.

Example: add a port on a Frame Relay interface (PVC)

```
ASRT config> add port
Interface Number [0]? 5
Port Number [7]? 7
Use FR PVC? [Yes]: yes
Frame Relay Circuit number [16]? 17
```

Example: add a port on a Frame Relay interface (SVC)

```
ASRT config> add port
Interface Number [0]? 5
Port Number [8]? 8
Use FR PVC? [Yes]: no
Frame Relay SVC Circuit Name []? 05svc020
```

ATM Commands

To enable bridging over the ATM interface, you must associate a VCC with a bridge port.

Once a bridge port is configured, all the function associated with bridge ports, including protocol filtering and address filtering are available.

You need to specify PVC or SVC support. For PVC support, you must specify the VPI and VCI of the PVC. For SVC support, you must provide the remote ATM address and the local selector byte.

At the `ASRT config>` prompt, use the following command to enable bridging on the ATM interface:

add port *interface# port# VCC-id*

interface#
The interface number of the ATM interface.

port# The unique bridge-specific number associated with the VCC.

Valid Range: 1 to 254

Default Value: none

Once the port has been added on the ATM interface, the port number will identify the port to the ATM ARP client and to the VCC associated with this port.

See "Chapter 27. Using ARP" on page 571 for ATM ARP client configuration information.

vcc-id To define a PVC, provide the VPI and VCI information. To define a SVC, provide the destaddr and selector information.

VPI The VPI of the PVC on which bridging is enabled.

VPI Valid Values: 0 to 255

VPI Default Value: 0

VCI The VCI of the PVC on which bridging is enabled.

VCI Valid Values: 0 to 65535

VCI Default Value: 0

destaddr
The destination ATM address of the SVC.

Destination ATM address Valid Values: any valid 20-byte ATM address

Destination ATM address Default Value: none

selector
The selector of the destination ATM address of the SVC.

Selector Valid Values: X'00' – X'FF'

Selector Default Value: X'00'

Example: add a port on an ATM interface (PVC)

```
ASRT config> add port
Interface number [0]?
Port number [1]?
Use PVC? [Yes]:
VPI, Range 0..255 [0]? 0
VCI, Range 0..65535 [0]? 795
```

Example: add a port on an ATM interface (SVC)

```
ASRT config> add port
Interface number [0]?
Port number [2]?
Use PVC? [Yes]:No
Destination ATM Address []? 39112233445566778899001122334455667788 99
Selector, Range 00..FF [00]? 0A
ASRT config>
```

Accessing the ASRT Monitoring Environment

To access the ASRT monitoring environment, enter the **protocol asrt** command at the + (GWCON) prompt:

```
+protocol asrt
ASRT>
```

ASRT Monitoring Commands

This section describes the ASRT monitoring commands. These commands allow you to view and modify parameters from the active monitoring. Information you modify with the monitoring commands is reset to the SRAM configuration when you restart the bridging device.

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 monitoring commands.

Monitoring commands for NetBIOS are entered at the `NetBIOS>` monitoring 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>` monitoring prompt. The NetBIOS-filtering prompt is a subset of the major ASRT commands.

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 7 shows the ASRT monitoring commands.

Table 7. ASRT Monitoring Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
Add	Adds permanent (static) address entries to the bridging device’s permanent database.
BAN	Allows you to access the boundary access node (BAN) monitoring prompt for entering specific BAN monitoring commands. See Table 8 on page 129 for a detailed description.
Cache	Displays cache entries for a specified port.
Delete	Deletes MAC addresses entries from the bridging device 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.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Add

Use the **add** command to add static address entries and destination address filters to the bridging device’s database. These additions to the database are lost when you restart the device.

Syntax:

add destination-address-filter
static-entry

destination-address-filter *mac_address*
Adds a destination address filter to the bridging device’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 device’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) monitoring prompt. Enter the **ban** command from the `ASRT>` prompt.

Syntax: ban

Example: `ASRT>`**`ban`**

```
BAN>
```

Once you access the BAN monitoring prompt, you can begin entering specific monitoring 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 IEEE 802.1d 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 monitoring 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 device'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 monitoring 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 device configuration or to display information about selected configuration or bridging options.

Syntax:

list
- adaptive . . .
- bridge . . .
- conversion . . .
- database . . .
- dmac
- filtering . . .
- multiaccess-database . . .
- port
- source-routing . . .
- spanning-tree-protocol . . .
- transparent . . .
- tunnel . . .

adaptive *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**. 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.

Example: `list adaptive 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
   -  002      Adaptive   Enabled   1470
```

Adaptive bridge
Shows the current state of the SR-TB adaptive 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.Also, displays the VPI/VCI if an ATM port and the DCI if a Frame Relay 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 counters
Hash collision count: 28
Adaptive database entry count: 0
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 Entry Count
Displays the number of entries currently stored in the adaptive bridge database.

Adaptive Database Overflow Count
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** command lets you 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 database** command options.

Note: These are only displayed if adaptive bridging is enabled.

Example: `list adaptive database address` ***mac-address***

Example: `list adaptive database all`

Example: `list adaptive database port` ***segment#***

Example: `list adaptive 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.

adaptive database duplicate
Lists database entry of all duplicate MAC addresses. It displays primary and secondary RIFs for each duplicate MAC address.

Example: `list adaptive database duplicate`

```
Canonical Address  Interface  Port  Seg    Age  RIF: Type Direct   Length  LF IBM MAC Address    RIF

08-00-5a-ee-ee-ee  TKR/0         3  001    180       SRF  Forward  14  1470 90:00:5a:77:77:77  0e10fef0dcab001b960395029001  PRI. RIF(3)

                   TKR/2         5  003    185       SRF  Reverse  14  1470                    0c9070087109003bdcabfef00000  SEC. RIF(3)
```

bridge
Lists all general information regarding the bridge device configuration.

Example: `list bridge`

```
Bridge ID (prio/add):  32768/10-00-5A-63-01-00
Bridge state:          Enabled
UB-Encapsulation:      Disabled
Bridge type:           STB
Number of ports:       2
STP Participation:     IEEE802.1d
IPX Conversion:        Enabled
Conversion Mode:       Automatic
Ethernet Preference:   IEEE-802.3
**Bridge is enabled for Data Link Switching**

                                                        Maximum
Port  Interface      State  MAC Address         Modes   MSDU   Segment Flags

   1  Eth /0            Up  10-00-5A-63-01-00       T    1514          RD
   2  FR /0:16          Up  00-00-00-00-00-00     SRT    2038     001  RD
   2  FR /0:18          UP  00-00-00-00-00-00     SR     2038     002  RD

Flags: RE = IBMRT PC behavior Enabled, RD = IBMRT PC behavior Disabled


SR bridge number:      00a
SR virtual segment:    ff6 (1 : N SRB Active)
Adaptive segment:      107
```

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.

UB-Encapsulation
Indicates whether UB Encapsulation 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.

STP Participation
Describes the types of spanning tree protocol in which a bridge participates.

IPX Conversion
Indicates whether IPX Conversion is enabled or disabled.

Conversion Mode
Indicates the IPX Conversion Mode as either automatic or manual.

Ethernet Preference
Indicates the preferred Ethernet frame type used for IPX Conversion as IEEE 802.3 or Ethernet.

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 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 (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
03-00-00-00-80-00*  Reserved              All
08-00-17-00-35-F9   Dynamic/ETH-II        300  1 (Eth/1)
08-00-17-00-4D-DA   Dynamic               300  1 (Eth/1)
```

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
```

Example: `list database dynamic`

```
MAC Address    MC* Entry Type      AGE  Port(s)
00-00-00-00-AA-AA  Dynamic         295  4 (Eth/2)
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)
08-00-17-00-35-F9  Dynamic/ETH-II  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 IEEE 802.1d 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 monitoring 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

If IPX Conversion is enabled and the entry was “learned” during the process of forwarding a Novell IPX frame, the Ethernet/802.3 frame type (encapsulated) is also displayed as:

- **ETH-II** - Ethernet-V2.0 (IPX frame type ETHERNET_II)
- **802.3** - 802.3/Novell Proprietary (IPX frame type ETHERNET_8023)
- **802.2** - 802.3/LLC (IPX frame type ETHERNET_8022)
- **SNAP** - 802.3/SNAP (IPX frame type ETHERNET_SNAP)

Free This type is not used and should not be normally be seen except in occasional “race” conditions between the monitoring 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.

dmac Displays information about configured options for the duplicate MAC address feature.

Example: `list dmac`

```
ASRT>list dmac
          Duplicate MAC address feature is    ENABLED
          Load balance feature is    ENABLED
          Age value for Duplicate MAC address :00000096
          Duplicate MAC ADDRESSES CONFIGURED
          ==================================
          10-00-5A-66-66-00
          10-00-5A-66-66-01
          10-00-5A-66-66-02
          10-00-5A-66-66-03
          10-00-5A-66-66-04
          10-00-5A-66-66-05
```

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
```

multiaccess-database *port#*

Displays the contents of the multiaccess database. This database maps a source routing segment number to a Frame Relay circuit number.

all-ports

Specifies that all database entries should be displayed.

Example: list multiaccess-database

```
Aging Time (in seconds): 300

4 entries used out of 512

Segment  Age    Port  Interface   Circuit
204      100    2     FR/0        16
267      200    3     FR/1        16
375      120    2     FR/0        18
400      220    3     FR/1        18
```

port *port#*

Displays database entries for a specific bridge port.

Example: list multiaccess-database

```
Aging Time (in seconds): 300

4 entries used out of 512

Segment  Age    Port  Interface   Circuit
204      100    2     FR/0        16
375      120    2     FR/0        18
```

In the displays:

Segment

Is the destination source routing segment number.

Age Is the entry time-to-live in seconds.

Port Is the port number of the multiaccess bridge port that built this entry.

Interface

Is the name of the network interface that built this entry.

Circuit

Is the Frame Relay circuit number that built this entry.

port *port#*

Displays port information.

Example: list port

```
Port Id (dec)    : 128: 3, (hex): 80-03
Port State       : Forwarding
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface #/name : 5/Eth/1
```

Example: list port 1

```
Port Id (dec)    : 128: 4, (hex): 80-04
Port State       : Disabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface #/name : 1/FR/0  PVC Circuit name:
  PVC Circuit number: 16
```

Example: list port 2

```
Port Id (dec)    : 128: 5, (hex): 80-05
Port State       : Disabled
STP Participation: Enabled
Port Supports    : Source Route Bridging Only
SRB: Segment Number: 0x004       MTU:  1979     STE Forwarding: Auto
Assoc Interface #/name : 10/PPP/1
```

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 bridge can transmit and receive on this interface.

Segment
Displays the source routing bridge segment number assigned to that port (if any).

source-routing *datagroup-option*
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 ports associated with source routing bridging.

Interface
The associated interface names. VPI/VCI is shown for ATM. DLCI is shown 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:                    0
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.

SRF Frames Received/Sent
Lists the number of Specifically Routed Frames received and sent on this bridge.

STE Frames Received/Sent
Lists the number of Spanning Tree Explorer Frames received and sent on this bridge.

ARE Frames Received/Sent
Lists the number of All Routes Explorer Frames received and 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 overflow
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 on 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 on all active ports.)

Dropped, MTU exceeded:
The number of frames that were discarded by this port due to an excessive size.

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 *datagroup-option*

- 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  ETH/1  32768/12-34-56-78-90-12    0 32768/12-34-56-78-90-12 90-01
2  ATM/0:0:48  0/00-00-00-00-00-00    0     0/00-00-23-45-00-00 80-00
```

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 **list transparent all-ports** to display the counters for all ports or enter the specific port number after the **list transparent** 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
```

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 *datagroup-option*

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.

NetBIOS

Use the **netbios** command to access the `NetBIOS>` prompt. NetBIOS monitoring commands may be entered at the `NetBIOS>` prompt.

See “NetBIOS Commands” on page 155 for the NetBIOS monitoring commands.

Syntax:

netbios

Accessing the BAN Monitoring Prompt

Use the **ban** command from the `ASRT>` or `DLSw>` monitoring prompt to access BAN commands.

To access the BAN monitoring prompt, enter the **ban** command from the ASRT monitoring prompt of the DLSw monitoring prompt. For example:

```
ASRT> ban
BAN>

   or

DLSw> ban
BAN>
```

Once you access the BAN monitoring prompt, you can begin entering specific monitoring commands. To return to the monitoring prompt you came from, enter the **exit** command.

BAN Monitoring Commands

This section describes the BAN monitoring commands. Enter the commands at the `BAN>` prompt.

Table 8. BAN Monitoring Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
List	Displays all information concerning BAN ports.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

List

Use the **list** command 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
```

ASRT Bridge Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

ASRT Bridge supports the CONFIG (Talk 6) **delete interface** command with no restrictions.

GWCON (Talk 5) Activate Interface

ASRT Bridge supports the GWCON (Talk 5) **activate interface** command with the following considerations:

- The interface cannot be activated if it is configured as a bridge port and the bridging protocol was not already active.
- The interface cannot be activated if it is configured as a bridge port and the addition of this port causes a change to the bridge personality. For example, you may not add a source routed (SR) bridge port to a bridge configuration that has only transparent (TB) bridge ports defined. Refer to “ASRT Configuration Matrix” on page 37 in “Chapter 2. Bridging Methods” on page 11 for information on bridge personality.
- The interface cannot be activated if it is configured as a bridge port and the addition of this port causes a change to the bridge internal virtual segment. For example, the interface cannot be activated if source routing was inactive and the addition of the newly activated port causes it to become active, requiring the use of the internal virtual circuit.
- The interface cannot be activated if it is configured as a bridge port and NetBIOS name or byte filters are configured for the port.
- The interface cannot be activated if it is configured as a bridge port and LNM is configured for the port.

All ASRT Bridge interface-specific commands are supported by the GWCON (Talk 5) **activate interface** command.

GWCON (Talk 5) Reset Interface

ASRT Bridge supports the GWCON (Talk 5) **reset interface** command with the following considerations:

- The interface cannot be reset if it is newly configured as a bridge port and the bridging protocol was not already active.
- The interface cannot be reset if the bridge configuration for this interface has changed such that the addition or deletion of the port causes a change to the bridge personality. For example, you may not add a source routed (SR) bridge port to a bridge configuration that has only transparent (TB) bridge ports defined. Refer to “ASRT Configuration Matrix” on page 37 in “Chapter 2. Bridging Methods” on page 11 for information on bridge personality.
- The interface cannot be reset if the bridge configuration for this interface has changed such that the addition or deletion of the port causes a change to the usage or the value of the bridge internal virtual segment.
- The interface cannot be reset if the port is configured for bridging and the port is one port of a two-port bridge.
- The interface cannot be reset if the bridge configuration for this interface has changed and NetBIOS name or byte filters are configured for the port being added or deleted.

- The interface cannot be reset if the bridge configuration for this interface has changed and LNM is configured for the port being added or deleted.

All ASRT Bridge interface-specific commands are supported by the GWCON (Talk 5) **reset interface** command.

CONFIG (Talk 6) Immediate Change Commands

ASRT Bridge supports the following CONFIG commands that immediately change the operational state of the device. These changes are saved and are preserved if the device is reloaded, restarted, or you execute a dynamically reconfigurable command.

Commands
CONFIG, protocol asrt, enable dmac-addr
CONFIG, protocol asrt, enable dmac_load-balance
CONFIG, protocol asrt, disable dmac-addr
CONFIG, protocol asrt, disable dmac_load-balance

BAN Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

Boundary Access Node (BAN) supports the CONFIG (Talk 6) **delete interface** command with no restrictions.

GWCON (Talk 5) Activate Interface

Boundary Access Node (BAN) supports the GWCON (Talk 5) **activate interface** command with the following consideration:

SRT bridging must be globally enabled before BAN can be activated for a bridge port (interface).

All Boundary Access Node (BAN) interface-specific commands are supported by the GWCON (Talk 5) **activate interface** command.

GWCON (Talk 5) Reset Interface

Boundary Access Node (BAN) supports the GWCON (Talk 5) **reset interface** command with the following consideration:

- SRT bridging must be globally enabled before BAN can be reset for a bridge port (interface).

All Boundary Access Node (BAN) interface-specific commands are supported by the GWCON (Talk 5) **reset interface** command.

Non-Dynamically Reconfigurable Commands

All Boundary Access Node (BAN) configuration parameters can be changed dynamically.

Chapter 7. Using NetBIOS

This chapter describes IBM's implementation of NetBIOS over bridged networks and over DLSw networks. It includes the following topics:

About NetBIOS

The NetBIOS protocol was designed for use on a Token-Ring LAN. 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) can 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.

It does this 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, the server 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 that 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 are found, the NetBIOS client can 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 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 lists the filters that IBM provides.

Table 9. 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 43 and “NetBIOS host-name Filtering” on page 42 describe byte and name filtering, respectively. The *Software User’s Guide* describes MAC address filtering.

For an introduction to host-name filtering and byte filtering, see “NetBIOS Name and Byte Filters” on page 42.

The following sections describe frame type, duplicate frame, response frame filtering, NetBIOS name lists, NetBIOS name and route caching.

Frame Type Filtering

Frame type filtering allows certain categories of NetBIOS frames to be filtered entirely for bridge traffic, DLSw 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 that 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. In the following explanation, these frames are called *NetBIOS command* frames and the possible response frames are called *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.

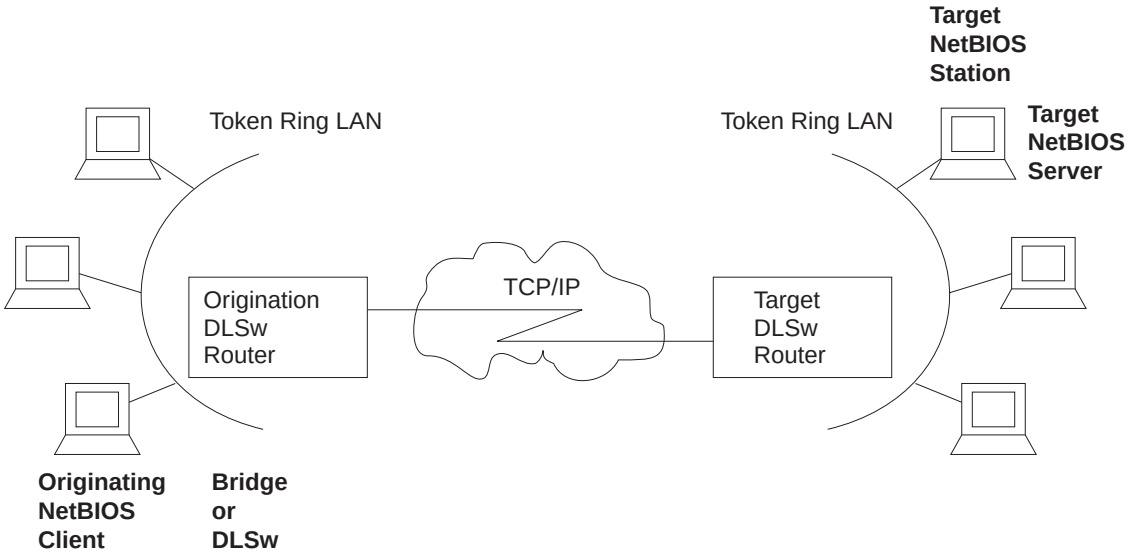


Figure 24. Setting Up a NetBIOS Session Over DLSw. Duplicate filtering reduces the number of broadcast frames forwarded over the DLSw WAN.

Figure 24, together with the following sequence, shows how the process works, using the default values. To simplify things, it is assumed that no response frame is received.

```
            Origin        Originating                          Target          Target
            NB stn        Router                               Router          NB stn
                          Bridge  DLSw                         DLSw  Bridge
BEGIN
TIME            NB cmd frame (1)                                     NB cmd frame (1)
                ────────────────────┐to dlsw──────────────────────────────────────►
 |                ◄──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)                  .5 second (command frame retry timer)
 |                     (reduced search timer)
 |
 |              NB cmd frame (4)                                     NB cmd frame (4)
 |              ────────────────────┐to dlsw──────────────────────────────────────►
 |                ◄──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)             2.5 seconds (dup frame detect timer)
                 deleted NB cmd frame                          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.

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.

Using NetBIOS

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 155 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 10.

Table 10. 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 11.

Table 11. 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 155.

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 to create 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. Configure host-name filter items 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 “NetBIOS Commands” on page 155 for syntax).

     Note: 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 because 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. Configure host-name filters 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 155 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, because 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. Configure host-name filters 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 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 153.

   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
```

Chapter 8. Configuring and Monitoring NetBIOS

This chapter describes IBM's configuring and monitoring of NetBIOS over bridged networks and over DLSw networks. It includes the following topics:

About NetBIOS Configuration and Monitoring Commands

NetBIOS configuration commands are available at the `ASRT/DLSW config>` prompt. Changes you make to the router's configuration are not effective 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>`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.

Note: The NetBIOS configuration commands are not effective immediately. You must restart or reload the device before they become effective.

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 Monitoring Environment

You can display the `NetBIOS>` prompt from either the ASRT monitoring environment or the DLSw monitoring environment.

Changes you make at the `NetBIOS>` monitoring prompt affect both bridging and DLSw.

To display the `NetBIOS>` monitoring 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
Default priority for SNA DLSw session traffic (C/H/M/L) [M]?  C
Default priority for NetBIOS DLSw session traffic (C/H/M/L) [M]? L
Default priority for SNA DLSw explorer traffic (C/H/M/L) [M]?  H
Default priority for NetBIOS DLSw explorer traffic (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]? 516
```

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 12 lists the NetBIOS configuration and monitoring commands.

Table 12. NetBIOS Configuration and Monitoring Commands

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
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.

Table 12. NetBIOS Configuration and Monitoring Commands (continued)

Command	Function
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Response to NetBIOS Configuration Commands

The NetBIOS configuration (Talk 6) commands are not effective immediately. They remain pending until you issue the **reload** or **restart** command.

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)
Enter NetBIOS name[]? 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)
  Enter NetBIOS name []? 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. 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 the 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 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 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 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
```

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 100 000 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 30 000. You can lower this value to save router memory. To optimize memory usage, processor usage, and the amount of broadcast traffic, set the number of local

name cache entries 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 30 000. You can lower this value to save router memory. To optimize memory usage, processor usage, and the amount of broadcast traffic, set the number of remote name cache entries 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 9. Configuring and Monitoring NetBIOS Filtering" on page 175.

From the `NetBIOS >` monitoring prompt, displays the NetBIOS filtering monitoring prompt (`NetBIOS Filter>`). Monitoring NetBIOS filtering is explained in "Monitoring NetBIOS Filtering" on page 184.

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 9. Configuring and Monitoring NetBIOS Filtering" on page 175.

From the `NetBIOS >` monitoring prompt, displays the NetBIOS filtering monitoring prompt (`NetBIOS Filter>`). Monitoring NetBIOS filtering is explained in "Monitoring NetBIOS Filtering" on page 184.

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 136 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 during which 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.

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
```

NetBIOS Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

NetBIOS does not support the CONFIG (Talk 6) **delete interface** command.

GWCON (Talk 5) Activate Interface

The GWCON (Talk 5) **activate interface** command is not applicable for NetBIOS. There are no interface-specific NetBIOS parameters.

GWCON (Talk 5) Reset Interface

The GWCON (Talk 5) **reset interface** command is not applicable for NetBIOS. There are no interface-specific NetBIOS parameters.

GWCON (Talk 5) Temporary Change Commands

NetBIOS supports the following GWCON commands that temporarily change the operational state of the device. These changes are lost whenever the device is reloaded, restarted, or you execute any dynamically reconfigurable command.

The following commands do not affect any NetBIOS functions already active at the time of the command unless otherwise noted. The changes do apply to all subsequent NetBIOS operations and traffic.

Commands
GWCON, protocol asrt, netbios, add cache-entry
GWCON, protocol asrt, netbios, add name-list
GWCON, protocol asrt, netbios, delete cache-entry
GWCON, protocol asrt, netbios, delete name-list
GWCON, protocol asrt, netbios, disable duplicate-filtering
GWCON, protocol asrt, netbios, disable name-list local
GWCON, protocol asrt, netbios, disable name-list remote
GWCON, protocol asrt, netbios, disable route-caching
GWCON, protocol asrt, netbios, enable duplicate-filtering
GWCON, protocol asrt, netbios, enable name-list local
GWCON, protocol asrt, netbios, enable name-list remote
GWCON, protocol asrt, netbios, enable route-caching
GWCON, protocol asrt, netbios, set cache-parms
GWCON, protocol asrt, netbios, set filters bridge
GWCON, protocol asrt, netbios, set filters dlsw
GWCON, protocol asrt, netbios, set general
GWCON, protocol asrt, netbios, set name-list

Non-Dynamically Reconfigurable Commands

All NetBIOS configuration parameters can be changed dynamically.

Chapter 9. Configuring and Monitoring NetBIOS Filtering

This chapter describes 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.

It includes 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 13 shows the NetBIOS filtering configuration commands.

NetBIOS Filtering Configuration Commands

Note: The NetBIOS filtering configuration commands are not effective immediately. You must restart or reload the device before they become effective.

Table 13. NetBIOS Filtering Configuration Commands

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
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.
Update	Adds information to or deletes information from a host-name or byte filter list.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

Response to NetBIOS Configuration Commands

The NetBIOS configuration (Talk 6) commands are not effective immediately. They remain pending until you issue the **reload** or **restart** command.

Create

Use the **create** command to create a byte filter-list 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 using 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-lists 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 using the create command. Enter an optional operator as either AND or OR 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-lists 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 using 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 under "**Update Byte-Filter** Command Options".

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 using 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 under "**Update Name-Filter** (Command Options)".

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 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 digits.
- 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 digits.
- 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 except the following:

  . / \ [ ] : | < > + = ; , <space>

  Note: Host-name is case sensitive.
- 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 the 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 except 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.

Monitoring NetBIOS Filtering

This section describes the NetBIOS Filtering monitoring commands. These commands let you monitor and display NetBIOS Filter information as an added feature to ASRT bridging. Monitoring commands are entered at the `NetBIOS >` monitoring prompt.

Changes you make at the `NetBIOS>` monitoring prompt affect both bridging and DLSw.

Accessing the ASRT and the DLSw NetBIOS Filtering monitoring Environments

To display the `NetBIOS>` monitoring prompt from the ASRT monitoring environment, enter the **netbios** command at the ASRT> prompt:

```
+ protocol asrt

ASRT> netbios
NetBIOS Support User monitoring

NetBIOS monitoring> set filters name or byte

NetBIOS filter>
```

To display the `NetBIOS>` monitoring prompt from the DLSw monitoring environment:

```
+ protocol dls
DLSw> netbios
NetBIOS Support User monitoring

NetBIOS Console> set filters name or byte
NetBIOS filtering

NetBIOS filter>
```

NetBIOS Filtering Monitoring Commands

Table 14 lists the NetBIOS filtering commands.

Table 14. NetBIOS Filtering Monitoring Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
List	Displays all information concerning created filters.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

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.

Chapter 10. Using LAN Network Manager (LNM)

This chapter describes IBM's ASRT LAN Network Manager (LNM). It includes the following sections:

- "About LNM"
- "LNM Agents and Functions"
- "LNM Configuration Restrictions" on page 190

About LNM

Use LNM to 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 25 on page 188 illustrates the connection between the IBM bridge, LNM agents, and the IBM LNM station.

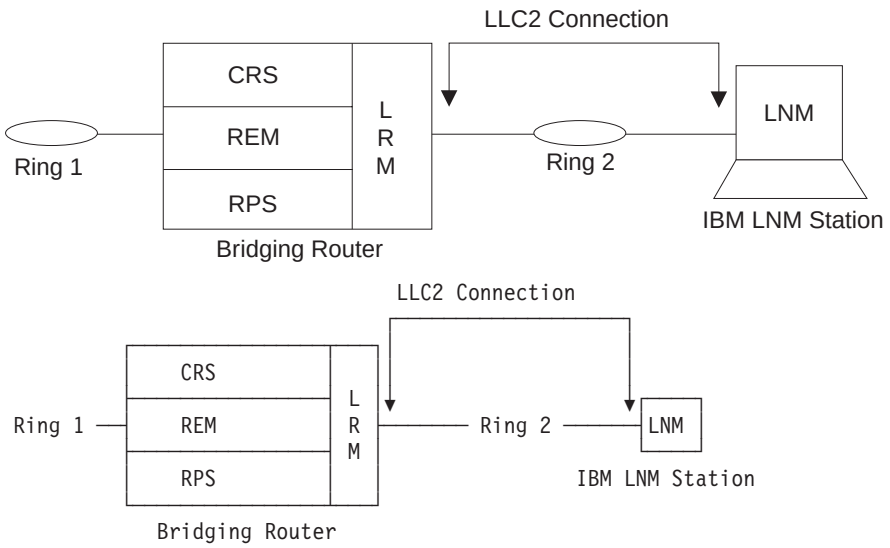


Figure 25. LNM Station and Agents

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. 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.

Attention: When a station attempts to insert into a ring, it sends a Request Initialization MAC frame to the Ring Parameter Server (RPS) for that ring. If this frame is copied successfully by the RPS, then the station expects to receive an Initialize Ring Station MAC frame back from the RPS. If no such frame is received, the station will not insert into the ring.

A station may fail to insert into the ring if the device is configured for LNM, becomes the Ring Parameter Server, and enters a congested state that prevents the sending of the Initialize Ring Station MAC frame. The solution to this problem is to disable RPS on the affected port. If RPS is not enabled and no server copies the Request Initialization frame, the sending station does not expect a response and it will insert into the ring.

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 exceeds 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 2210 supports multi-port 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.

In a multi-port configuration, LNM can be enabled on any source-routing token-ring bridge port. An instance of LNM is created for each port upon which LNM is enabled.

In a two token-ring configuration, the other port is always designated by a pseudo address. This is known as a multi-port bridge. It can correspond to a virtual ring or a serial line interface.

Only in the case where the IBM 2210 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`** at the `ASRT>` prompt.

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.

Chapter 11. Configuring and Monitoring LNM

This chapter describes IBM's ASRT implementation of the LNM. It includes the following sections:

- "Configuring LNM"
- "LNM Commands"
- "LAN Network Manager Dynamic Reconfiguration Support" on page 196

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.

Note: The LNM configuration commands are not effective immediately. You must restart or reload the device before they become effective.

LNM Commands

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 15 lists the LNM commands.

Table 15. LNM Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
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 you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Response to LNM Configuration Commands

The LNM configuration (Talk 6) commands do not become effective immediately. They remain pending until you issue the **reload** or **restart** command.

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. The command is entered at the `ASRT>` prompt.

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)

Displays information about the status of the LNM configuration. The command is entered at the `ASRT>` prompt.

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
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
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 must 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`**

Example:

`set password`

```
Link Number [0]? 1
Enter new password :  [ABCDEFGH]? guesswho
```

LAN Network Manager Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

LAN Network Manager (LNM) supports the CONFIG (Talk 6) **delete interface** command with no restrictions.

GWCON (Talk 5) Activate Interface

LNM does not support the GWCON (Talk 5) **activate interface** command.

GWCON (Talk 5) Reset Interface

LNM does not support the GWCON (Talk 5) **reset interface** command.

Chapter 12. Configuring and Monitoring 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:

See "TCP/IP Host Services (Bridge-Only Management)" on page 41 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 device for IP routing; instead, refer to "Chapter 14. Using IP" on page 213.

Note: To configure Host services, you cannot have any IP address configured on the interfaces. The device cannot be configured as a router for IP. The Host services are for bridging only.

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 hst
TCP/IP-Host Services user configuration
TCP/IP-Host Config>
```

Basic Configuration Procedures

The following sections describe the basic configuration procedures for enabling TCP/IP Host Services on your 2210.

Setting the IP Address

To minimally configure TCP/IP Host services, assign the 2210 an IP address by using the **set ip-host** command. This IP address is associated with the 2210 as a whole, instead of being associated with a single interface.

Enabling TCP/IP Host Services

Use the **enable services** command to enable TCP/IP Host Services.

Adding a Default Gateway

The 2210 uses its default gateway to communicate with hosts and gateways that are not on the bridged network to which the 2210 is directly connected. The 2210 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 can also statically specify one or more default gateways by using the **add default gateway** command. The 2210 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,

1. Exit from the `TCP/IP-Host config>` prompt to the `Config>` prompt.
2. Use the **write** command at the `Config>` prompt to write the current configuration to memory.
3. Enter **CTRL-P** to get to the OPCON prompt and use the **reload** or **restart** OPCON command to load a new copy of the software.
4. After restarting or reloading the 2210, return to the `TCP/IP-Host config>` prompt.

TCP/IP Host Configuration Commands

This section describes 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 device to activate the configuration commands.

Note: The TCP/IP host configuration commands are not effective immediately. They remain pending until you restart or reload the device.

Enter the TCP/IP Host configuration commands at the `TCP/IP-Host config>` prompt. Table 16 shows the commands.

Table 16. TCP/IP Host Configuration Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
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 2210’s IP address.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Response to TCP/IP Host Configuration Commands

The TCP/IP host configuration (Talk 6) commands are not effective immediately. They remain pending until you issue the **reload** or **restart** command.

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 subnet. The routing table is then built up through redirect processing. An attempt is made to detect routers that disappear. If the 2210 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 2210 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

Example: `list`

```
TCP/IP-Host config>list

TCP/IP Host SERVICES    : enabled
IP-HOST Address         : 128.185.142.1
        Mask            : 255.255.255.0
DEFAULT-GATEWAY Address : 128.185.142.47
RIP-LISTENING           : disabled
ROUTER-DISCOVERY        : enabled

TCP/IP-Host config>
```

TCP/IP Host SERVICES	Displays whether TCP/IP Host SERVICES is enabled or disabled.
IP-HOST Address	Displays the current IP-HOST Address.
IP-HOST Mask	Displays the current IP-HOST Mask.
DEFAULT-GATEWAY Address	Displays the current DEFAULT-GATEWAY Address.
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 2210's IP address. You must assign the 2210 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 2210 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.
```

Monitoring TCP/IP Host Services

This section describes how to monitor the TCP/IP Host Services on the IBM 2210.

Accessing the TCP/IP Host Monitoring Environment

To access the TCP/IP Host monitoring environment, enter the following command at the + (GWCON) prompt:

```
+ protocol hst
TCP/IP-Host>
```

TCP/IP Host Monitoring Commands

This section describes the TCP/IP Host monitoring commands. These commands allow you to view parameters and enter information requests from the active terminal. Enter these commands at the `TCP/IP-Host>` prompt. Table 17 shows the commands.

Table 17. TCP/IP Host Monitoring Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
Dump	Displays the current IP routing table. One line is printed for each destination.
Interface	Displays the IBM 2210’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 2210.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

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:

```
TCP/IP Host> dump
Type   Dest net         Mask      Cost     Age       Next hop(s)

Stat   0.0.0.0          00000000  0        51        128.185.142.47
 Dir*  128.185.142.0    FFFFFF00  1        50        BDG/0

Default gateway in use.
Type Cost     Age       Next hop
Stat 0        51        128.185.142.47

Routing table size: 768 nets (52224 bytes), 2 nets known
                      0 nets hidden, 0 nets deleted, 0 nets inactive
                      0 routes used internally, 766 routes free
```

Type — Route type which indicates how the route was derived:

- RIP - the route was learned through the RIP protocol.
- Stat - a statically configured route.
- Dir - a directly connected network or subnet.

Dest net — Displays the IP address of the destination network/subnet.

Mask — Displays the IP address mask.

Cost — Displays the Route Cost.

Age	For RIP routes displays the time, in seconds, since the route was refreshed. For other types of routes displays the time, in seconds, since the route was installed.
Next Hop	Displays the IP address of the next device on the path toward the destination host. Also displayed is the interface type used by the sending device 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 2210's IP address. When TCP/IP Host Services are running over the bridge, a single address is displayed on the terminal as Bridge/0.

Syntax:

interface

Example:

```
TCP/IP Host> interface
Interface     MTU    IP Address(es)   Mask(s)          Address-MTU
  BDG/0      1500    128.185.142.16   255.255.255.0    Unspecified
```

Interface	Displays the type of interface. For TCP/IP Host Services, this is always BDG/0, indicating the bridge.
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 device 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 terminal. 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 rate of pinging are all user-configurable. The default values are a size of 56 bytes, a TTL of 64, and a rate of 1 ping per second.

Syntax:

ping *destination source size ttl rate*

Example:

```
TCP/IP Host> ping
Destination IP address [0.0.0.0]? 128.185.142.11
Source IP address [128.185.142.16]?
Ping data size in bytes [56]?
Ping TTL [64]?
Ping rate in seconds [1]?
PING 128.185.142.16 -> 128.185.142.11: 56 data bytes, ttl=64, ... every 1 sec.
56 data bytes from 128.185.142.11: icmp_seq=0. ttl=254. time=0. ms
56 data bytes from 128.185.142.11: icmp_seq=1. ttl=254. time=0. ms
56 data bytes from 128.185.142.11: icmp_seq=2. ttl=254. time=0. ms
56 data bytes from 128.185.142.11: icmp_seq=3. ttl=254. time=0. ms
56 data bytes from 128.185.142.11: icmp_seq=4. ttl=254. time=0. ms
56 data bytes from 128.185.142.11: icmp_seq=5. ttl=254. time=0. ms

----128.185.142.11 PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
```

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 left-most number indicating the distance from the device executing the command (in network device hops).

The traceroute is complete when the destination is reached, an ICMP Destination Unreachable message is received, or the path length reaches 32 network device hops.

Syntax:

traceroute *destination source size probes wait ttl*

Example:

```
TCP/IP Host> traceroute
Destination IP address [0.0.0.0]? 128.185.144.239
Source IP address [128.185.142.16]?
Data size in bytes [56]?
Number of probes per hop [3]?
Wait time between retries in seconds [3]?
Maximum TTL [32]?
TRACEROUTE 128.185.142.16 -> 128.185.144.239: 56 data bytes
 1 128.185.142.11 16 ms 0 ms 0 ms
 2 128.185.143.33 16 ms 0 ms 0 ms
 3 128.185.144.239 16 ms 0 ms 0 ms
```

In the display:

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 device 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 IBM 2210. Routers can be learned through:

- Static configuration (using the **add default-gateway** command explained on page "Add" on page 198).
- 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`

TCP/IP Host Services Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

TCP/IP Host Services (HST) does not support the CONFIG (Talk 6) **delete interface** command.

GWCON (Talk 5) Activate Interface

The GWCON (Talk 5) **activate interface** command is not applicable for TCP/IP Host Services (HST). TCP/IP Host Services has no interface-specific configuration parameters.

GWCON (Talk 5) Reset Interface

The GWCON (Talk 5) **reset interface** command is not applicable for TCP/IP Host Services (HST). TCP/IP Host Services has no interface-specific configuration parameters.

Non-Dynamically Reconfigurable Commands

The following table describes the TCP/IP Host Services (HST) configuration commands that cannot be dynamically changed. To activate these commands, you need to reload or restart the device.

TCP/IP Host Monitoring Commands (Talk 5)

Commands
CONFIG, protocol hst, add default-gateway
CONFIG, protocol hst, delete default-gateway
CONFIG, protocol hst, disable rip-listening
CONFIG, protocol hst, disable router-discovery
CONFIG, protocol hst, disable services
CONFIG, protocol hst, enable rip-listening
CONFIG, protocol hst, enable router-discovery
CONFIG, protocol hst, enable services
CONFIG, protocol hst, set ip-host address

Part 2. Configuring and Monitoring Router Protocols

Chapter 13. Overview of Routing over ATM

Note: See the glossary for definitions of the abbreviations and terms used in this chapter. This chapter describes the routing over native ATM.

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 Figure 26 and Figure 27.
IP routing is supported between arbitrary combinations of Classical IP (CIP) and

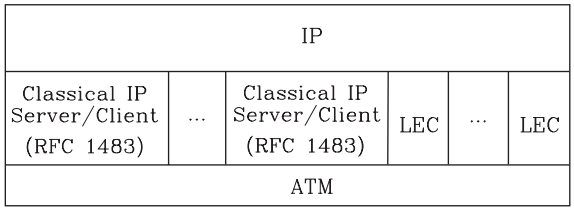


Figure 26. IP Routing

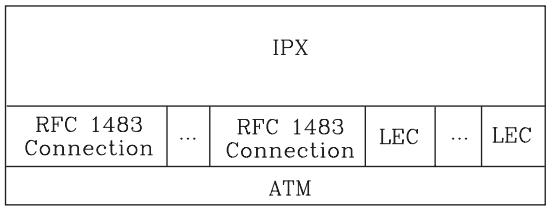


Figure 27. IPX Routing

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.

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 multiprotocol 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.

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

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 device 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 device information to and from all RFC 1483 connections with other IPX routers.

Like other routers that support RFC 1483 LLC/SNAP encapsulation on ATM, the router 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.

As with Classical IP, Quality of Service characteristics can be specified by configuring VCC traffic parameters such as Peak and Sustained Rates, and multiple circuits can be configured on a single ATM interface.

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.

You can specify destination IPX host numbers during VCC configuration or they can be 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 correctly 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 information that is 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 yet been standardized, the combination is specified for IPX over Frame Relay in RFC 1490/2427. [2]

2. T. Bradley, C. Brown, and A. Malis, “Multiprotocol Interconnect Over Frame Relay,” RFC 1490/2427, Wellfleet Communications Inc. and Ascom Timeplex Inc., July 1993.

Chapter 14. Using IP

This chapter describes how to configure the Internet Protocol (IP). It includes the following sections:

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 about 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 243.)
2. Assign IP addresses to network interfaces. (See “Assigning IP Addresses to Network Interfaces”.)
3. Enable dynamic routing. (See “Enabling Dynamic Routing” on page 216.)
4. Add static routing information, if necessary. (See “Adding Static Routing Information” on page 218.)
5. Enable ARP subnet routing, if necessary. (See “Enabling ARP Subnet Routing” on page 221.)
6. Set up ARP parameters, if necessary. (See “Setting Up ARP Configuration” on page 220.)
7. Exit the IP configuration process.
8. Restart the router to activate the configuration changes.

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 with 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
```

Multiple IP addresses can be assigned to a single network interface.

By default the IP addresses assigned to the network interfaces must each be in a different network or subnet. The **enable same-subnet** command removes the restriction.

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
```

Using Dynamic Address

Dynamic Address can be used to identify an interface that will learn its IP address from the remote end of a Point-to-Point Protocol (PPP) link using Internet Protocol Control Protocol (IPCP). The interface must first be added as an unnumbered serial line (0.0.0.n). At the time of IPCP completion IP will be notified and the negotiated IP address will be installed on the specified interface. To enable Dynamic Address, use the following steps:

- PPP must be configured to request an IP address as follows:

```
PPP 3 Config>set ipcp
IP COMPRESSION [no]:
Request an IP address [no]: yes
Interface remote IP address to offer if requested (0.0.0.0 for none) [0.0.0.0]?
```

- IP must be configured on the PPP interface as an unnumbered serial line:

```
IP config>add address
Which net is this address for [0]? 3
New address []? 0.0.0.3
Address mask [0.0.0.0]? 255.255.255.255
```

- IP must enable Dynamic Address on the same interface:

```
IP config>enable dynamic-address
Interface address []? 0.0.0.3

IP config>list address
IP addresses for each interface:
intf  0   192.168.8.1      255.255.255.0    Local wire broadcast, fill 1
intf  1                                     IP disabled on this interface
intf  2                                     IP disabled on this interface
intf  3   0.0.0.3          255.255.255.0    Local wire broadcast, fill 1
                                            DYNAMIC-ADDRESS Enabled
```

Assigning IP Addresses to the Bridge Network Interface

The 2210 routes IP packets on the network interfaces to which IP addresses are assigned (*routing interfaces*) and bridges IP packets on the network interfaces on which bridging is configured, but on which no IP address is assigned (*bridging interfaces*). The 2210 can receive IP datagrams from the bridging interfaces, send IP datagrams to the bridging interfaces, and route IP packets between the bridging interfaces and the routing interfaces. You can enable these functions on the 2210 by adding one or more IP addresses to the Bridge Network Interface. The Bridge Network Interface is a logical interface that connects IP to the bridged network to which the 2210 is connected.

To add IP addresses to the Bridge Network Interface, use the **add address** command, specifying **bridge** as the network interface:

```
IP config> add address bridge ip-address ip-address-mask
```

This command does not assign an IP address to any individual bridging interface but, in effect, to all of the bridging interfaces.

Assigning IP addresses to the Bridge Network Interface can free up one of the physical network interfaces (physical ports) on the 2210. To understand this, first consider Figure 28, which illustrates an IP internetwork with separate devices performing the router and bridge functions. LAN 2 and LAN 3 are connected by the bridge to form a bridged network; to the router, this bridged network is a single IP subnet defined by the IP address 9.67.5.1 and the mask 255.255.255.0.

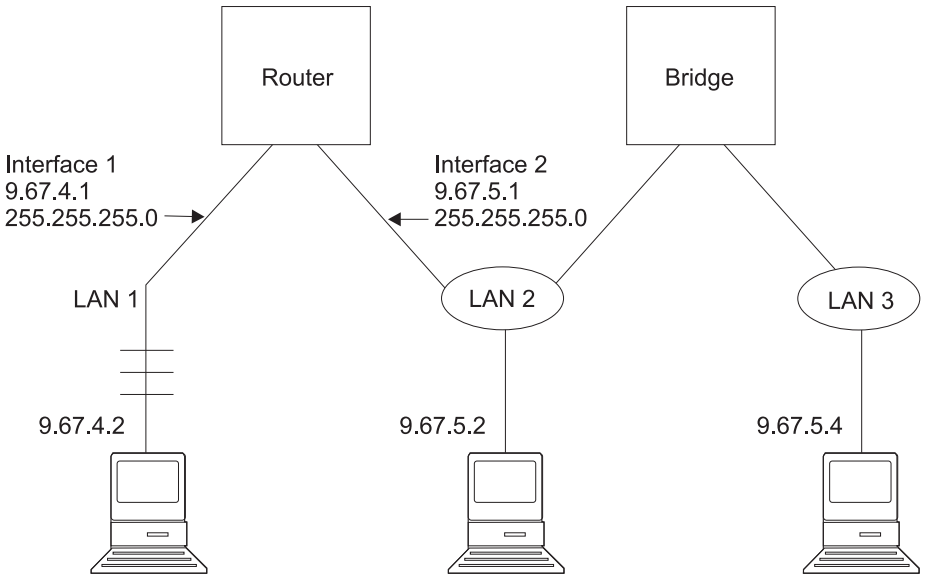


Figure 28. Routing to a Bridged Network-Alternative 1

Figure 29 illustrates the same internetwork with the router and bridge functions merged into a single device. In this figure, the router still has its own physical network interface (Interface 2) to the bridged network.

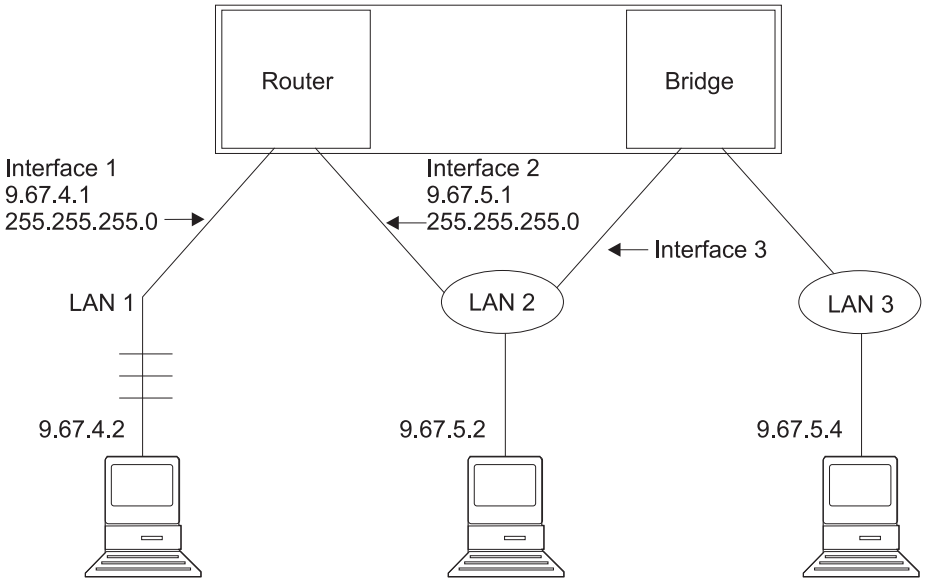


Figure 29. Routing to a Bridged Network-Alternative 2

Finally, in Figure 30, the physical network interface of the router to the bridged network is replaced by the Bridge Network Interface, which is an internal interface. This is the same internetwork illustrated in Figures 28 and 29, but the router no longer requires its own physical network interface to the bridged network.

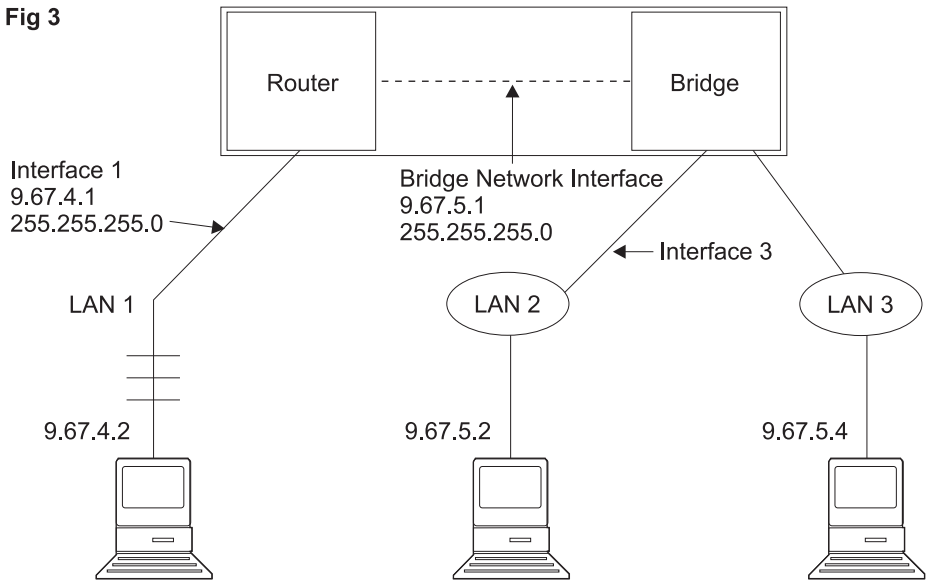


Figure 30. Routing to a Bridged Network-Alternative 3

Note: If IP addresses are configured on the bridge network interface, you cannot configure IP addresses on any token-ring interface on which source route bridging is configured.

Setting the Internal IP Address

This is an IP address that is independent of the state of any interface and is set without reference to any interface. Some IP configurations require it. See the command **set internal-IP-address** on page 293 for more information.

Enabling Dynamic Routing

Use the following procedures to enable dynamic routing on the router. The router software supports OSPF, RIPv1, and RIPv2 for interior gateway protocols (IGPs) as well as BGP, which is an external gateway protocol.

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
Maximum LSA size [2048]?
```

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 16. Using OSPF” on page 325.

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.

RIP is not supported on X.25 or on native ATM (RFC 1577) network interfaces. For these types of interfaces, use OSPF instead of RIP for an interior gateway protocol (IGP). RIP is supported on ATM LAN Emulation network interfaces.

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
IP config> set originate-rip-default
```

Note: These commands are not displayed when IP routing policies are configured. See “Route Filtering With Policies” on page 228 for more information.

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 1*.

Adding Static Routing Information

This procedure is necessary only for routing information you cannot obtain from any of the above dynamic routing protocols. Static routing information persists over power failures and is used for routes that never change or cannot be learned dynamically.

The destination of a static route is described by an IP address (*dest-addr*) and an IP address mask (*dest-mask*). The mask indicates the range of IP addresses to which the route applies; for example, a route with IP address 10.0.0.0 and mask 255.0.0.0 applies to IP addresses from 10.0.0.0 through 10.255.255.255. The route to the destination is described by the IP address of the next hop router (*next-hop*) and the cost of forwarding a packet on this route (*cost*). You may define up to four static routes per IP destination. Static routes to a given destination may have the same cost, in which case IP may use the routes simultaneously, or they may have different costs, in which case IP uses the lowest cost route that works.

To create, modify, or delete a static route, use the following commands:

```
IP config>add route dest-addr dest-mask next-hop cost
IP config>change route dest-addr dest-mask next-hop cost
IP config>delete route dest-addr dest-mask
```

These commands take effect immediately without the need to reboot the router.

Longest Match Rule

Because the destination of a route includes the IP address mask, it is possible for more than one route to match a particular IP address; for example, for the IP address 10.1.2.3, a route with IP address 10.0.0.0 and mask 255.0.0.0 and a route with IP address 10.1.0.0 and mask 255.255.0.0 both match. To determine which route to use, the longest match rule is applied. The route with the largest mask is used (in this case the route with IP address 10.1.0.0 and mask 255.255.0.0).

Default, Network, Subnet and Host Routes

Routes can be classified as *default*, *network*, *subnet*, or *host*, according to their destination IP address and mask.

A *default* route has an IP address/mask of 0.0.0.0/0.0.0.0. This route matches all destination IP addresses, but because of the longest match rule, it is used only if there is no other matching route. The following command creates a static default route:

```
IP config> add route
IP destination [ ]? 0.0.0.0
Address mask [255.0.0.0]? 0.0.0.0
Via gateway 1 at [ ]? 192.9.1.4
Cost [1]? 5
Via gateway 2 at [ ]?
IP config>
```

The static default route may also be set by the **set default network-gateway** command; however, this command does not take effect immediately, and it allows you to define only one default static route. The following example creates the same static default route as the above **add route** command:

```
IP config> set default network-gateway
Default gateway [ ]? 192.9.1.4
gateway's cost [1]? 5
IP config>
```

A *network route* has a mask that depends on the value of the route's destination IP address as specified by the IP address classes defined in RFC 791:

IP Address Class	IP Address Range	Network Mask
A	0.0.0.0 - 127.255.255.255	255.0.0.0
B	128.0.0.0 - 191.255.255.255	255.255.0.0
C	192.0.0.0 - 223.255.255.255	255.255.255.0

The **add route**, **change route**, and **delete route** commands use the network mask that corresponds to the destination IP address as the default mask value. The following command creates a static network route:

```
IP config> add route 172.16.0.0
Address mask [255.255.0.0]?
Via gateway 1 at [ ]? 192.9.1.4
Cost [1]? 5
Via gateway 2 at [ ]?
IP config>
```

A static network route may also be set by the **set default subnet-gateway** command; however, this command does not take effect immediately, and it allows you to define only one static route per destination. The following example creates the same static network route as the above **add route** command:

```
IP config> set default subnet-gateway
For which subnetted network [ ]? 172.16.0.0
Default gateway [ ]? 192.9.1.4
gateway's cost [1]? 5
IP config>
```

A *subnet route* has a mask that is larger than the network mask for the route's destination IP address. The following command creates a static subnet route:

```
IP config> add route 172.16.1.0
Address mask [255.255.0.0]? 255.255.255.0
Via gateway 1 at [ ]? 192.9.1.4
Cost [1]? 5
Via gateway 2 at [ ]?
IP config>
```

A *host route* is a route to a specific IP address; it has a mask of 255.255.255.255. The following command creates a static host route:

```
IP config> add route 172.16.1.2
Address mask [255.255.0.0]? 255.255.255.255
Via gateway 1 at [ ]? 192.9.1.4
Cost [1]? 5
Via gateway 2 at [ ]?
IP config>
```

Interaction Between Static Routing and Dynamic Routing

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
Use Route Policy [No]?
Import BGP routes [No]?
Import RIP routes [No]?
Import static routes [No]? yes
Import direct routes [No]?
Import subnet routes [Yes]?
```

Nexthop Awareness

Nexthop Awareness allows the router to sense whether a neighboring router is up or down. When this option is enabled, the router makes a more accurate determination of whether a static route that uses the neighboring router as its next hop will function. It also allows the router to determine over which network interface a static route's next hop can be reached when that next hop is in an IP subnet that is defined on multiple network interfaces.

To enable Nexthop Awareness on a particular IP interface, enter the following command at the IP configuration prompt:

```
IP config> enable nexthop-awareness ip-interface-address
```

To disable Nexthop Awareness on a particular IP interface, enter the following command at the IP configuration prompt:

```
IP config> disable nexthop-awareness ip-interface-address
```

Nexthop Awareness is supported only on frame relay networks on which the neighboring routers support inverse ARP.

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 27. Using ARP” on page 571.

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 591).

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 correct 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 use 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 allows you to specify certain criteria that the router uses to control packet forwarding. The following main types of filtering are provided to help you achieve your security and administrative goals:

- Access control
- The policy feature (see Using the Policy Feature in *Using and Configuring Features*)
- Route filtering

Note: For IPv4, you now have the option to configure access control rules in a policy database to designate access control and determine how IP packets are filtered. See the chapter “Using Policy” in *Using and Configuring Features* for details.

Access Control

Access control allows the IP router to control the processing of individual packets based upon the following parameters:

- IP source address
- IP destination address
- IP protocol number
- TCP or UDP source port number
- TCP or UDP destination port number
- TCP SYN and ACK bits
- ICMP type and code
- Precedence and Type of Service (TOS) filtering

Access control can limit the ability of particular sets of IP hosts and services to communicate with one another.

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 apply only 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 31 illustrates the series of access control lists through which a packet must pass before being forwarded.

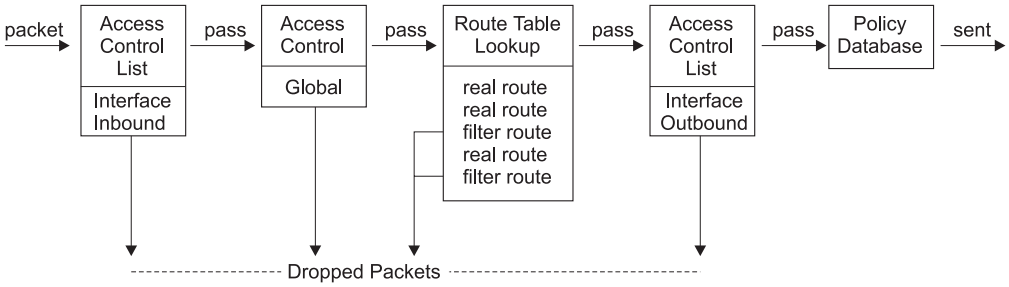


Figure 31. Access Control Lists in the Packet Forwarding Path

Access Control Rules

Each access control list consists of one or more access control rules that set the filtering criteria. Some access control rules define the global filters that affect all the interfaces on the router and others define the interface-specific access control lists (also called packet filters). The global access control rules are configured using the **add access** command at the `IP config>` prompt. The packet filters are set using two commands at the `IP config>` prompt: the **add packet-filter** command to define the filter and the **update packet-filter** command to configure it.

As IP packets flow through the router, IP packet fields are compared to the access control rules. A packet matches a rule if every specified field in the rule matches a corresponding field in the packet. If a packet matches a rule, and the rule filter type is inclusive, the packet *passes*. If the rule filter type is exclusive, the packet is *dropped* and is not processed any further by the router. If no rules 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 information:

- 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 rule, an inclusive rule must exist for any packets that do not match any of the access control rules to pass the list. One method of allowing all packets that do not match any of the specified rules to pass is to include the following wildcard rule as the last rule in the list:

```
IP config> add access-control
Enter type [E]? i
```

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. You can use the **enable packet-filter** and the **disable packet-filter** commands to enable and disable specific packet filters when IP access control is enabled.

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 rule as the last in the access control list. Alternatively, you can add specific rules 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 rules are added at the `IP config>` prompt:

```
IP config>  add access-control...
```

Global access control rules can be listed, moved, or deleted using the **list**, **move**, or **delete** commands. See these commands for further information.

Defining Packet Filters

To define packet filters, which are interface-specific, 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.

```
IP config> add packet filter
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 Rules for Packet Filters

You must define access control rules for each defined list (packet filter). Otherwise, defined packet filters will have no effect on incoming or outgoing traffic. Use the **update packet-filter** command at the `IP config>` prompt to define access control rules. The router first prompts you for the name of the packet filter that you want to update. The `IP config>` prompt then changes to `Packet-filter 'name' Config>` where ‘name’ is the list name that you provide.

```
IP config>  update packet-filter
Packet-filter name [ ]?  test
Packet-filter 'test' Config>
```

From this prompt, you can issue **add**, **disableenablelist**, **move**, and **delete** commands. These commands are similar to those used to modify the global access control list.

Parameters for Access Control Rules

Access control rules consist of multiple parameters. Some parameters can be specified in all access control rules, while others can be specified only in the rules for packet filters. The following parameters can be specified in all access control rules:

- Type (inclusive, exclusive)
- IP source address and mask

- IP destination address and mask
- IP protocol number range
- TCP/UDP destination port number range
- TCP/UDP source port number range
- TCP SYN filtering
- ICMP message type and code
- Precedence and TOS filtering support
- Policy-based routing (selecting the next-hop gateway)
- SysLog facility option
- Security logging options
- Compression-bypass
- Encryption-bypass

The following parameters are for packet filters only:
- Packet filter name
- Source address verification

Additional types:
- Network address translation (NAT)

Type: The type designation of an access control rule determines how it affects packets that match it, as follows:

- An *exclusive* (E) rule discards packets.
- An *inclusive* (I) rule allows packets to be processed further by the router.
- A *network address translation*, or NAT (N) rule, passes packets to NAT for address translation.

NAT rules are valid only in packet filters and only when specified in combination with inclusive (IN). Use the Configuration Program to first specify inclusive, and then to specify NAT.

IP Source and Destination Addresses and Masks: Each rule has an IP address and mask pair for both the IP source and destination addresses. When an IP packet is compared to an access control rule, the IP address in the packet is ANDed with the mask in the rule, and the result compared with the address in the rule. For example, a source address of 26.0.0.0 with a mask of 255.0.0.0 in an access control rule will match any IP 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 match only IP destination host address 192.67.67.20. An address of 0.0.0.0 with mask 0.0.0.0 is a wildcard that matches any IP address.

IP Protocol Number Range: 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 range specified by the access control rule will match (including the first and last numbers of the range). 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).

TCP/UDP Source and Destination Port Number Range: If the IP protocol number range includes 6 (TCP) or 17 (UDP), TCP/UDP port number ranges can also be specified in an access control rule, for both source and destination ports. These ranges are compared to the port number field in the TCP or UDP header of the IP packet; a port number value within the specified range (including the first and last numbers) will match. These fields are 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.

TCP Connection Establishment (SYN) Filtering: If the protocol number range includes 6 (for TCP) and the filter type is exclusive, you can set TCP connection establishment filtering. When TCP connection establishment filtering is enabled, the access control rule is applied to a TCP packet only if that packet establishes a TCP connection. (These are the packets in which the TCP SYN bit is 1 and the ACK bit is 0.)

ICMP Message Type and Code: If the protocol number range includes 1 (for ICMP), you can specify the ICMP message type and code. The default is to apply the access control rule to all ICMP message types and codes.

Precedence and TOS Filtering Support: The router that supports TOS has identified certain routes that provide the requested levels of service. The router sends packets over the routes according to the setting of their TOS bits.

TOS in IP is not a guarantee of any particular type of service, but a request to the router to provide service of the type requested. For example, a packet with a TOS field requiring maximum throughput can be sent over several hops that have different bandwidths. It will get normal service - no special treatment - if it should pass over a hop managed by a router that does not support TOS. See the **add access-controls** command on page 244 for descriptions of these parameters.

You can also set filters to provide QoS based on TOS bits using the Bandwidth Reservation System (BRS) feature. BRS is used with PPP and frame relay interfaces. Refer to “Using Bandwidth Reservation and Priority Queuing” and “Configuring and Monitoring Bandwidth Reservation” in *Using and Configuring Features*.

Parameters for TOS-Based Routing Support: To enable the router to interpret TOS bits and route packets according to those bits, you create an access control rule from which the router will receive TOS packets for filtering and Type of Service routing. This access control rule applies to all the interfaces on the router. The following parameters are used to define the TOS bits that the router will compare:

- Range-start value for the TOS byte bits
- Range-end value for the TOS byte bits
- Filter mask to determine which bits in the TOS byte are included in the range

Modification of the TOS Bits: To enable the router to modify the TOS bits of incoming packets, you create a global access control rule from which the router will receive TOS packets that are to be modified. Modifying the value of the TOS bits is a separate activity from interpreting them and routing the packet. If both interpretation and modification are configured, the modification will be done after the interpretation. The following parameters are used to define the TOS bits to be modified:

- A new value for the TOS bits
- A modification mask to determine which bits in the TOS byte are to be changed

Policy-Based Routing (Selecting the Next-Hop Gateway): You can filter inbound packets to direct them to a manually selected next hop gateway address

(known as policy-based routing). To do this, create an inclusive inbound access control rule either globally, for the router, or for a particular interface, and provide the following parameters:

- Whether to use policy-based routing
- The IP address of the next hop gateway
- Whether or not to send the packet using the normal routing table if the next hop is unavailable

SysLog Facility Option: SysLog is a logging option that generates a SysLog message to a remote logging server. If SysLog is enabled, the SysLog facility option specifies the SysLog facility that is used for remote logging. This option, which has a default of *User*, defines the remote logging file where the SysLog messages can be stored and later analyzed. The SysLog facility option is displayed both in the Configuration Program and in the command line interface.

Security Logging Options: If you enable security logging, you can specify any or all of these logging options:

- ELS messages
- SNMP traps
- SysLog

If specified, ELS messages and SysLog can use *short* or *long* message format. SNMP traps can be *enabled* or *disabled*. If no logging option is specified, security logging is disabled.

The SysLog priority level can also be configured. It specifies the level of the error message that will be displayed, such as *Emergency* or *Information*. The default is the router system default value. The SysLog priority levels are displayed both in the Configuration Program and in the command line interface.

The SysLog messages are sent to a remote server and saved to the SysLog files of the current SysLog facility option.

Packet Filter Name: This interface-specific parameter can consist of any name. It can be up to 16 characters long and can include dashes (-) and underscores (_). Up to two access control record lists can be configured for each packet filter name, one for outgoing packets and one for incoming packets.

Source Address Verification: This input packet filter option verifies that a received packet's source IP address is consistent, based on the IP routing table, with the interface from which it was received. This option helps prevent the forwarding of packets from a misbehaving IP host that is using a source IP address that does not belong to it, a behavior known as *spoofing*.

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 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
```

This example shows how to create a global access control rule. Values are entered to enable the interpretation of TOS bits of packets arriving from IP address 9.1.2.3 and to change the values of these bits before sending the packets. See “Add” on page 244 for an explanation of the meaning of the parameters that create TOS filtering and policy-based routing.

```
IP config> add access-control
Enter type [E]? i
Internet source [0.0.0.0]? 9.1.2.3
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]?
Destination mask [0.0.0.0]?
Enter starting protocol number ([0] for all protocols) [0]?
Enter starting DESTINATION port number ([0] for all ports) [0]?
Enter starting SOURCE port number ([0] for all ports) [0]?
Filter on ICMP Type ([-1] for all types) [-1]?
TOS/Precedence filter mask (00-FF - [0] for none) [0]? e0
TOS/Precedence start value (00-FF) [0]?
TOS/Precedence end value [0]?
TOS/Precedence modification mask (00-FF - [0] for none) [0]? 1f
New TOS/Precedence value (00-FF) [0]? 08
Use policy-based routing? [No]: y
Next hop gateway address [ ]? 9.2.160.1
Use default route if next hop gateway unreachable? [Yes]:
Enable Logging (Yes or [No]):
```

Route Filtering Without Policies

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 packet fields other than the destination IP address can be done using access control, described above, or by using route filtering policies as described in “Route Filtering With Policies” on page 228.

The following methods are used in this router to influence the content of the routing table.

- Filter routes
- RIP input filters
- Route table filtering

Defining Filter Routes

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. Filter routes are ***not*** recommended when OSPF is used in your network; OSPF-learned internal routes will override filtered routes in the routing table.

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
```

If more granular filtering of RIP routes is required, the route policies that are described in the following command can be utilized:

```
IP config> add accept-rip-route ip-network/subnet/host
```

Defining Route Table Filtering

When route table filtering is enabled and route filters are defined, checking is performed before adding routes to the IP routing table. If the route to be added matches on an inclusive route filter, it will be added to the IP route table. If it matches on an exclusive route filter, it will not be added to the IP route table. Direct and static routes will never be filtered.

This function can be used to prevent routes from being added to the IP route table in situations where the network administrator does not want all routes advertised by routing protocols to be available. This function could be used in a service provider environment to prevent customers from having access to each other's networks.

Route Filtering With Policies

Route filter policies are definitions that describe a route or set of routes. A route filter policy consists of the name of the route filter policy and at least one entry that defines an address or range of addresses for the routes to be filtered. Each entry includes instructions to include or exclude the routes defined in that entry from the routing table. Route filter policies can be used to filter the routes that RIP and OSPF install in the IP forwarding table and advertise from the IP forwarding table.

A route filter policy is identified by a 15-character ASCII string, for example, *ospf-import*. After the route filter policy is named, you need to configure at least one entry that is associated with that route filter policy. Use the **add route-policy** command at the `IP config>` prompt to add the policy, the **change route-policy** command to bring up the `IP Route Policy Config>` prompt, and the **add entry** command at the `IP Route Policy Config>` prompt to define each entry for the policy.

You are required to assign an index number to each entry when you configure it. This number is used to identify the entry for matching.

Matching is done either by *linear match* or by *longest match lookup*. You select one of these methods when you use the **add route policy** command to create the route filter policy. If you choose linear match, the route being filtered will be compared to the entries in the list one after the other, based upon the index number. As soon as a match is found, the route is filtered. If you choose longest match lookup, the route being filtered is compared to the filter entries according to the longest match lookup. If more than one entry specifies the same IP address and mask, then the route being filtered is compared in ascending order by index number.

For example, suppose that you want to exclude the addresses for the network 9.8.0.0 with the mask 255.255.0.0 but you want to include the host address 9.8.1.8 with the mask 255.255.255.255. According to the longest match lookup filtering

method, you can include 9.8.1.8 with the mask 255.255.255.255 and exclude address 9.8.0.0 with the mask 255.255.0.0. Then, of all the addresses in that subnet, only 9.8.1.8 will be included.

To get the same result using linear match, you would have to assign a lower index number to the inclusive filter than to the exclusive one. For example, the address 9.8.1.8 with the mask 255.255.255.255 requires a lower index number than the address 9.8.0.0 with the mask 255.255.0.0. Otherwise, the rule that excludes 9.8.0.0 will also exclude the address 9.8.1.8.

The *match type* is a parameter that determines how the address mask for the entry will be processed. If this parameter is *exact*, the software will match the route only on the exact address and mask specified by the entry and will not regard the address as a range. If the match type is *range*, the router will read the address and mask as a range and match the route if it falls within the range.

In addition to entries, you can configure actions and match conditions associated with each entry. Actions are changes made to the route before it is advertised, such as setting a metric on a route. Match conditions change the rules according to which the route is selected. After a match is found based on the destination address, the match condition puts further constraints on the match. For example, if the match condition is protocol BGP, routes are not matched unless the entry address matches and the packet belongs to the BGP protocol. These are the match conditions:

- Protocol, such as RIP, OSPF, or BGP
- AS (Autonomous System) number
- Gateway address for the route's next hop
- Network number for the route's next hop
- Metric range
- Source gateway, which is applicable only to RIP receive policy

Actions and match conditions, which fine tune the filtering of the entry, are optional.

AS Boundary Routing Filter Policies for OSPF

If a route filter policy is used to control OSPF routing tables, it is configured during the configuration of OSPF. See the **enable** command on page 351 for more information.

Send and Receive Filter Policies for RIP

You can use route filter policies to define which routes RIP will send or receive. These route filter policies can be configured globally, for all the IP interfaces on the router, or per IP interface. If a send route filter policy is enabled, all routes that conform to the policy are advertised and the values for *default-routes*, *host-routes*, *net-routes*, *subnet-routes*, and *static-routes* are ignored. The values for *poison-reverse-routes*, *ripv1-only routes*, and *outage-only* routes are not affected by the send route filter policy. If sending *all-routes* is disabled, no routes will be advertised, even if a global send policy is specified.

If a receive route filter policy is configured and receiving RIP is enabled, the configured policy will take the place of any enabled or disabled dynamic route types. In other words, all routes included by the route filter policy and conforming to the constraints of the RIP protocol will be accepted.

Route Aggregation

To reduce the proliferation of routing information in the routing tables of boundary routers, the 2210 offers route aggregation. Route aggregation enables you to use one IP address and subnet mask to define a range of subnets that belong to different routing domains.

A certain level of route aggregation occurs in all routers in that the router will generate classful routes corresponding to each set of subnet routes. This automatic aggregation can be disabled by enabling classless routing support through IP configuration. For example, a router will generate the route 10.0.0.0/255.0.0.0 as long as there are subnet routes within that 10 network. This route can then be advertised by protocols that do not support variable length subnetting (for example, RIPv1).

Configuring route aggregation in the router can replace the two address ranges with one new aggregate address that identifies both the RIPv2 domain and the OSPF domain. The aggregate route defined in the example is 10.0.0.0 with the subnet mask 255.254.0.0. This single aggregate route is advertised to both the RIPv2 and the OSPF domains. This is a simple example. Depending upon the circumstances, an aggregate route can be used to represent multiple domains, not just two.

An aggregate route can be configured to be generated unconditionally or conditionally. If it is generated unconditionally, it will be generated whether or not there are any component routes. A metric value must be configured for an unconditional aggregate route. .

If an aggregate route is generated conditionally, the metric is derived from a component route or the aggregate route's policy. You are required to associate a conditionally generated aggregate route with a route filter policy. You do this by providing a route policy identifier for the aggregate route. In this case, the aggregate route is advertised only when a component route becomes available. A component route meets two requirements:

- It has a destination address that falls within the range defined by the aggregate route.
- It meets all the conditions defined by the route filter policy associated with it.

For example, the route with a destination address of 10.3.34.216 constitutes a component route of the aggregate route 10.0.0.0/255.252.0.0 if the route filter policy that has been associated with this aggregate route has no conditions other than the address range. See "Route Filtering With Policies" on page 228 for more information about route filter policies.

Using Route Aggregation in RIP and OSPF

RIP and OSPF are two of the protocols that import and export routing information. Route aggregation is used in the context of protocols such as these that advertise routing information through the IP network. The following paragraphs explain how to implement route aggregation for RIP and OSPF.

For the IP configuration command **enable**, a new option **enable sending aggregate-routes** is supported. This option enables RIP to determine whether to advertise aggregate routes. For RIPv1, the aggregate routes will not be advertised unless they adhere to RIPv1 policy constraints.

For OSPF AS boundary policy, a new aggregate route question is supported for the **enable as boundary routing** command. The question is whether to import

aggregate routes into OSPF as OSPF external routes. This question is not displayed if you configure a route filter policy.

Route filter policies can be used to filter the aggregate routes that RIP will send or that OSPF will import. To identify a route filter policy entry to be used to filter aggregate routes, a new match condition has been added to the route filter policy configuration. This is the protocol *aggregate*. When a route filter policy entry has this match condition, the route filter policy is applied only to aggregate routes.

The following example shows how to use a route filter policy to filter aggregate routes sent by RIP.

Example:

First, add a route policy which allows aggregates. The second entry in this example of the configuration of a route policy filter is optional. It is not necessary to exclude the routes that do not match entry 1 because the default behavior of a route filter policy is to exclude routes which do not match on any route policy entries.

```
IP config>add route-policy
Route Policy Identifier [1-15 characters] []? all-aggregates
Use strictly linear policy? [No]:
IP config>change route-policy
Route Policy Identifier [1-15 characters] []? all-aggregates
all-aggregates IP Route Policy Configuration
IP Route Policy Config>add entry 1
IP Address [0.0.0.0]?
IP Mask [0.0.0.0]?
Address Match (Range/Exact) [Range]?
Policy type (Inclusive/Exclusive) [Inclusive]?
IP Route Policy Config>
IP Route Policy Config>add entry
Route Policy Index [1-65535] [0]? 2
IP Address [0.0.0.0]?
IP Mask [0.0.0.0]?
Address Match (Range/Exact) [Range]?
Policy type (Inclusive/Exclusive) [Inclusive]? exclusive
```

Add a match-condition for protocol aggregate. This parameter enables the route filter policy to process only aggregate routes.

```
IP Route Policy Config>add match-condition protocol aggregate

Route Policy Index [1-65535] [0]? 1
Route policy entry match condition updated or added
IP Route Policy Config>exit
```

List the route filter policy:

```
IP config>list route-policy all-aggregates

                Route Policy: all-aggregates (0x3904)


IP Address      IP Mask           Match  Index  Type
-----------------------------------------------------
0.0.0.0         0.0.0.0           Range  1      Include
    Match Conditions:  Protocol: Aggregate
0.0.0.0         0.0.0.0           Range  2      Exclude
```

Advertise aggregates in RIP using the policy *all-aggregates*.

```
IP config>enable sending policy global
Route Policy Identifier [1-15 characters] [ ]? all-aggregates
```

Make the policy active through the console (Talk 5) by dynamically resetting IP. Display the run-time state of RIP using the console.

```
IP>reset ip
IP>RIP
                                RIP Interfaces

Interface-Addr  Interface-Mask  Version  In Out  Send-Flags       Receive-Flags
10.69.1.1       255.255.255.0      1      1   0  OFF              N,S
153.2.2.25      255.255.255.240    1      1   0  Global,D,P       N,S,H
10.2.1.1        255.255.255.0      1      1   0  Global,P         N,S
Send Flags: A=Aggr N=Network S=Subnet H=Host St=Static D=Default O=Outage-Only
            P=PoisonReverse Policy=Send-Policy Global=Global-Send-Policy
Recv Flags: N=Network S=Subnet H=Host OSt=Override-Static OD=Override-Default
            Policy=Receive-Policy Global=Global-Receive-Policy

        RIP Policy

Interface-Address  Send Policy        Receive-Policy
10.69.1.1          all-aggregates     Not-Applicable
153.2.2.25         all-aggregates     Not-Applicable
10.2.1.1           all-aggregates     Not-Applicable
RIP global receive policy: NONE
RIP global send policy: all-aggregates
```

RIP always originates default with cost 5.

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 2131, 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 DHCP/BOOTP server that is providing these services. The router can provide DHCP/BOOTP Server function. Refer to “Using DHCP Server” in *Using and Configuring Features*.
- *BOOTP relay agent* or *BOOTP forwarder* - a device that forwards requests/replies exchanged by the Client and Server. This router can provide 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 added 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
```

Note: The DHCP Server feature described in "Using DHCP Server" in *Using and Configuring Features* and this BOOTP forwarding process should not both be enabled on the same router. If both are enabled, the DHCP Server feature will take precedence and BOOTP forwarding will not occur.

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.

Adding a BOOTP/DHCP Server

To add a BOOTP or DHCP server to the router's relay agent configuration, enter the following command at the IP configuration prompt:

```
IP config> add bootp-server server-IP-address
```

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.

IP and SNA Integration

You can use TN3270E to integrate IP and SNA. Refer to the chapter entitled “Using APPN” in the *Protocol Configuration and Monitoring Reference Volume 2* and the chapter entitled “Configuring and Monitoring APPN” in the *Protocol Configuration and Monitoring Reference Volume 2* for more information about TN3270E.

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 2210-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 dest-ip-address
```

Configuring Virtual Router Redundancy Protocol (VRRP)

The use of a statically configured default route is popular for host IP configurations. It minimizes configuration and processing overhead and is supported by virtually every IP implementation. This mode of operation is likely where dynamic host configuration protocols are deployed that typically provide configuration for an end-host IP address and default gateway. However, this creates a single point of

failure. Loss of the default router results in a catastrophic event, isolating all end-hosts that are unable to detect any alternate path that may be available.

The Virtual Router Redundancy Protocol (VRRP) is designed to eliminate the single point of failure inherent in the static default routed environment. VRRP specifies an election protocol that dynamically allows a set of routers to back up each other. The VRRP router controlling one or more IP addresses is called the master router, and forwards packets sent to these IP addresses. The election process provides dynamic fail-over in the forwarding responsibility should the master become unavailable. Any of the IP addresses on a virtual router can then be used as the default first hop router by end-hosts. The advantage gained from using the VRRP is a higher availability default path without requiring configuration of dynamic routing or router discovery protocols on every end-host.

In order to use and configure VRRP you must first define a Virtual Router ID (VRID) on each LAN segment running VRRP. The VRID is a number in the range of 1 to 255. This VRID identifies the routers that will back one another up. Therefore, all VRRP routers that are backups for one another must have the same VRID. For each VRRP segment, one router called the master router owns the default IP address configured for hosts on the LAN segment. As long as the master is available, it responds to ARP requests for that address and forwards packets. One of the backup routers takes the place of the master router if the master router becomes unavailable. When a backup router takes over, it becomes accessible at the default IP address so that the hosts now use it as the master router.

The VRID represents a unicast or multicast virtual MAC address. You can configure the backup routers with a virtual MAC address or configure each VRRP router to use its own unique burned-in hardware MAC address. If you use the multicast option, you cannot use the hardware MAC address. If you use the hardware MAC address, the hosts that communicate with the VRRP router must support gratuitous ARPs. Using the hardware MAC address can provide improved performance in your network.

The following example illustrates a very simple VRRP topology. In this example, the virtual MAC address is used. If the hardware MAC address were used, the master router and the backup router would each use its own hardware MAC address.

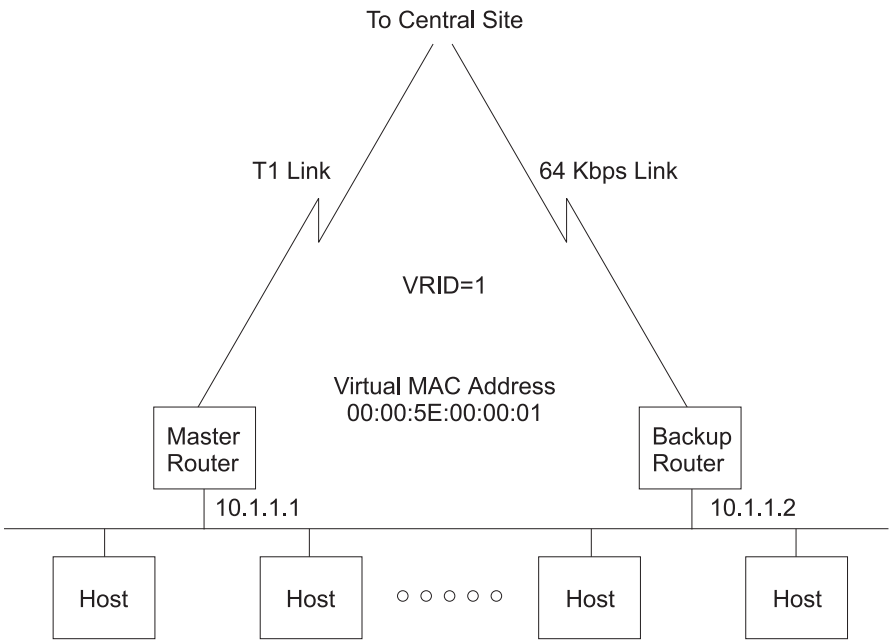


Figure 32. Ethernet LAN with subnet 10.1.1.0/255.255.255.0 All Host Configured with Default Gateway 10.1.1.1

1. All hosts are configured with default gateway of 10.1.1.1
2. The master router will answer to all ARP requests for 10.1.1.1 with the virtual MAC address of 00:00:5E:00:00:01.
3. The master router will forward packets addressed to the virtual MAC address.
4. If the master router is unavailable, the backup determines this via the absence of VRRP advertisements and will commence receiving packets addressed to the virtual MAC address. The backup will also answer to ARP requests for 10.1.1.1.

A complicated topology would be one where there are multiple VRRP routers and the desire is to balance the load between the routers but still have complete backup capability. In this case 2 VRIDs would need to be defined and each router would the master for one and the backup for the other. This illustration follows:

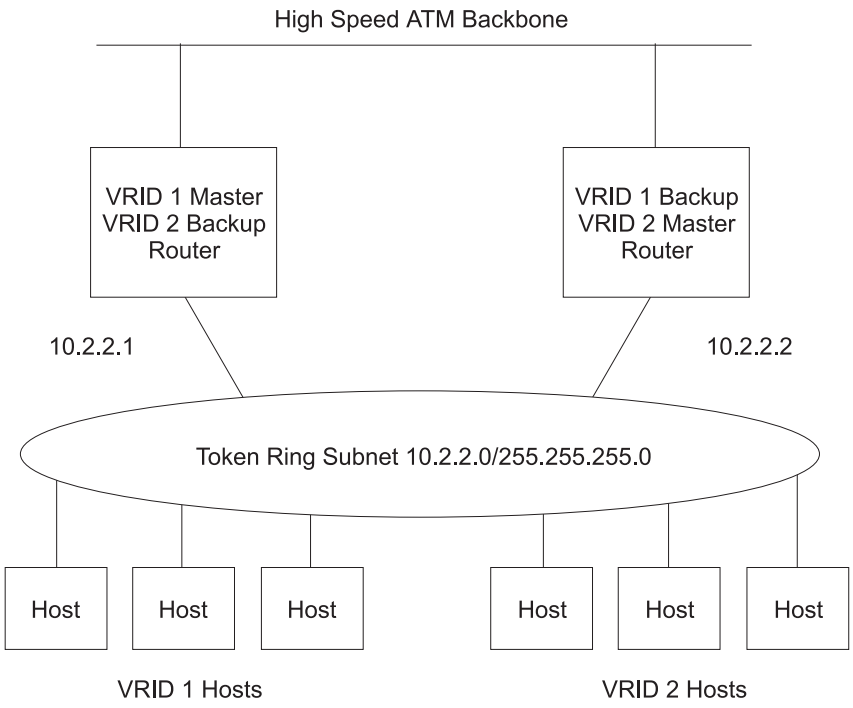


Figure 33. Multiple VRRP Routers

1. All VRID 1 hosts will be configured with a default gateway address of 10.2.2.1.
2. All VRID 2 hosts will be configured with a default gateway address of 10.2.2.2.
3. The VRID 1 master router will respond to ARP requests for address 10.2.2.1 with the virtual MAC address C0:00:00:10:00:00. It will also receive and forward packets addressed to virtual MAC address C0:00:00:10:00:00.
4. The VRID 2 master router will respond to ARP requests for address 10.2.2.2 with the virtual MAC address C0:00:00:20:00:00. It will also receive and forward packets addressed to virtual MAC address C0:00:00:20:00:00.
5. If either router becomes unavailable, the other will take over.
6. If a router does not become unavailable but loses it's outside connectivity, it will re-direct traffic through the other with ICMP redirects (this assumes the 2 routers are exchanging routes via routing protocol such as RIP or OSPF).

VRRP is supported on Ethernet, Fast Ethernet, and Token Ring.

Multicast VRRP is not supported on the bridge network when source-routed LANs are part of the bridged network. The restriction is only applicable in topologies where IP is configured on the bridge network.

Configuring the Redundant Default IP Gateway

This section outlines the steps used to configure redundant default IP gateways on ELANs. Configuration of a redundant gateway allows end stations with manually configured default gateways to continue passing traffic to other subnets after their primary gateway goes down.

To configure a device with a primary gateway or backup gateway:

1. Determine the IP address end stations use as the default gateway.

2. Determine a MAC address not used by any interfaces on the ELAN. To determine which MAC addresses are used, see “Database List” in the “Monitoring LAN Emulation Services” chapter of *Software User’s Guide*.
3. Select a device to have the primary gateway. This device must have a LEC interface on the ELAN of the end station.
4. Select a device or set of devices to have the backup gateway. This device or set of devices must have a LEC interface on the ELAN of the end station.
5. Config a redundant gateway on each device using the “Add” option for IP.

Note: The primary gateway and the backup gateway must have the same MAC 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 to 239). 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 327.

DVMRP is a multicast routing protocol derived from the Routing Information 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 device’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 25. Using DLSw” on page 481.

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.

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
```

The 2210 supports IVMP version 2 and DVMRP version 3. IGMP can be configured to operate in version 1 mode.

Refer to the discussion on configuring DVMRP in *Protocol Configuration and Monitoring Reference Volume 1* 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.

Using Simple-Internet-Access

Simple Internet Access is a means of quickly configuring many of the options required to give Internet access to a group of Dynamic Host Configuration Protocol (DHCP) clients. Enabling this option, and adding a LAN interface are all that are required to configure IP. When it is combined with a PPP interface configured to access an Internet Service Provider (ISP) account, multiple DHCP clients can access the Internet using a single public IP address. This is accomplished using both the DHCP Server feature and Network Address Translation (NAT) feature.

Note: This option will only be available on router software loads which include both the DHCP feature and the NAT feature. If similar connectivity to the Internet is required on loads that do not include the DHCP Server feature but do include the NAT feature, Dynamic-Address (see “Using Dynamic Address” on page 214) should be used with a configuration that is similar to the one shown in the example.

Example:

- If PPP is configured to request an IP address on interface 3, as follows:

```
PPP 3 Config>set ipcp
IP COMPRESSION [no]:
Request an IP address [no]: yes
Interface remote IP address to offer if requested (0.0.0.0 for none) [0.0.0.0]?
```

- Simple Internet Access can be enabled as follows:

```
IP config>enable simple-internet-access
Interface to Service Provider [0]? 3
SIMPLE-INTERNET-ACCESS enabled on interface 3

IP config>add address
Which net is this address for [0]? 0
New address []? 192.168.8.1
Address mask [255.255.255.0]?

IP config>list address
IP addresses for each interface:
intf  0   192.168.8.1      255.255.255.0    Local wire broadcast, fill 1
intf  1                                     IP disabled on this interface
intf  2                                     IP disabled on this interface
intf  3   0.0.0.3          255.255.255.255  Local wire broadcast, fill 1
                                            SIMPLE-INTERNET-ACCESS Enabled
```

- The following Packet Filters in the IP configuration will automatically be generated:

```
IP config>list packet-filter
List of packet-filter records:
Name               Direction   Interface   State  Src-Addr-Ver
simple-in          In          3           On     Off
simple-out         Out         3           On     N/A
Access Control is: enabled
```

- The following Access Controls in the IP configuration will automatically be generated:

```
IP config> list packet-filter simple-in

Name               Direction      Interface      State     Src-Addr-Ver
simple-in          In             3              On        Off
Access Control is: enabled
Access Control facility: USER

List of access control records:
1   Type=IN   Source=0.0.0.0          Dest  =0.0.0.0          Prot=  0-255
              SMask =0.0.0.0          DMask =0.0.0.0
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N

IP config>list packet-filter simple-out

Name               Direction      Interface      State     Src-Addr-Ver
simple-out         Out            3              On        N/A
Access Control is: enabled
Access Control facility: USER

List of access control records:

1   Type=IN   Source=0.0.0.0          Dest  =0.0.0.0          Prot=  0-255
              SMask =0.0.0.0          DMask =0.0.0.0
              SPorts=    0-65535      DPorts=    0-65535
                      T/C= **/**      Log=N
```

- The following Static Route in the IP configuration will automatically be generated:

```
IP config>list routes

route to 0.0.0.0        ,0.0.0.0         via 0.0.0.3         cost 1
```

- The following NAT configuration will automatically be generated:[3]

```
NAT config>list all

NAT Globals:
Current State    TCP Timeout    Non-TCP Timeout
ENABLED             24:00:00            0:01:00

NAT Reserve Pool(s):
Index   First Address       Reserve Mask     Size    NAPT Address  Pool Name
    1         Dynamic    255.255.255.255     1       FromNet:    3 simple-net

NAT Translate Range(s):
Index    Base Address      Range Mask  Associated Reserve Pool
    1     192.168.8.0   255.255.255.0  simple-net

NAT Static Mapping(s):
Index  Private Address//Port  Public Address//Port
None.
```

- The following DHCP Server configuration will automatically be generated: [3]

```
DHCP Server enabled: Yes

DHCP Server config>list subnet all
subnet      subnet           subnet           starting         ending
name        address          mask             IP Addr          IP Addr
-------------------------------------------------------------------------
simple-net  192.168.8.0      255.255.255.0    192.168.8.2      192.168.8.50

DHCP Server config>list option subnet
Enter the subnet name []? simple-net
option  option
code    data
---------------------------------------------------------------
1       255.255.255.0
3       192.168.8.1
6       0.0.0.3
```

3. If the DHCP Server has already been configured prior to enabling Simple-Internet-Access in IP, there will be no DHCP configuration generated or changed. The existing DHCP subnets will be used as the translation ranges for the NAT configuration.

Chapter 15. Configuring and Monitoring IP

This chapter describes the IP configuring and monitoring commands. It includes the following sections:

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 describes the 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 18. IP Configuration Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
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, the various RIP flags controlling the sending and receiving of RIP information, diffserv, bypass of Layer 2 compression and encryption in access control records, and route-table-filtering.
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 TTL (time-to-live) of packets originated by the router, the size of the IP routing table, cache size, and RIP interface metrics and sets IGMP configuration parameters.
Update	Used to assign access control entries to packet filters.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

Add

Use the **add** command to add IP information to your configuration.

Syntax:

add	accept-rip-route . . .
	access-control . . .
	address . . .
	aggregate . . .
	bootp-server
	filter . . .
	packet-filter
	redundant default gateway
	route . . .
	route-policy . . .
	route-table-filter
	udp-destination . . .
	vrid . . .
	vr-address . . .

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 and subnets that have already been entered using the **list rip** 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 receiving dynamic nets** or **disable receiving dynamic subnets** or **disable receiving dynamic hosts** commands.

IP network/subnet

Valid Values: any valid IP address

Default Value: none

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-dest-port last-dest-port first-source-port last-source-port] [tcp-syn] [icmp-type icmp-code] [tos-mask tos-range-low tos-range-high tos-mod-mask new-tos-value policy-based-routing next-hop-gateway use-default-route] [log els snmp-trap syslog syslog-level]*

From the `IP config>` prompt, use this command to add an access control record to the end of the global access control list. From the Packet-filter *packet-filter-name* Config> prompt, use this command to add an access

control rule to the end of the packet filter access control list. Access control allows you to define categories of packets to forward, to drop, or to process with network address translation, based on packet values specified in the access control rules. The length and order of the access control lists can affect the IP packet forwarding performance.

Note: The **add access-control** command configures access control rules, but it does not automatically enable access control; see the **set access-control** command.

type Indicates what is done with packets that match the access control rule parameters.

E Exclusive; matching packets are discarded.

I Inclusive; matching packets are processed further by the router.

N Network address translation (NAT); matching packets are passed to NAT for address translation. This type is valid only when specified in combination with inclusive, for example, *IN*. This parameter is valid only in the packet filter configuration console (accessed by the **update packet-filter** command).

IP-source source-mask
Source IP address and mask. The source-mask is bit-ANDed with the received source IP address to allow the rule to match a range of source IP addresses. Where bits of the source mask are 0, the corresponding bits of the IP source address must also be 0.

Valid Values: 0.0.0.0 to 255.255.255.255

Default Values: 0.0.0.0 for source IP address. The default for the source-mask is based on the configured IP-source address.

IP-dest dest-mask
Destination IP address and mask. The dest-mask is bit-ANDed with the received destination IP address to allow the rule to match a range of destination IP addresses. Where bits of the destination mask are 0, the corresponding bits of the destination IP address must also be 0.

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: 0.0.0.0 for destination IP address. The default for the dest-mask is based on the IP-dest address.

first-protocol last-protocol
A range of IP protocol numbers.

Some common IP protocol numbers are:
1 for ICMP
6 for TCP
17 for UDP
89 for OSPF

Valid Values: 0 to 255

Default Values: 0 for first protocol and 255 for last protocol

first-dest-port last-dest-port
A range of TCP/UDP destination port numbers. These parameters

are valid only if the range of IP protocol numbers includes 6 (for TCP) or 17 (for UDP). These parameters are ignored for packets in which the IP protocol number is not 6 or 17.

Some commonly used port numbers are:
21 for FTP
23 for Telnet
25 for SMTP
513 for rlogin
520 for RIP

Valid Values: 0 - 65535

Default Value: 0 for first destination port and 65535 for last destination port

first-source-port last-source-port
A range of TCP/UDP source port numbers. These parameters are valid only if the range of IP protocol numbers includes 6 (for TCP) or 17 (for UDP). These parameters are ignored for packets in which the IP protocol number is not 6 or 17. See the description of *first-dest-port last-dest-port* for a list of commonly used TCP/UDP port numbers.

Valid Values: 0 - 65535

Default Value: 0 for first source port and 65535 for last source port

tcp-syn
This parameter matches TCP packets that establish TCP connections (that is, TCP packets in which the SYN bit is 1 and the ACK bit is 0). This parameter is valid only if the range of IP protocol numbers includes 6 (for TCP) and the rule type is exclusive. This parameter is invalid for types IPSec and NAT, which are always inclusive. This parameter is ignored for packets in which the IP protocol number is not 6.

Valid Values: Yes or No

Default Value: No

icmp-type
This parameter, which defines the ICMP type, is valid only if the range of IP protocol numbers includes 1 (for ICMP). The value of this parameter defines the ICMP type of the access rule. ICMP packets can match the access rule only if the ICMP type of the packet matches the ICMP type of the access rule. If the default value -1 is specified, all ICMP type values are treated as matching the access rule. This parameter is ignored for packets in which the IP protocol number is not 1.

Valid Values: -1 to 255

Default Value: -1 (all ICMP types)

icmp-code
This parameter, which defines the ICMP code, is valid only if the range of IP protocol numbers includes 1 (for ICMP). The value of this parameter defines the ICMP code of the access rule. ICMP packets can match the access rule only if the ICMP code of the packet matches the ICMP code of the access rule. If the default

value -1 is specified, all ICMP code values are treated as matching. This parameter is ignored for packets in which the IP protocol number is not 1.

Valid Values: -1 to 255

Default Value: -1 (all ICMP codes)

tos-mask, tos-range-low, tos-range-high

Setting *tos-mask* to a non-zero value enables filtering according to bits in the TOS byte. *Tos-mask* identifies the bits in the precedence/TOS byte that are to be filtered. For example, if the *tos-mask* is X'E0' (B'11100000'), filtering applies only to the 3 precedence bits in the TOS byte (the 3 most significant bits of the TOS byte).

The *tos-range-low* and the *tos-range-high* define the range of consecutive values within the selected bits. If you want to filter all 8 values of the precedence bits (decimal 0 - 7), the *tos-range-low* is X'00' (B'00000000') and the *tos-range-high* is X'e0' (B'11100000', which defines decimal 7 within the 3 bits that are selected for filtering). If you want to filter the binary values B'000', B'001', B'010', and B'011' (decimal 0 - 3) of the 3 precedence bits, the *tos-range-low* is X'00' (B'00000000') and the *tos-range-high* is X'60' (B'01100000').

If you need to filter bit patterns that do not form a consecutive sequence of values, you need to define a separate access control rule for each range desired. For example, to filter the two precedence bit values B'001' (decimal 1) and B'011' (decimal 3) without filtering B'010' (decimal 2), you would have to define the first access control rule with *tos-mask* equal to X'e0' and *tos-range-low* and *tos-range-high* both equal to X'20'. Then you would have to define the second access control rule with *tos-mask* equal to X'e0' and *tos-range-low* and *tos-range-high* both equal to X'60'.

Valid Values for *tos-mask*: X'00' - X'FF'

Default Value: 0 for none

Valid Values for *tos-range-low*: X'00' - X'FF'

Default Value: 0

Valid Values for *tos-range-high*: X'00' - X'FF'

Default Value: The configured *tos-range-low.*

new-tos-value, tos-mod-mask

Setting these parameters enables the router to modify specified bits in the TOS byte. The *tos-mod-mask* identifies the bits within the TOS byte that are to be changed. The *new-tos-value* defines the new value for the selected bits. For example, if the *tos-mod-mask* is X'1e' and the *new-tos-value* is X'00', the 4 bits of the TOS field (identified within the byte by the *tos-mod-mask* value X'1e' [B'00011110']) are set to B'0000'. To set the TOS bits to the value for maximum throughput (B'0100'), use the *tos-mod-mask* X'1e' and the *new-tos-value* X'08' (B'00001000').

Valid Values for *tos-mod-mask*: X'00' - X'FF'

Default Value: 0 for none

Valid Values for *new-tos-value*: X'00' - X'FF'

Default Value: 0

policy-based-routing, next-hop-gateway, use-default-route
These parameters enable policy based routing, which is the ability to specify the next hop gateway to which the filtered packets will be sent. Setting the *policy-based-routing* parameter to Yes indicates that you plan to have the filtered packets sent to the defined next hop gateway. *Next-hop-gateway* is the address of the next hop gateway to which these packets will be sent.

Setting *use-default-route* to Yes enables the router to route the packet using the normal routing table if the defined gateway becomes unavailable. If this parameter is set to No, the packet is discarded if the defined gateway becomes unavailable and an ICMP *unreachable* message is sent to the source address of the discarded packet.

Valid Values for *policy-based-routing*: Yes or No

Default Value: No

Valid Value for *next-hop-gateway*: a valid IP address

Default Value: none

Valid Value for *use-default-route*: Yes or No

Default Value: Yes

log Enables logging.

Valid Values: Yes or No

Default Value: No

els If logging is enabled, enables ELS messages for this access control rule.

Valid Values: No, short, or long

Default Value: No

snmp-trap
If logging is enabled, enables the sending of SNMP traps for this access control rule.

Valid Values: Yes or No

Default Value: No

syslog
If logging is enabled, enables SysLog for this access control rule. SysLog posts system messages to an attached remote workstation.

Valid Values: No, short, or long

Default Value: No

syslog-level
If SysLog is enabled, specifies the level of the SysLog messages.

Valid Values: Sys Def, Emerg, Alert, Crit, Error, Warn, Notice, Info, or Debug

Default Value: Router system default value

Example:

```
IP config> add access-control
Enter type [E] I
Internet source [0.0.0.0]?
Source mask [0.0.0.0]?
Internet destination [0.0.0.0]?
Destination mask [0.0.0.0]?
Enter starting protocol number ([CR] for all) [-1]?
Enter starting destination port number ([CR] for all) [-1]?
Enter starting source port number ([CR] for all) [-1]?
Enter ICMP Type ([CR] for all) [-1]? 3
Enter ICMP Code ([CR] for all) [-1]?
TOS/Precedence filter mask (00-FF - [0] for none) [0]? CD
TOS/Precedence start value (00-FF) [0]?
TOS/Precedence end value [0]?
TOS/Precedence modification mask (00-FF - [0] for none) [0]? FA
New TOS/Precedence value (00-FF) [0]?
Next hop gateway address [ ]? 8.8.8.2
Use default route if next hop gateway unreachable? [Yes]:
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*.

Note: To assign an IP address to the 2210's bridge network, specify **bridge** for the *interface number*. See "Assigning IP Addresses to the Bridge Network Interface" on page 214 for more information.

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, or **bridge**

Default Value: none

ip-address

Valid Values:

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 unnumbered serial line interfaces, 0.0.0.n, where *n* is the interface number

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`

aggregate *aggregate-IP-address aggregate-IP-mask unconditional-aggregate route-policy-ID metric*

Adds an aggregate route to the IP configuration. Route aggregation enables you to use one IP address and subnet mask to define a range of subnets that belong to different routing domains. See “Route Aggregation” on page 230 for more information.

aggregate IP address

Valid Values: Any valid IP address

Default Value: none

aggregate IP mask

Valid Values: Any valid IP mask

Default Value: none

unconditional aggregate

Answering **yes** to this question indicates that the aggregate route is generated unconditionally. If it is generated unconditionally, it will be advertised when it is needed. A metric value is associated with an unconditional aggregate route.

Answering **no** indicates that the aggregate route is generated conditionally. If an aggregate route is generated conditionally, it is generated only when the destination address to be processed matches the range specified by the aggregate route and also meets the conditions of a specified route filter policy.

Valid Values: Yes or no

Default Value:No

route policy identifier

This is the route filter policy identifier of the route filter policy associated with the aggregate route. This parameter applies only when the aggregate route is conditional.

Valid Values: A valid route filter policy identifier (1 to 15 ASCII characters)

Default Value: None. This parameter is required when *No* is specified or defaulted for unconditional aggregate.

metric This parameter is a metric to measure the cost of the aggregate route in relation to the cost of other aggregate routes. It is applied only to unconditional aggregate routes.

Valid Values: 1-65535

Default Value: 1

Example:

```
IP config> add aggregate
IP Address [ ]? 10.0.0.0
IP Mask [ ]? 255.0.0.0
Aggregate Route Always Generated? [No]:yes
Metric value [1-16777215] [1]?
Route aggregate 10.0.0.0/255.0.0.0 added or modified.
```

bootp-server *server-IP-address*

Adds a BOOTP/DHCP server to the list of servers to which the router will forward BOOTP/DHCP requests. See “Configuring the BOOTP/DHCP Forwarding Process” on page 232 for more information.

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 interface-number*

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.

interface-number

Valid Values: any defined interface, or **bridge** for the Bridge Network 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
```

redundant default gateway *interface-number gateway-IP-address address-mask MAC-address primary-gateway*

Adds a Redundant Default Gateway IP address to your configuration.

interface-number

Specifies the net number of LEC interfaces on the ELAN.

Valid Values: net numbers of LEC interfaces

Default Value: none

gateway-IP-address
Specifies the Default Gateway of the end station.

Valid Values: IP addresses used as default gateways

Default Value: 0.0.0.0

address-mask
Specifies the mask of the IP address.

Valid Values: any valid IP net mask

Default Value: 0.0.0.0

MAC-address

Note: The primary gateway and the backup gateway must have the same MAC address
Valid Values: any valid MAC address not used by other interfaces on the ELAN

Default Value: 00.00.00.00.00.00

primary-gateway
Specifies whether the gateway is used as the primary or as the backup gateway.

This query asks whether the gateway on this device is the primary gateway active during the normal operation of the network, or the backup gateway that is active when the LEC interface containing the primary gateway is not operational. Answering **Yes** configures a primary gateway. There should be only one primary gateway per ELAN.

Valid Values Yes or No

Default Value: No

Example: `add redundant`

```
Which net is this redundant gateway for [0]? 1
IP address of gateway [0.0.0.0]? 9.67.205.1
Address mask [255.255.0.0]? 255.255.240.0
MAC address [00.00.00.00.00.00.]? 00.00.00.00.00.BA
Is this the primary gateway [No]? Yes or No
```

route *dest-addr dest-mask next-hop1 cost1 [next-hop2 cost2 [next-hop3 cost3 [next-hop4 cost4]]]*
Adds 1 to 4 static routes to the device'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 (*dest-addr*) together with an address mask (*dest-mask*). If the destination IP address is a network address, then the dest-mask must be a network mask. If the destination IP address is a subnet address, then the dest-mask must be a subnet mask. Finally, if the destination IP address is a host address, then the dest-mask must be a host mask (which means that the only valid value is 255.255.255.255). The dest-mask must be accurate; if it is not, the static route will not be accepted.

You may define up to four static routes per destination, each route having an IP address of its next hop (next-hop) and a cost (cost) of routing a 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 to a given destination may have the same cost, in which case IP may use the routes simultaneously, or they may have different costs, in which case IP uses the lowest cost route that works.

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 **add route** command takes effect immediately; you do not have to reboot the router.

dest-addr

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

next-hop1, next-hop2, next-hop3, next-hop4

Valid Values: any valid IP address

Default Value: none

cost1, cost2, cost3, cost4

Valid Values: an integer in the range 0 to 255

Default Value: 1

Example:

```
IP config> add route
IP destination []? 1.1.0.0
Address mask [255.0.0.0]? 255.255.0.0
Via gateway 1 at []? 10.1.1.1
Cost [1]? 1
Via gateway 2 at []?
IP config> add route 1.1.0.0 255.255.0.0
Via gateway 2 at []? 20.1.1.1
Cost [1]? 2
Via gateway 3 at []? 30.1.1.1
Cost [1]? 3
Via gateway 4 at []?
IP config> add route 2.2.0.0 255.255.0.0 10.2.2.2 1 20.2.2.2 2
IP config> list routes

route to 1.1.0.0        ,255.255.0.0     via 10.1.1.1        cost 1
                                         via 20.1.1.1        cost 2
                                         via 30.1.1.1        cost 3
route to 2.2.0.0        ,255.255.0.0     via 10.2.2.2        cost 1
                                         via 20.2.2.2        cost 2

IP config>
```

route-policy *route-policy-identifier use-strictly-linear-policy*

Adds a route filter policy. A route filter policy consists of entries that define a set of routes that can be filtered to be included or excluded from the routing table of an external routing protocol such as OSPF or RIP.

route-policy-identifier

A string that identifies a route filter policy.

Valid Values: any 1-to-15-character ASCII string

Default Value: none

use-strictly-linear-policy

Yes indicates that matching will be done based strictly upon the sequence of index numbers of the route filter policy entries. The entry with the lowest index number will be processed first. *No* indicates that matching will be done using the longest-match application. The entry with the lower index number will be chosen only when more than one entry has the same address and mask.

Valid Values: Yes or No

Default Value: No

route-table-filter *destination mask [both | exact | more-specific] [exclusive | inclusive]*

Adds a route table filter for the specified routes. When **route-table-filtering** is enabled, the route-table-filter will be matched against routes added to the IP route table. The order in which route-table-filters is unimportant. Rather, the route-table-filter with the most specific match is chosen. If no match is found, the route is added to the route table. When **exact** is specified, the route destination and mask must be exactly the same as the route-table-filter destination and mask for a match to occur. When **more-specific** is specified, the route destination and mask must part of the range subsumed by the route-table-filter destination and mask. Specifying **both** is the superset of both and more-specific (that is, a match will occur in both the case of an exact match and a more-specific match). If the route-table-filter indicates **include**, the route will be added to the IP route table. If the route-table-filter indicates **exclude**, the route will not be added to the IP route table. Static and direct routes are never excluded from the IP route table.

destination mask

Valid Values: any valid IP mask

Default Value: both exclude

udp-destination *port-number address*

Adds a UDP forwarding destination address. Received UDP datagrams with the specified destination UDP port number will be forwarded to the specified 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 UDP datagram 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
```

vrid ... Adds a Virtual Router ID definition for a VRRP router on a LAN segment.

interface-ip-address

Indicates the IP interface for which this VRID is being defined.

Valid Values: Any configured IP interface.

Default Value: none

vrid
The Virtual Router identifier. The combination of the *ip-interface-address* and *vrid* uniquely define the VRID. The same *vrid* can be used on more than one physical interface. If the VRID already exists, it will be modified.

Valid Values: 1-255

Default Value: none

advertisement-interval
The interval between VRRP advertisements.

Valid Values: 1-255

Default Value: 1

backup-router
Indicates whether this router is the master or a backup router for this VRID.

Valid Values: Yes or No

Default Value: No

backup-ip-address
Indicates the first IP address that is the backup for this VRID. Additional addresses may be added using the *add vr-address* command for LAN segments supporting more than one subnet. It is not applicable if **No** was configured for *backup-router*.

Valid Values: Any valid IP address.

Default Value: none

priority
Indicate the VRRP priority for backup routers. If a backup router takes over for the primary router, it will use this priority in its VRRP advertisements. It is not applicable if **No** was configured for *backup-router*. A master router will always advertise a priority of 255.

Valid Values: 1-254

Default Value: 100

preempt-mode
Indicates that a VRRP backup router will preempt a lower-priority backup VRRP. Preempt means to take over as the router supporting forwarding for VRID. It is not applicable if **No** was configured for *backup-router*. A master router will always preempt a backup router that has temporarily taken over responsibilities for the VRID.

Valid Values: Yes or No

Default Value: Yes

hardware MAC mode
Indicates whether or not the hardware MAC address is used as the VRID virtual MAC address. All routers configured for this VRID should have the same value for this parameter in order for VRRP to

function correctly. If **Yes** is entered for this parameter, *functional/group mode* defaults to **No** and is not displayed.

functional/group mode
Indicates whether or not a multicast MAC address is used as the VRID virtual MAC address. All routers configured for this VRID should have the same value for this parameter in order for VRRP to function correctly. This parameter defaults to **No** and is not displayed if *hardware MAC mode* is configured as **Yes**.

Valid Values: Yes or No

Default Value: No

authentication-type
Indicates the type of authentication used for VRRP advertisements. The choices for authentication types are 1, which indicates a simple password; or 0, which indicates that no authentication is used.

Valid Values: none, simple

Default Value: none

authentication-key
The parameter that defines the password for this VRID. When password authentication is used, only packets with the correct authentication key are accepted. The *authentication key* is not applicable when *none* is specified or defaulted for *authentication type*.

Valid Values: Any 1 - 8 characters.

Default Value: A null string.

require network for master eligibility
Indicates whether or not a network's operational status (up or down) will be monitored to determine master eligibility. If this parameter has the value **yes**, when the network goes down and the router is the VRRP master, the router will relinquish master status and a new master will be elected.

Valid Values: Yes or No

Default Value: No

required network
The network that will be monitored to determine master eligibility. This question is not applicable if **No** is specified for *require network for master eligibility*.

Valid Values: 1 to the numbers of the configured networks

Default Value: No

required route destination
The route destination that will be monitored to determine master eligibility. This question is not applicable if **No** is specified or defaulted for *require route for master eligibility*.

Valid Values: Any valid IP address

Default Value: 0.0.0.0

required route mask
The route mask that will be monitored to determine master

eligibility. This question is not applicable if **No** is specified or defaulted for *require route for master eligibility*.

Valid Values: Any valid IP mask

Default Value: 0.0.0.0

Example:

```
IP config>add vrid
IP Interface []? 153.2.2.1
VRID (1-255) [0]? 1
Advertisement Interval (1-255) [1]?
Backup Virtual Router? [No]:
Use Hardware MAC Address? [No]: yes
Authentication Type (0 - None, 1 - Simple) [0]?
Require network for master eligibility? [No]:
Require route for master eligibility? [No]:
VRID 153.2.2.25/1 added/modified successfully
```

vr-address ...

Adds a secondary address to a configured Virtual Router ID (VRID) definition. Secondary addresses will be included in VRRP advertisements for the VRID. Secondary addresses are necessary on physical LANs supporting multiple IP subnets. Each address designates the default gateway address for that subnet. If the router is a master router, addresses added using the *add vr-address* command will be advertised in addition to the *ip-interface-address* for the VRID. If the router is a backup router for the VRID, addresses added using the *add vr-address* command will be advertised in addition to the *backup-ip-address*.

interface-ip-address

The IP interface for the VRID.

Valid Values: Any configured IP interface.

Default Value: none

vrid

The Virtual Router identifier. The combination of the *ip-interface-address* and *vrid* uniquely define the VRID. The VRID must be configured for addresses to be added to its definition. A master router and its backup routers must both be configured with the same VRID.

Valid Values: 1 to 255

Default Value: none

ip-address

The additional IP address that will be included in VRRP advertisements for the VRID.

Valid Values: Any IP address.

Default Value: none

Example: `add vr-address`

```
IP config>add vr-address
IP Interface [ ]? 153.2.2.25
Virtual Router ID (1-255) [0]? 1
Additional IP Address [ ]? 5.1.1.1
VRID 153.2.2.25/1 address 5.1.1.1 added successfully.
```

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 . . .
route-policy

access-control *rule-number type IP-source source-mask IP-dest dest-mask first-protocol last-protocol [first-dest-port last-dest-port first-source-port last-source-port] [tcp-syn] [icmp-type icmp-code] [tos-mask tos-range-low tos-range-high tos-mod-mask new-tos-value policy-based-routing next-hop-gateway use-default-route] [log els snmp-trap syslog syslog-level]*

Modifies an existing global access-control record. Use the **list access-control** command to view all existing records and obtain the rule number. See the talk 6 **Add** command for definitions of the parameters.

Example:

```
IP config> change access-control 2
Enter type [E]? i
Internet source [9.1.2.3]?
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]?
Destination mask [0.0.0.0]?
Enter starting protocol number [0]?
Enter starting DESTINATION port number [0]?
Enter starting SOURCE port number [0]?
Filter on ICMP Type [-1]?
TOS/Precedence filter mask [e0]?
TOS/Precedence start value [0]?
TOS/Precedence end value [0]?
TOS/Precedence modification mask [1f]? 1e
New TOS/Precedence value[0]? 08
Use policy-based routing? [Yes]:
Next hop gateway address [9.2.160.1]?
Use default route if next hop gateway unreachable? [Yes]:
Enable Logging [No]:
```

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.

Valid IP addresses:

- 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 unnumbered serial interfaces, 0.0.0.n, where n is the hardware interface number

For serial line interfaces:

- 0.0.0.n, where n is the hardware interface number.

old-address

Valid Value: a currently configured IP interface address

Default Value: none

new-address

Valid Value: any valid IP address

Default Value: none

new-mask

Valid Value: 0.0.0.0 - 255.255.255.255

Default Value: none

Example: `change address 192.9.1.1 128.185.123.22  255.255.255.0`

route *dest-addr dest-mask new-next-hop1 new-cost1 [new-next-hop2 new-cost2 [new-next-hop3 new-cost3 [new-next-hop4 new-cost4]]]*

Modifies either the next hops or the costs associated with the configured static routes to the specified destination. The effect of this command is immediate; you do not have to reboot the router for it to take effect.

dest-addr

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

new-next-hop1, new-next-hop2, new-next-hop3, new-next-hop4

Valid Values: any valid IP address

Default Value: none

new-cost1, new-cost2, new-cost3, new-cost4

Valid Values: an integer in the range 0 to 255

Default Value: 1

Example:

```
IP config>list routes

route to 1.1.0.0        ,255.255.0.0     via 10.1.1.1        cost 1
                                         via 20.1.1.1        cost 2
                                         via 30.1.1.1        cost 3
route to 2.2.0.0        ,255.255.0.0     via 10.2.2.2        cost 1
                                         via 20.2.2.2        cost 2

IP config>change route
IP destination []? 1.1.0.0
Address mask [255.0.0.0]? 255.255.0.0
Via gateway 1 at [.10.1.1.1]? 10.10.10.1
Cost [1]? 10
Via gateway 2 at [20.1.1.1]? 20.20.20.1
Cost [2]? 20
Via gateway 3 at [30.1.1.1]? 30.30.30.1
Cost [3]? 30
Via gateway 4 at []? 40.40.40.1
Cost [1]? 40
IP config>change route 2.2.0.0 255.255.0.0 10.10.10.2 10
IP config>list routes

route to 1.1.0.0        ,255.255.0.0     via 10.10.10.1      cost 10
                                         via 20.20.20.1      cost 20
                                         via 30.30.30.1      cost 30
                                         via 40.40.40.1      cost 40
route to 2.2.0.0        ,255.255.0.0     via 10.10.10.2      cost 10
```

route-policy *route- policy-identifier*

Use this command to change an existing route filter policy, which was created using the **add route-policy** command. You use the **change route-policy** command to configure the entries, actions, and match-conditions associated with the route filter policy. The **change route-policy** command brings up the `IP Route Policy Config>` prompt.

route-policy-identifier

Valid Values: the 1 to 15-character ASCII string that identifies an existing route filter policy

Default Value: none

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 . . .
	aggregate . . .
	bootp-server
	default network/subnet-gateway . . .
	filter . . .
	packet-filter
	redundant default gateway
	route . . .
	route-policy . . .
	route-table-filter
	udp-destination . . .
	vrid . . .
	vr-address . . .

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 *rule-number*

Deletes one of the access control rules 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`

aggregate *ip-address ip-mask*

Deletes one of the router's aggregate routes, defined by the IP address and IP mask of the aggregate route to be deleted.

aggregate IP address

Valid Values: Any valid IP address

Default Value: none

aggregate IP mask

Valid Values: Any valid IP mask

Default Value: none

bootp-server *server-IP-address*

Removes a BOOTP server from an IP configuration.

Valid Values: any configured BOOTP server IP address

Default Value: 0.0.0.0

Example: **`delete bootp-server 128.185.123.22`**

default network/subnet-gateway *[ip-network-address]*

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 *dest-addr dest-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.

dest-addr

Valid Values: any valid IP address

Default Value: 0.0.0.0

dest-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:

```
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>
```

redundant *interface-number*

Deletes the Redundant IP Gateway from a LEC interface.

interface-number

Valid Values: Interface numbers of LECs with a Redundant Default IP Gateway.

Default Value: none

Example:

```
Enter the Net number of Redundant Gateway to delete:? 1
Gateway deleted.
```

route *dest-addr dest-mask [delete-next-hop1 [delete-next-hop2 [delete-next-hop3 [delete-next-hop4]]]]*

Deletes one of the device's configured static routes. The effect of this command is immediate; you do not have to reboot the router for it to take effect.

dest-addr

Valid Values: any valid IP address

Default Value: none

dest-mask

Valid Values: any valid IP mask

Default Value: none

delete-next-hop

Valid Values: Yes or No

Default Value: No

Example:

```
IP config>list routes

route to 1.1.0.0        ,255.255.0.0     via 10.10.10.1      cost 10
                                         via 20.20.20.1      cost 20
                                         via 30.30.30.1      cost 30
                                         via 40.40.40.1      cost 40
route to 2.2.0.0        ,255.255.0.0     via 10.10.10.1      cost 10

IP config>delete route 1.1.0.0 255.255.0.0
Delete gateway 10.10.10.1? [No]:
Delete gateway 20.20.20.1? [No]: y
Delete gateway 30.30.30.1? [No]:
Delete gateway 40.40.40.1? [No]: y
IP config>delete route 2.2.0.0 255.255.0.0
IP config>delete route 1.1.0.0 255.255.0.0 n y
IP config>list routes

route to 1.1.0.0        ,255.255.0.0     via 10.10.10.1      cost 10

IP config>
```

route-policy *route-policy-identifier delete-route-policy-entries*

Deletes an existing route filter policy from the configuration. You have the option to delete all route filter policy entries associated with the route filter policy. If the entries have not been deleted, when you reconfigure the deleted route filter policy, the entries associated with that route filter policy are reinstated. Use the **add route-policy** command to reconfigure a deleted route filter policy.

route policy identifier

Valid Values: the 1 to 15-character ASCII string that identifies a configured route filter policy

Default Value: none

delete-route-policy-entries

Yes deletes the corresponding route policy entries; *No* saves them.

Valid Values: Yes or No

Default Value: No

route-table-filter *destination mask mask-definition[both | exact | more specific]*

Deletes a route filter from the route table filters added using **add route-table-filter.** See "route-table-filter" on page 254 for the command extension definitions.

destination
Valid Values: any valid IP mask

Default Value: none

mask **Valid Values**: any valid IP mask

Default Value: none

mask-definition
Valid Values: any valid IP mask

Default Value: none

Example: `delete route-table-filter`

```
IP config>delete route-table-filter
Route Filter IP address []? 7.0.0.0
Route Filter IP mask []? 255.0.0.0
Enter Match type (B, E, or M) [B]?
Enter Definition type (I or E) [E]?
Route filter deleted
IP config>
```

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 0 to 65535

Default Value: none

address
Valid Values: any valid IP address

Default Value: none

Examples:
`delete udp-destination 36 20.1.2.2`

vrid *interface-ip-address vrid*
Deletes a configured Virtual Router ID definition for a VRRP router.

interface-ip-address
Indicates the IP interface for which this VRID is being deleted.

Valid Values: Any configured IP interface.

Default Value: none

vrid The Virtual Router identifier. The combination of the *ip-interface-address* and *vrid* uniquely define the VRID. It is used to identify the VRID which is going to be deleted.

Valid Values: 1-255

Default Value: none

Example:

```
IP config>delete vrid
IP Interface [ ]? 153.2.2.25
Virtual Router ID (1-255) [0]? 1
VRID 153.2.2.25/1 deleted.
```

vr-address *interface-ip-address vrid ip-address*

Deletes a secondary address from a configured Virtual Router ID (VRID) definition.

interface-ip-address

The IP interface for the VRID.

Valid Values: Any configured IP interface.

Default Value: none

vrid

The Virtual Router identifier. The combination of the *ip-interface-address* and *vrid* uniquely define the VRID. The VRID must be configured for addresses to be deleted from its definition.

Valid Values: 1-255

Default Value: none

ip-address

The additional IP address that will be deleted from the VRRP definition.

Valid Values: Any IP address.

Default Value: none

Example:

```
IP config>delete vr-address
IP Interface [ ]? 153.2.2.25
Virtual Router ID (1-255) [0]? 1
IP Address to delete [ ]? 5.1.1.1
VRID 153.2.2.25/1 addr 5.1.1.1 deleted.
```

Disable

Use the **disable** command to disable IP features previously enabled by the **enable** command.

Syntax:

disable

arp-net-routing
arp-subnet-routing
bootp-forwarding
classless
compression-bypass
directed-broadcast
dynamic-address
echo-reply
encryption-bypass
fragment-offset-check
icmp-redirect . . .
nexthop-awareness . . .
override default/static-routes . . .
packet-filter
per-packet-multipath

receiving policy . . .

receiving rip . . .

receiving dynamic all/hosts/nets/subnets . . .

record-route

rip

rip2

route-table-filtering

same-subnet

sending all/default/net/subnet/poisoned/host/static/...

sending outage-only . . .

sending policy . . .

sending rip1-routes-only

simple-internet-access

source-addr-verification

source-routing

tftp-server

timestamp

trace

udp-forwarding . . .

vrrp . . .

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`

classless

Disables the suppression of classful network routes. Classful network routes (for example, class A, B, or C routes) will be automatically generated for advertisement in protocols that do not advertise the subnet mask (for example, RIPv1).

compression-bypass *rule-number*

Disables bypassing of Layer 2 compression for the access control specified by *rule-number*, causing packets to be compressed if compression is

enabled on the output network interface. Compression-bypass may only be disabled for access controls of type **include**.

Example: `disable compression-bypass 1`

directed-broadcast

Disables the forwarding of IP packets whose destination is a non-local (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`

dynamic-address *0.0.0.n*

On the specified PPP network interface, disables the router from learning its IP address from the remote node on that network interface. By default, this option is disabled.

This option is valid only on a network interface that has been configured as an unnumbered serial line interface (the IP address assigned to the network interface by the **add address** command is *0.0.0.n*, where *n* is the network interface number).

Example:

```
IP config> disable dynamic-address
Interface address []? 0.0.0.1
IP config>
```

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`

encryption-bypass *rule-number*

Disables bypassing of Layer 2 encryption for the access control specified by *rule-number*, causing packets to be encrypted if encryption is enabled on the network interface. Encryption-bypass may only be enabled for access controls of type **include**.

Example: `disable encryption-bypass 1`

fragment-offset-check

Disables the checking of the fragment offset of received IP packets. When this check is enabled, the router checks each fragment to ensure that no secondary fragment has overlaid the first eight bytes of the first fragment’s payload. By default this check is disabled.

icmp-redirect *ip-interface-address*

Disables the router from sending ICMP Redirect messages on the specified IP interface. If you enter nothing at the prompt for the IP interface address, the router will be disabled from sending ICMP Redirect messages on all IP interfaces.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example:

```
IP config> disable icmp-redirect
Interface address (NULL for all) []? 192.9.200.44
IP config>
```

nexthop-awareness *ip-interface-address*

Disables nexthop awareness on an IP interface.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example:

```
IP config>disable nexthop-awareness 1.1.1.1
IP config>disable nexthop-awareness
Interface address []? 2.2.2.2
IP config>
```

override default/static-routes *ip-interface-address*

By default, routes received by RIP do not override static routes. However, the command **enable override static-routes** enables routes received by RIP to override static routes. After RIP routes have been enabled to override static routes, you can use the command **disable override default-route** or **disable override static-route** to again prevent static routes from being overridden by routes received by RIP. The command **disable override default-route** prevents a default route received by RIP on interface *ip-interface-address* from replacing a default route already installed in the IP routing table. The command **disable override static-routes** 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, equal cost multipath routes will load balance by destination when the destination is placed in the IP cache. The default is disabled.

receiving policy global/interface *ip-interface-address*

Disables the use of the policy for determination of what routes are accepted by RIP. The **disable receiving policy global** command disables the use of the global route filter receiving policy for all RIP interfaces that meet these two conditions:

- They do not have interface-level RIP route filter receiving policies configured. If an interface-level RIP route filter receiving policy is configured on an interface, that policy continues to determine which routes are accepted.
- Reception of RIP routes is not disabled.

After the global RIP route filter receiving policy has been disabled, the RIP interfaces are no longer affected by that policy.

The **disable receiving policy interface** *ip-interface-address* command disables the use of the route filter policy for the specified interface.

ip-interface-address
Valid Values: any valid IP address

Default Value: none

receiving rip *ip-interface-address*
Prevents RIP from processing any RIP updates 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 all/hosts/nets/subnets *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`

record-route
Disables the router from receiving or forwarding IP packets that contain a record route IP option. By default, the router receives and forwards these packets.

rip Turns off the RIP protocol.

Example: `disable rip`

rip2 Disables RIP2 on an IP interface on which it was previously enabled.

ip-interface-address
Indicates the IP interface on which RIP2 is disabled.

Valid Values: any valid IP address

Default Value: none

Example: `disable rip2 128.185.123.22`

route-table-filtering

Disables application of route-table-filters when routes are added to the routing table.

Example: `disable route-table-filtering`

same-subnet

Disables the same subnet option. When the router is rebooted, it will not allow multiple IP interfaces to the same subnet to be installed. This is the default.

Example: `disable same-subnet`

sending policy global/interface *ip-interface-address*

Disables the use of the route filter policy for the determination of what routes are advertised by RIP. The **disable sending policy global** disables the use of the global route filter sending policy for all RIP interfaces that meet these two conditions:

- They do not have an interface-level RIP sending route filter policy configured. If an interface-level RIP sending route filter policy is configured, that policy continues to determine which routes are advertised.
- Sending of RIP routes is not disabled.

After the RIP global sending route policy has been disabled, the RIP interfaces are no longer affected by that policy.

The **disable sending policy interface** *ip-interface-address* command disables the use of the route filter policy on the specified interface.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

sending all/default/host/net/poisoned/static/subnet *ip-interface-address*

Prevents the router from advertising the specified type of route in RIP updates sent out using 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`

sending rip-routes-only *ip-interface-address*

Stops advertising only RIP routes in the RIP2 multicast packets.

ip-interface-address

Valid Values: any valid IP address of an interface that has RIP2 enabled.

Default Value: none

Example: `disable sending rip1-routes-only 128.185.123.22`

sending outage-only *interface-IP-address*

Disables the sending of RIP updates contingent on the presence of the route specified on the analogous enable command. When this function is disabled, RIP advertisements will be sent unconditionally.

interface-IP-address

Valid Values: any valid IP address

Default Value: none

Example: `disable sending outage-only`

simple-internet-access

This option disables Simple Internet access on the specified interface.

source-addr-verification

This inbound packet filter option verifies that a received packet's source IP address is consistent, based on the IP routing table, with the interface from which it was received. This option helps prevent the forwarding of packets from an IP host that is using a source IP address that does not belong to it, a technique known as *spoofing*. This command is valid only in the packet filter configuration console (accessed by the **update packet-filter** command).

source-routing

Prevents the router from forwarding source-routed packets (that is, IP packets that include a source-route option). This option defaults to source-routing enabled.

Example: `disable source-routing`

tftp-server

Prevents the router from accepting TFTP GET or PUT requests from the network. This prevents the inadvertent overlaying of configuration files or load images from another device. You will still be able to perform TFTP client operations (GETs and PUTs) from the router through a directly attached terminal or Telnet session.

timestamp

Disables the router from receiving or forwarding IP packets that contain a timestamp IP option. By default, the router receives and forwards these packets.

trace *rule-number*

Disables tracing of IP packets that match the specified access control rule number. By default, this option is disabled. For more information on tracing IP packets, see the **enable trace** command.

Example: `disable trace 1`

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 0 to 65535

Default Value: 0

Example: `disable udp-forwarding 36`

vrrp Disables Virtual Router Redundancy Protocol.

Example: `disable vrrp`

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
classless
compression-bypass
directed-broadcast
dynamic-address
echo-reply
encryption-bypass
fragment-offset-check
icmp-redirect
nexthop-awareness
override default ...
override static-routes ...
packet-filter
per-packet-multipath
receiving policy . . .
receiving rip ...
receiving dynamic all ...
receiving dynamic hosts...
receiving dynamic nets ...
receiving dynamic subnets ...
record-route
rip
rip2
route-table-filtering
same-subnet
sending all-routes ...
sending default-routes ...
sending host-routes ...
sending net-routes ...

sending outage-only . . .

sending poisoned-reverse-routes

sending policy . . .

sending rip1-routes-only

sending static-routes ...

sending subnet-routes ...

simple-internet-access

source-addr-verification

source-routing

tftp-server

timestamp

trace

udp-forwarding ...

vrrp ...

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 hosts unaware of subnetting 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 or paths as a standby.

Default Value: 0

classless
Indicates the router will be operating in a classless IP addressing environment. The IBM 2210 fully supports CIDR addressing as described in RFC 1817 without this option enabled. Enabling this option prevents automatic generation of the classful network routes (for example, Class A, B, or C network routes) corresponding to routes added to the IP route table. If you are not running RIPv1 you do not require the classful network route.

Example: `enable classless`

compression-bypass *rule-number*
Enables bypassing of Layer 2 compression for the access control specified by *rule-number*, causing packets to bypass compression, even if compression is enabled on the network interface. Compression-bypass can be enabled only for access controls of type **include**.

Example: `enable compression-bypass 1`

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`

dynamic-address *0.0.0.n*
On the specified PPP network interface, enables the router to learn its IP address from the remote node on that network interface. By default, this option is disabled.

This option is valid only on a network interface that has been configured as an unnumbered serial line interface (the IP address assigned to the network interface by the **add address** command is *0.0.0.n*, where *n* is the network interface number).

Note: For the router to learn its IP address from the remote node, in addition to enabling this option, you must also configure IPCP on the PPP network interface to request an IP address from the remote node.

Example:

```
Config>network 1
Point-to-Point user configuration
PPP 1 Config>set ipcp
IP COMPRESSION [no]:
Request an IP address [no]: yes
Interface remote IP address to offer if requested (0.0.0.0 for none) [0.0.0.0]?
PPP 1 Config>exit
Config>protocol ip
Internet protocol user configuration
IP config>add address
Which net is this address for? [0]? 1
New address []? 0.0.0.1
Address mask [0.0.0.0]?
IP config>enable dynamic-address
Interface address []? 0.0.0.1
IP config>
```

echo-reply

Enables the building and sending of an ICMP Echo Reply in response to an ICMP Echo Request.

Example: `enable echo-reply`

encryption-bypass *rule-number*

Enables bypassing of Layer 2 compression for the access control specified by *rule-number*, causing packets to bypass encryption, even if encryption is enabled on the network interface. Encryption-bypass can be enabled only for access controls of type **include**.

Example: `enable encryption-bypass 1`

fragment-offset-check

Enables the checking of the fragment offset of received IP packets in which the IP protocol number is 6 (that is, TCP). Packets with a fragment offset of 1 are dropped. By default, this check is disabled.

Note: After it has been enabled, this function can be activated without affecting any other functions of IP. See the talk 5 **reset IP** command for more information.

icmp-redirect *ip-interface-address*

Enables the router to send ICMP Redirect messages on the specified IP interface. If you enter nothing at the prompt for the IP interface address, the device will be enabled to send ICMP Redirect messages on all IP interfaces.

ip-interface-address

Valid Values: any valid IP address, or nothing for all IP interfaces

Default Value: none

Example:

```
IP config> enable icmp-redirect
Interface address (NULL for all) []? 192.9.200.44
IP config>
```

nexthop-awareness *ip-interface-address*

Enables nexthop awareness on an IP interface.

ip-interface-address

Valid Values: any valid IP address

Default Value: disabled

Example:

```
IP config>enable nexthop-awareness 1.1.1.1
IP config>enable nexthop-awareness
Interface address []? 2.2.2.2
IP config>
```

override default *ip-interface-address*

Enables received RIP information to override any default route installed in the IP routing table. 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 overwrites the router's current default route, 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 load balances between equal cost paths for each packet in round-robin fashion. The default for this feature is disabled.

Example: `enable per-packet-multipath`

receiving policy global/interface *ip-interface-address route-policy-identifier*

Enables the use of the route filter policy for determination of what routes are accepted by RIP. The **enable receiving policy global** *route-policy-identifier* command enables the use of the global route filter receiving policy for RIP interfaces that meet these two conditions:

- They do not have an interface-level RIP receiving route filter policy configured. If an interface-level RIP receiving route filter policy is configured, that policy continues to determine which routes are accepted.
- Reception of RIP routes is not disabled.

After the global route filter receiving policy has been enabled, the RIP interfaces that meet these conditions will accept routes as defined by the policy.

The command **enable receiving policy interface** *ip-interface-address route-policy-identifier* enables the use of the route filter policy for the determination of which routes are accepted on one specified RIP interface. Note that dynamic nets, subnets, and hosts are not applicable if global or interface receiving policy is enabled.

ip-interface-address
Valid Values: any valid IP address

Default Value: none

route-policy-identifier
Valid Values: a valid route policy identifier consisting of a 1 to 15-character string of ASCII characters

Default Value: none

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.

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.

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.

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`

record-route

Enables the router to receive and forward IP packets that contain a record route IP option. This is the default.

Note: After it has been enabled, this function can be activated without affecting any other functions of IP. See the talk 5 **reset IP** command for more information.

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`

rip2 *ip-interface-address RIP2-authentication authentication-keys*

Enables RIP2 on an IP interface. RIP2 advertisements are sent to the 224.0.0.9 multicast address. RIP2 is described in RFC 1723.

ip-interface-address

Indicates the IP interface on which RIP2 is enabled.**Valid Values**: any valid IP address

Default Value: none

RIP2-authentication

Indicates whether or not a simple clear-text key will be used for RIP2 authentication. Authentication is not required.**Valid Values**: yes or no

Default Value: yes

authentication-key

Defines a clear-text password which will be used for RIP2 authentication. You are prompted for this string only when you answer **yes** to the question "Set RIP-2 Authentication?" When RIP2 authentication is used, only RIP2 packets with a matching password are accepted.**Valid Values**: a clear-text ASCII string

Default Value: a null string

Example:

```
IP config>enable rip2
Set for which interface address [0.0.0.0]? 153.2.2.25
RIP2 is enabled on this interface.
Set RIP-2 Authentication? [Yes]: yes
Authentication Key []? C1C3C5C5
Retype Auth. Key []? C1C3C5C5
RIP2 Authentication is enabled on this interface.
```

route-table-filtering

Applies route table filters to any route added to the routing table. Route table filters are applied based on a most-specific match of the destination and network mask. Route table filters are never applied to direct routes or static routes.

Example: `enable route-table-filtering`

same-subnet

Enables the same subnet option. When the device is rebooted, it will allow multiple IP interfaces to the same subnet to be installed. Multiple IP interfaces to the same subnet are useful under only one of the following conditions:

- OSPF Point-to-Multipoint is configured on the IP interfaces.
- Nexthop Awareness is enabled on the IP interfaces, and static routes are defined for the routes that go through the IP interfaces.

By default, this option is disabled.

Example: `enable same-subnet`

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: By default, RIP will send network, subnet, and static routes.

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 outage-only *interface-ip-address outage-network outage-network-mask*
Enables the sending of RIP update packets on the interface specified by *interface-ip-address* contingent on the presence of the IP route specified by the *outage-network* and *outage-network-mask*. Normally, updates are sent unconditionally on interfaces configured to advertise RIP routes. Additionally, RIP updates are ignored on an outage-only interface when the specified route is present. This function can be useful in backup scenarios where the backup dial circuit is configured as a Dial-on-Demand circuit.

ip-interface-address
Valid values: any valid IP address

Default value: none

outage-network
Valid values: any valid IP address

Default value: none

outage-network-mask
Valid values: any valid IP mask

Default value: none

Example: `enable sending outage-only`

```
IP config>enable sending outage-only
Set for which interface address [0.0.0.0]? 0.0.0.2
Outage network []? 10.50.0.0
Outage network mask []? 255.255.0.0
```

In this example, RIP advertisements will only be sent on the unnumbered interface when the 10.50.0.0/255.255.0.0 route is absent from the routing table.

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 policy global/interface *ip-interface-address route-policy-identifier*
Enables the use of the route filter policy for determination of what routes are advertised by RIP. The **enable sending policy global** *route-policy-identifier* command enables the use of the global route filter sending policy for RIP interfaces that meet these two conditions:

- They do not have an interface-level RIP sending policy configured. If an interface-level RIP sending policy is configured, it continues to determine which routes are advertised.
- Sending of RIP routes is not disabled.

After the global route filter sending policy is enabled, RIP interfaces that meet these two conditions will advertise routes as determined by the global route filter sending policy.

The **enable sending policy interface** *ip-interface-address route-policy-identifier* command enables the use of the route filter policy for the determination of which routes are advertised on the specified RIP interface. The route filter policy can be used to filter aggregate routes. See "Route Aggregation" on page 230 for more information.

ip-interface-address
Valid Values: any valid IP address

Default Value: none

route-policy-identifier
Valid Values: a valid route policy identifier consisting of a 1 to 15-character string of ASCII characters

Default Value: none

sending rip-routes-only *ip-interface-address*
To advertise only RIP routes in the RIP2 multicast packets.

ip-interface-address
Valid Values: any valid IP address of an interface that has RIP2 enabled.

Default Value: none

Example: `enable sending rip-routes-only 128.185.123.22`

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 that meet other sending criteria 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.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

simple-internet-access

This option enables Simple Internet Access on the specified interface. Use Simple Internet Access to create an integrated configuration that includes an unnumbered serial line with the Dynamic Address option enabled, IP Packet Filters (both inbound and outbound), IP Access Controls (both inbound and outbound), a Default Route, Dynamic NAT/NAPT Reserve pool and a private address subnet in the DHCP feature. Users that have additional requirements for the listed integrated configuration should consider manual configuration.

Note: This option is only available on images that include both the DHCP and NAT feature.

source-addr-verification

This inbound packet filter option verifies that a received packet's source IP address is consistent, based on the IP routing table, with the interface from which it was received. This option helps prevent the forwarding of packets from an IP host that is using a source IP address that does not belong to it, a technique known as *spoofing*. This command is valid only in the packet filter configuration console (accessed by the **update packet-filter** command).

source-routing

Allows the router to forward IP packets containing an IP source route option.

Example: `enable source-routing`

tftp-server

Allows the router to accept TFTP GET or PUT requests from the network for configuration files or image loads.

Example: `enable tftp-server`

timestamp

Enables the router to receive and forward IP packets that contain a Timestamp IP option. This is the default.

Note: After it has been enabled, this function can be activated without affecting any other functions of IP. See the talk 5 **reset IP** command for more information.

trace *rule-number*

Enables tracing of IP packets that match the specified access control rule number. By default, this option is disabled.

Tracing of IP packets uses the packet trace function of the Event Logging System. See the **set trace** and **view** commands in the chapter Configuring and Monitoring the Event Logging System (ELS) in the *Software User's Guide* for details on these commands. Only IP packets that match an access control rule for which trace is enabled are traced. The access control rule may be in the global access control list or in a packet filter access control list. Use the **list access-control** command to see the rules for which trace has been enabled.

Example: Tracing all IP Packets

```
IP config>set access-control on
IP config>add access-control
Access Control type [E]? i
Internet source [0.0.0.0]?
Source mask [0.0.0.0]?
Internet destination [0.0.0.0]?
Destination mask [0.0.0.0]?
Starting protocol number ([0] for all protocols) [0]?
Starting DESTINATION port number ([0] for all ports) [0]?
Starting SOURCE port number ([0] for all ports) [0]?
Filter on ICMP Type ([-1] for all types) [-1]?
TOS/Precedence filter mask (00-FF - [0] for none) [0]?
TOS/Precedence modification mask (00-FF - [0] for none) [0]?
Use policy-based routing? [No]:
Enable logging? [No]:
IP config>list access-control
Access Control is: enabled
Access Control facility: USER

List of access control records:


   1   Type=I   Source=0.0.0.0    Dest   =0.0.0.0     Prot=17
                SMask =0.0.0.0    DMAask =0.0.0.0
                SPorts=5004-5511  DPorts=5004-5511
                       T/C=**/**  Log=N
                BypassComp BypassEncIP config>

enable trace
Index of access control to be traced [1]?
IP config> Ctrl-P
*talk 5
+protocol ip
IP>reset ip
IP>exit
+event
Event Logging System user console
ELS>set trace memory-trace-buffer-size
Amount of memory (in bytes) reserved for tracing [0]? 10000
ELS>set trace on
ELS>set trace decode on
ELS>view first
#1 Dir:INCOMING  Time:0.5.47.53  Trap:450
Comp:IPV4  Type:UNKNOWN  Port:65535  Circuit:0x000000  Size:64

** IPv4 Packet **
Ver/Hdr Len/TOS:           4 20 0x00
```

```
Packet Length/ID:          64 0x9E4E
Fragment Offset:           0x0000
TTL/Protocol/Hdr Chksum:   1 OSPF  0xA89D
Source Addr/Dest Addr:     10.0.10.106 224.0.0.5

** OSPF Header **
Version:                   2
Packet type:               Hello
Packet length:             44
Router ID:                 10.0.0.106
Area ID:                   0.0.0.0
Checksum:                  0xDDB5
Authentication type:       0
Authentication:            0x00000000
Authentication:            0x00000000
Network mask:              255.255.255.0
Hello interval:            10
Options:                   E-bit
Options:                   MC-bit
Router priority:           1
Router dead interval:      40
Designated router:         10.0.10.106
Backup Designated router:  0.0.0.0
ELS>
```

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 0 to 65535

Default Value: 0

Example: `enable udp-forwarding 36`

vrrp Enables Virtual Router Redundancy Protocol

Example: `enable vrrp`

List

Use the **list** command to display various pieces of the IP configuration data, depending on the particular subcommand invoked.

Syntax:

list

all

access-control

addresses

aggregate

bootp

filters

icmp-redirect

igmp

mtu

nexthop-awareness

packet-filter

parameters

protocols

redundant default gateway

rip

route-policy

route-table-filtering

routes

simple-internet-access

sizes

tags

udp-forwarding

vrid

all Displays the entire IP configuration.

Example: `list all`

access-control
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`

```
list access-control
1   Type=I   Source=0.0.0.0    Dest   =0.0.0.0     Prot=17
             SMask =0.0.0.0    DMAask =0.0.0.0
             SPorts=5004-5511  DPorts=5004-5511
                    T/C=**/**  Log=N
             BypassComp BypassEnc
```

addresses
Displays the IP interface addresses that have been assigned to the router, along with their configured broadcast formats. The interface identified by *BDG/0* is the bridging interface.

Example: `list addresses`

aggregate
Displays all configured aggregate routes.

Example:

```
IP config>list aggregate

                Aggregate Route Configuration

Address         Mask             Type           Policy/Metric
-----------------------------------------------------------------
10.1.0.0        255.255.0.0      Unconditional         12
10.2.0.0        255.255.0.0      Policy-Driven  EAST-REGION
```

bootp Indicates whether BOOTP forwarding is enabled or disabled as well as the configured list of BOOTP servers.

Example: `list bootp`

filters Lists the router's configured filtered networks.

icmp-redirect
Lists whether the sending of ICMP redirect messages is enabled or disabled on each IP interface.

igmp Displays the IGMP configuration.

Example:

```
IP config>list igmp

Net          IGMP          Query          Response          Leave Query
            Version       Interval        Interval           Interval
                            (secs)          (secs)             (secs)
---         -------       --------        --------          -----------
   0            2             250             10                  1
   1            1             125             10                  1
   4            2             125             10                  2
   5            2             125             20                  1

IP config>
```

mtu Lists configured MTU values.

nexthop-awareness
Lists the setting of nexthop awareness on all IP interfaces.

Example:

```
IP config>list nexthop-awareness
Nexthop awareness for each IP interface address:
   intf  0   1.1.1.1          255.0.0.0        nexthop awareness enabled
   intf  1   2.2.2.2          255.0.0.0        nexthop awareness disabled
IP config>
```

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. If you have configured a packet filter on the bridge interface, the interface is identified by *BDG/0*.

Example: `list packet-filter pf-in-0`

```
Name               Direction     Interface
pf-in-0            In            0

Access Control is: enabled

List of access control records:


   1   Type=I   Source=0.0.0.0    Dest   =0.0.0.0     Prot=17
                SMask =0.0.0.0    DMAask =0.0.0.0
                SPorts=5004-5511  DPorts=5004-5511
                       T/C=**/**  Log=N
                BypassComp BypassEnc

2  Type=IN   Source=10.1.1.1      Dest=10.1.1.2          Prot=0-255
             Mask=255.255.255.255 Mask=255.255.255.254
             Sports=  N/A         Dports=  N/A
                                  Log=Yes ELS=N SNMP=Y SLOG=L(Emergency)

3  Type=I    Source=0.0.0.0       Dest=0.0.0.0          Prot=0-255
             Mask=0.0.0.0         Mask=0.0.0.0
             Sports= 1-65535      Dports=  1-68835
                                  Log=No
             Trace=Enabled
```

parameters
Lists the various global IP parameters.

Example: `list parameters`

```
IP config>list parameters
ARP-SUBNET-ROUTING    : enabled
ARP-NET-ROUTING       : enabled
CLASSLESS             : disabled
DIRECTED-BROADCAST    : enabled
DSCACHE-SIZE          : 1024 entries
ECHO-REPLY            : enabled
FRAGMENT-OFFSET-CHECK : enabled
PER-PACKET-MULTIPATH  : disabled
REASSEMBLY-SIZE       : 12000 bytes
RECORD-ROUTE          : enabled
```

```
ROUTING TABLE-SIZE    : 768 entries (52224 bytes)
(Routing) CACHE-SIZE  : 64 entries
SAME-SUBNET           : disabled
SOURCE-ROUTING        : enabled
TIMESTAMP             : enabled
TTL                   : 64
```

protocols

Displays the configured state of the IP routing protocols (OSPF, RIP, BGP) along with other general configuration settings.

Example: `list protocols`

redundant default gateway

Displays the Redundant Default IP Gateway for each interface configured.

Example: `list redundant`

```
Redundant Default IP Gateways for each interface:
   inf 4  11.1.1.6     255.0.0.0     00.00.00.00.00.BA  primary
   inf 8  33.3.3.6     255.0.0.0     00.00.00.00.00.AB  backup
```

rip Displays all RIP configuration parameters. RIP can be configured to receive and send dynamic routes or the routes can be defined by a route filter policy. See the IP configuration commands **enable receiving dynamic nets/subnets/hosts** for more information about dynamic routing. See “Route Filter Policy Configuration” on page 298 for more information about route filter policies.

Example:

```
IP config>list rip

RIP: enabled
RIP default origination: disabled
RIP global receive policy: rip-in

  Per-interface address flags:
  Net:     0  153.2.2.25       RIP Version 1
                               Send net, subnet and static routes
                               Receive routes based on global receive
                                    policy: rip-in
                               RIP interface input metric: 1
                               RIP interface output metric: 0
  Net:     1  153.2.1.1        RIP Version 1
                               Send net, subnet and static routes
                               Receive routes based on global receive
                                    policy: rip-in
                               RIP interface input metric: 1
                               RIP interface output metric: 0
  Net:     2  0.0.0.2          RIP Version 1
                               Send routes based on interface send
                                    policy: rip-import
                               Receive routes based on global receive
                                    policy: rip-in
                               RIP interface input metric: 1
                               RIP interface output metric: 0

Accept RIP updates always for:
[NONE]
```

route-policy *route-policy-identifier*

Displays configured route policy information. If you specify a particular route policy, a detailed listing of that route policy is displayed. If you do not specify a particular route policy, a summary display of all policies is displayed.

Example:

```
IP config>list route-policy
Route Policy Identifier [1-15 characters] [ ]?

 Route Policy        Checksum  Policy-Application
 rip-send            0x8637      Longest-match
 rip-receive         0x5049      Longest-match
 rip-global-send     0xC9EA      Longest-match
```

route-table-filtering

Displays the list of route filters added to the routing filter.

Example: `list route-table-filtering`

```
IP config>list route-table-filtering

Route Filtering Disabled

Destination     Mask           Match Type
10.1.1.0        255.255.255.0   BOTH  E
50.50.0.0       255.255.0.0     BOTH  I
10.1.1.1        255.255.255.255 EXACT I
50.0.0.0        255.0.0.0       BOTH  E

MORE-Match more-specific routes   EXACT-Match route exactly
BOTH-Match exact and more-specific routes  E-Exclude  I-Include
IP config>
```

routes

Displays the list of static routes that have been configured.

Example: `list routes`

```
IP config>list routes

route to 1.1.0.0        ,255.255.0.0     via 10.1.1.1        cost 1
                                         via 20.1.1.1        cost 2
                                         via 30.1.1.1        cost 3
route to 2.2.0.0        ,255.255.0.0     via 10.2.2.2        cost 10
route to 3.3.0.0        ,255.255.0.0     via 10.3.3.3        cost 100
                                         via 20.3.3.3        cost 200
```

simple-internet-access

Displays the Simple Internet Access interface number.

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`

vrid Displays the configured VRRP status, VRIDs, and VRID addresses. In this example, the *preempt-mode* parameter and the *hardware MAC address* option are both **yes** as shown by the Flags field that displays P and H.

Example:

```
IP config>list vrid

VRRP Enabled

                          --VRID Definitions--

IP address      VRID  Priority Interval Auth   Auth-key Flags   Address(es)
153.2.2.25         1       255        1 None   N/A      P,H
```

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...
- access-control log-facility
- broadcast-address...
- cache-size
- default network-gateway...
- default subnet-gateway...
- dscache-size
- igmp ...
- internal-ip-address
- mtu

originate-rip-default

reassembly-size

rip-in-metric

rip-out-metric

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`

access-control log-facility *log-facility*

Sets the SysLog facility for access control. The SysLog facility option defines the system upon which the SysLog messages will be displayed.

Note: After it has been enabled, this function can be activated without affecting any other functions of IP. See the talk 5 **reset IP** command for more information.

log-facility

Valid Values: KERN, USER, MAIL, DAEMON, AUTH, SYSLOG, LPR, NEWS, UUCP, CRON, AUTHPRIV, LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7, USER

Default Value: USER

Example:

```
IP config> set access-control log-facility
SYSLOG facility? (KERN, USER, MAIL, DAEMON, AUTH, SYSLOG, LPR,
NEWS, UUCP, CRON, AUTHPRIV, LOCAL0, LOCAL1, LOCAL2,
LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7) [USER]?
```

broadcast-address *ip-interface-address style fill-pattern*

Specifies the IP broadcast format that the router uses when broadcasting packets out on 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).

nexthop

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 0 to 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 0 to 255

Default Value: 1

Example: `set default subnet-gateway 128.185.0.0 128.185.123.22 6`

dscache-size *entries*

Configures the number of entries to be allocated for the DiffServ flow cache. The DiffServ flow cache is allocated if one of the following conditions is true:

- The Differentiated Services (DiffServ, or DS) feature is enabled (see Configuring and Monitoring the Differentiated Services Feature in *Using and Configuring Features* for details).
- The IP Security (IPSec) feature is enabled (see Configuring and Monitoring IP Security in *Using and Configuring Features* for details).
- The Policy feature is enabled (see Configuring and Monitoring the Policy Feature in *Using and Configuring Features* for details).

Valid Values: 64 to 8192

Default Value: 1024

igmp ...

Configures Internet Group Management (IGMP) parameters.You can specify values for the following parameters:

query interval *net interval*

Changes the interval between IGMP general queries.

net Specifies the network number for the interface being configured.

Valid values: Any valid network number

Default value: None

interval

Specifies the number of seconds between the transmissions of general queries.

Valid values: 1 - 3600

Default value: 125

response-interval *net interval*

Changes the maximum response time inserted into IGMP general queries.

net Specifies the network number for the interface being configured.

Valid values: Any valid network number

Default value: None

interval
Specifies the number of seconds between the transmissions of a query and a host sending an IGMP Report in response.

Valid values: 1 - 60

Default value: 10

robustness-variable *net variable*
Changes the robustness variable for a network.

net Specifies the network number for the interface being configured.

Valid values: Any valid network number

Default value: None

variable
Specifies the number of IGMP packets sent to combat packet loss on a network.

Valid values: 2 - 10

Default value: 2

leave-interval *net interval*
Changes the maximum response time inserted into IGMP specific queries.

net Specifies the network number for the interface being configured.

Valid values: Any valid network number

Default value: None

interval
Specifies the number of seconds allowed between the transmissions of specific queries and a host sending an IGMP Report in response.

Valid values: 1 - 60

Default value: 1

version *net vernum*
Changes the version of IGMP running on a network.

net Specifies the network number for the interface being configured.

Valid values: Any valid network number

Default value: None

vernum
Specifies the version number to run on the network.

Valid values: 1 or 2

Default value: 2

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. It will also show up as a host route and will be advertised in RIP if allowed by the RIP sending configuration of the interface.

Valid Values: any valid IP address.

Default Value: none

Example: `set internal-ip-address 142.82.10.1`

mtu Sets the MTU value for the IP protocol on this interface.

Valid Values: 0, 68 - 65535

Default Value: Minimum of all non-zero MTUs on the network

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 to indicate whether the router should always originate a RIP default or to originate a RIP default only when the route from other protocols are available.

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 0 to 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 0 to 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]?
```

- Answering "Yes" to the "Always originate" question 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.
- When 0 is specified for the BGP route AS (Autonomous System) number, a route meeting the network criteria from any AS will cause a RIP default to be originated.
- When 0.0.0.0 is specified for the BGP network criteria, any BGP route meeting AS criteria will cause a RIP default to be originated.

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`

rip-in-metric *ip-interface-address metric*
Allows the configuration of the metric to be added to RIP routes of an interface prior to installation in the routing table.

ip-interface-address
Valid Values: any valid IP address

Default Value: none

metric **Valid Values**: an integer in the range 1 to 15

Default Value: 1

Example: `set rip-in-metric 128.185.120.209 1`

rip-out-metric *ip-interface-address metric*
Allows the configuration of the metric to be added to RIP routes advertised on an interface configured to advertise RIP or RIP2 routes.

ip-interface-address
Valid Values: any valid IP address

Default Value: none

metric **Valid Values**: an integer in the range 0 to 15

Default Value: 0

Example: `set rip-out-metric 128.185.120.209 0`

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 317 for additional information about table sizes.

Valid Values: an integer number of entries in the range 64 to 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 information on originate, send, and receive policies in the chapter "Using and Configuring BGP" in *Protocol Configuration and Monitoring Reference Volume 1*.) Tags are propagated also by the OSPF routing protocol.

Valid Values: an integer in the range 0 to 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 1 to 255

Default Value: 64

Example: `set ttl 255`

Update

Use the **update packet-filter** command to configure packet filters. This is an example of the command:

```
IP config> update packet-filter
Packet-filter name [ ]? pf-1-in
Packet-filter 'pf-1-in' Config>
```

Packet-filter-name is any packet filter name that you have created by using the **add packet-filter** *packet-filter-name* command from the `IP config>` prompt. To enable the packet filter, you use the **set access-control on** command. From the `Packet-filter '`*`packet-filter-name`*`' Config>` prompt, you can enter these commands:

Syntax:

add access-control

change access-control

delete access-control

disable

enable

list access-control

move access-control

For the **add access-control**, **change access-control**, **delete access-control**,**list access-control**, and **move access-control** commands for the `Packet-filter '`*`filter-name`*`' Config>` prompt, see the descriptions of the parameters under the **access-control** parameter that is displayed at the `IP config>` prompt. For example, see **add access-control** for a description of the parameters for the **update packet-filter add access-control** command.

For the **disable** and **enable** commands, the keyword **source-addr-verification** can be configured only from the `Packet-filter '`*`filter-name`*`' Config>` prompt.

The following sections list the parameters that are unique to the **update packet-filter** command. These are parameters that apply to packet filters, but not to router-wide filters and are entered only at the `Packet-filter '`*`filter-name`*`' Config>` prompt.

add/change access-control *type*

Network Address Translation (NAT)

This type of packet filter access control rule passes packets to NAT for address translation. This type is valid only in packet filters and only when specified in combination with inclusive, for example, ***IN***. Refer to the description of the NAT feature in the *Software User's Guide* for more information. An example of access control rules for NAT is found in the in the chapter Using Network Address Translation in *Using and Configuring Features*.

Valid Value: N

Default Value: none

disable/enable source-addr-verification

This inbound packet filter option verifies that a received packet's source IP address is consistent, based on the IP routing table, with the interface from which it was received. This option helps prevent the forwarding of packets from an IP host that is using a source IP address that does not belong to it, a technique known as *spoofing*.

Example:

```
Packet-filter 'filter-name' Config> enable source-addr-verification
```

disable/enable trace *access-control-index-number*

This option disables or enables packet trace for a specific access control rule. To view the packets that have been traced, use the Talk 5 GWCON **event** command and use the Talk 5 ELS **view** command with the appropriate options to display the trace records.

Note: To make the trace setting active, enter the Talk 5 **reset IP** command.

Example:

```
Packet-filter 'filter-name' Config> enable trace 1
```

Examples:

The following examples show how to configure various access control rules for packet filters. Refer to the chapter "Using Network Address Translation" in Using and Configuring Features for an example of access control rules for NAT.

Example 1—Exclusive type access control rule

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]?
Enable Logging? (Yes or [No]):
```

Example 2—Deleting an access control rule

Use the **list access-control** command to find the access control index number.

```
    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 that you have specified.

```
4  Type=I    Source=1.2.9.9       Dest=0.0.0.0          Prot=0-255
             Mask=255.0.0.255     Mask=0.0.0.0
             Sports=  0-65535     Dports=  1-65535
                                  Log=No
Are you sure this is the record you want to delete (Yes or [No]): y
    Deleted
    Packet-filter 'test' Config>
```

Dports are destination ports and *Sports* are source ports.

Example 3— List access-control command

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
Access Control facility: USER

List of access control records:

1  Type=E    Source=128.185.0.0   Dest=0.0.0.0           Prot=0-255
```

```
                Mask=255.255.0.0     Mask=0.0.0.0
                Sports=  0-65535     Dports=  1-65535
                ACK0=N  T/C= **/**   Log=No
                Trace=Enabled

2  Type=I       Source=9.67.8.3      Dest=128.54.67.8     Prot=0-255
                Mask=255.255.255.255 Mask=255.255.255.254
                Sports=  N/A         Dports=  N/A
                                     Log=Yes ELS=N SNMP=Y SLOG=L(Emergency)

3  Type=I       Source=0.0.0.0       Dest=0.0.0.0           Prot=0-255
                Mask=0.0.0.0         Mask=0.0.0.0
                Sports= 1-65535      Dports=  1-68835
                                     Log=No
```

Route Filter Policy Configuration

This section describes the subset of commands used to configure route filter policies. To access this subset of IP configuration commands, follow these steps:

1. Create a route filter policy. See the **add route-policy** command on page 253.
2. Use the **change route-policy** command to bring up the `IP Route Policy Config>` prompt. The `IP Route Policy Config>` prompt applies only to the particular route policy identified by the **change route-policy** command.

   Example:

   ```
   IP config>change route-policy ospf-import
   ospf-import IP Route Policy Configuration
   IP Route Policy Config>
   ```

Note: Route filter policies can be used to determine which routes are imported in OSPF and the specific details of their advertisement, including OSPF external type, metric, and tag value. Refer to the **enable as boundary routing** command on page 351 for information about using route filter policies to configure OSPF.

Route filter policies can also be used to control what routes are advertised or accepted when RIP is used. See the previously described **enable receiving**, **enable sending**, **disable receiving**, and **disable sending** commands.

Table 19. IP Route Policy Configuration Commands Summary

Command	Function
Add	Adds an action, an entry, or a match condition to a route filter policy.
Delete	Deletes an action, an entry, or a match condition from a route filter policy.
List	Lists the route policy entries, actions, and match conditions for the route policy currently being changed.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Add

Use the **add** command to add route filter policy entries to the route filter policy, to add match conditions to existing entries, or to add actions to existing entries.

Syntax:

add action . . .
entry . . .
match-condition . . .

action . . .

Adds an action to an existing route filter policy entry. Adding an action to a route filter policy is optional. One action can be added to each entry. If you

need more than one action to apply to one address or address range, specify a second entry for that address or range. Then, define the second action for the second entry. These are the actions that can be specified:

Syntax:

auto-tag

set manual-tag

set metric

set route-type

auto-tag *route-policy-index*
Automatically sets the tag for the route, using a routing protocol specific heuristic. This option is described in RFC 1745.

route-policy-index
Identifies the entry to which the action should be applied.

Valid Value: 1 to 65535

Default Value: none

set manual-tag *route-policy-index manual-tag*
Sets the manual tag for the route to the specified value. This tag is usually the AS number when the protocol is OSPF.

route-policy-index
Identifies the entry to which the action should be applied.

Valid Value: 1 to 65535

Default Value: none

manual-tag

Valid Value: X'0' to X'FFFFFFFF'

Default Value: none

set metric *route-policy-index metric*
Sets the metric for the route to the specified value.

route-policy-index
Identifies the entry to which the action should be applied.

Valid Value: 1 to 65535

Default Value: none

metric

Valid Value: 1 to 255

Default Value: none

set route-type *route-policy-index route-type*
Sets the OSPF external route type. This action is ignored for applications other than OSPF AS boundary route importation.

route-policy-index
Identifies the entry to which the action should be applied.

Valid Value: 1 to 65535

Default Value: none

route-type

Valid Value: 1 or 2

Default Value: none

entry *route-policy-index ip-address ip-mask address-match policy-type*

Adds a route filter policy entry to the route filter policy being changed. Each entry within a route filter policy is identified by a unique index number, which is manually configured. If the entry with the specified index number already exists, that entry is changed according to the new parameters configured.

When you add the route filter policy, you define the processing of the entries as either strictly linear or longest match. If the route filter policy processing is strictly linear, the route filter policy entries are processed according to the ascending order of their index numbers. If the route filter policy processing is longest match, the route filter policy entries are processed according to the IP address and mask that has the longest match. If multiple route filter policy entries have the same IP address and mask when longest match is used, then the match will be in order of ascending index number among the entries with the same IP address and mask.

route-policy-index

Identifies the entry.

Valid Value: 1 to 65535

Default Value: none

ip-address

Valid Value: any valid IP address

Default Value: none

ip-mask

Valid Value: any valid IP mask

Default Value: none

address-match exact/range

If this value is *exact*, the route filter policy entry will match only on a route with that exact address and mask. If this value is *range*, the route filter policy entry will match on any route that is within the range encompassed by the address and mask, including the exact route.

Valid Value: exact or range

Default Value: range

policy-type inclusive/exclusive

If this value is *inclusive*, routes matching this route filter policy entry are included in the routing table. If this value is *exclusive*, routes matching this route filter policy entry are excluded, that is, they are not entered into the routing table. Even if actions are configured for a route filter policy entry that is exclusive, these actions are not applicable.

Valid Value: inclusive or exclusive

Default Value: inclusive

match-condition . . .

Adds a match condition to an existing route filter policy entry. A match condition, which is an optional parameter or set of parameters, is applied to a route that the entry definition has matched. The match condition filters the packet for particular conditions in addition to the IP address and the IP mask. Only one match condition can be configured per entry. To use two match conditions for the same address or address range, you can add a second entry to the route filter policy and specify the second match condition for that entry. These are the match conditions:

Syntax:

as

gateway

metric

net

protocol

source-gateway

as *route-policy-index as-number*

Matches the route according to its AS number. This value is interpreted only when the route filter policy is applied to AS boundary routing.

route-policy-index

An integer that identifies the entry with which the match should be made.

Valid Value: 1 to 65535

Default Value: none

as-number

Valid Value: 1 to 65535

Default Value: none

gateway *route-policy-index gateway-address-and-mask*

Matches the route with a next-hop gateway in the specified range.

route-policy-index

Identifies the entry with which the match should be made.

Valid Value: 1 to 65535

Default Value: none

gateway-address-and-mask

Valid Value: a valid IP address and mask

Default Value: none

metric *route-policy-index lower-metric-number upper-metric-number*

Matches the metric of the route with the one of the numbers in a range of metric numbers. You will be prompted for two numbers to identify the range of metric numbers: one for the low end of the range and one for the high end. If you want a single metric number, specify the same number twice.

route-policy-index

Valid Value: 1 to 65535

Default Value: none

lower-metric-number

Valid Value: 1 to 65535

Default Value: none

upper-metric-number

Valid Value: 1 to 65535

Default Value: none

net *route-policy-index lower-net-number upper-net-number*

Matches the routes that have a next hop with an outgoing network number in the range defined by the lower and upper network numbers. You will be prompted for two numbers to identify the range of outgoing network numbers: one for the low end of the range and one for the high end. If you want a single network number, specify the same number twice.

route-policy-index

Identifies the entry with which the match should be made.

Valid Value: 1 to 65535

Default Value: none

lower-net-number

The lower bound of the network number range for matching next-hop outgoing networks. These can be viewed using the **list devices** command from the `Config>` prompt.

Valid Value: 1 to 65536

Default Value: none

upper-net-number

The upper bound of the network number range for matching next-hop outgoing networks.

Valid Value: 1 to 65536

Default Value: none

protocol *protocol route-policy-index*

Matches the route with a protocol.

protocol

Valid Values:

Syntax:

aggregate

bgp

direct

natural-nets

ospf-intra

ospf-inter

ospf

ospf-all

ospf-ext

ospf-e1

ospf-e2

rip

static

Default Value:None. See the **add aggregate** command and “Route Aggregation” on page 230 for more information about the aggregate protocol.

route-policy-index
An integer that identifies the entry with which the match should be made.

Valid Value: 1 to 65535

Default Value:none

source-gateway *route-policy-index ip-address-and-mask*
Matches routes that come from a specified source gateway or a range of source gateways.

route-policy-index
An integer that identifies the entry with which the match should be made.

Valid Value: 1 to 65535

Default Value: none

ip-address-and-mask

Valid Values: any valid IP address and mask combination

Default Value: none

Delete

Use the **delete** command to delete route filter policy entries, match-conditions from existing route filter policy entries, or actions from existing route filter policy entries. See the **add** command in this section for a description of the parameters that can be deleted.

List

Use the **list** command to list the route filter policy entries, match conditions, and actions that exist for the route filter policy currently being changed.

Syntax: list

Example:

```
IP Route Policy Config>list

     IP Address      IP Mask           Match  Index  Type
     -----------------------------------------------------
     9.0.0.0         255.0.0.0         Range  1      Include
     10.0.0.0        255.0.0.0         Range  2      Exclude
         Match Conditions:  Protocol: BGP
     0.0.0.0         0.0.0.0           Range  3      Include
         Match Conditions:  Protocol: Static
                            Gateway IP Address Range: 153.2.2.20/255.255.255.255
```

```
10.1.1.0        255.255.255.0     Range  4      Include
0.0.0.0         0.0.0.0           Range  7      Include
    Policy Actions:    Set Manual Tag: 0xACEEACEE
0.0.0.0         0.0.0.0           Range  8      Include
    Match Conditions:  Protocol: RIP
```

Accessing the IP Monitoring Environment

Use the following procedure to access the IP monitoring commands. This process gives you access to the IP *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 terminal. If the prompt does not appear when you first enter configuration, press **Return** again.
2. At the + prompt, enter the **protocol ip** command to get you to the `IP>` prompt.

   Example:

   ```
   + prot ip
   IP>
   ```

IP Monitoring Commands

This section describes the IP monitoring commands. Table 20 lists the IP monitoring 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 20. IP Monitoring Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
Access controls	List the current IP access control mode, together with the configured access control records.
Aggregate	Displays aggregate routes.
Aggr-policy	Displays status and policy for the specified aggregate route.
Cache	Displays a table of all recent routed destinations.
Counters	Lists various IP statistics, including counts of routing errors and packets dropped.
Dscache	Lists the actions, stats, and order of the DiffServ flow cache.
Dump routing tables	Lists the contents of the IP routing table.
IGMP	Displays IGMP counters and parameters
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.
Parameters	Lists various parameter values.
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.

Table 20. IP Monitoring Command Summary (continued)

Command	Function
Redundant Default Gateway	Lists whether a redundant default gateway exists and if it is active or inactive.
Reset	Allows you to dynamically reset the IP/RIP configuration.
RIP	Displays the status of the RIP protocol.
RIP-Policy	Displays the route filter policy applied on the specified interface.
Route	Lists whether a route exists for a specific IP destination, and if so, the routing table entry that corresponds to the route.
Route-table-filtering	Lists any defined route filters and indicates whether route-filtering is enabled or disabled.
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.
UDP-Forwarding	Displays the UDP port numbers and destination IP addresses that you added using the **add** command or the **enable** command.
VRID	Displays detailed information for a specific VRID
VRRP	Lists the summary status for the VRRP protocol.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Access Controls

Use the **access controls** command to print the global access control mode in use together with a list of the configured global access control rules.

Access control is either disabled (meaning that no access control is being done and the access control rules are being ignored) or enabled (meaning that access control is being done and the access control rules are being recognized). The **set access on** talk 6 command enables access control.

Syntax:

access

Example: `access`

```
            Access Control currently enabled
            Access Control facility: USER
            Access Control run 702469 times, 657159 cache hits

            List of access control records:

   1   Type=I   Source=0.0.0.0    Dest   =0.0.0.0     Prot=17
                SMask =0.0.0.0    DMAask =0.0.0.0
                SPorts=5004-5511  DPorts=5004-5511
                       T/C=**/**  Log=N
                BypassComp BypassEnc

2  Type=E    Source=0.0.0.0          Dest=0.0.0.0           Prot= 1
             SMask =255.255.255.255  DMask=255.255.255.255  Use=18962
             Sports=  N/A            Dports=  N/A
                      T/C= 1/**      Log=Yes ELS=N SNMP=N   SLOG=L(Alert)

3  Type=I    Source=1.1.1.1          Dest=1.1.1.2           Prot= 6
             SMask =255.255.255.255  DMask=255.255.255.254  Use=42
             Sports=  2-200          Dports=  1-100
                                     Log=No

4  Type=I    Source=9.1.2.3          Dest=0.0.0.0       Prot=  0-255
```

```
                SMask =255.255.255.255  DMask=0.0.0.0      Use=0
                SPorts=  0-65535        DPorts=  0-65535
                         T/C= **/**     Log=N
                Tos=xE0/x00-x00         ModifyTos=x1F/x08
                PbrGw=9.2.160.1         UseDefRte=Y

5  Type=I       Source=0.0.0.0        Dest=0.0.0.0         Prot= 0-255
                Mask=0.0.0.0          Mask=0.0.0.0         Use=683194
                Sports=  1-65535      Dports=  1-65535
                                      Log=No
```

Exclusive (E) means that packets matching the access control rule are discarded. Inclusive (I) means that packets matching the access control rule are forwarded. When access control is enabled, packets failing to match any access control record are discarded. *Prot* (protocol) indicates the IP protocol number. *Sports* indicates the range of TCP/UDP source port numbers; *Dports* indicates the range of TCP/UDP destination port numbers. *SYN* indicates TCP connection establishment filtering. *T/C* stands for ICMP type and code; *SLOG* stands for SysLog.

The Use field 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.

In this example, access control rule number 4 has activated the TOS filter. The TOS parameters are shown. See the **add access-control** command in talk 6 for a description of these parameters.

Aggregate

Use the **aggregate** command to display the aggregate routes and whether they are active.

Syntax:

aggregate

Example: aggregate

```
IP>aggregate

                     Aggregate Routes

Address         Mask             Policy            Status      Metric
-----------------------------------------------------------------------
10.1.0.0        255.255.0.0      UNCONDITIONAL     ACTIVE          12
10.2.0.0        255.255.0.0      EAST-REGION       ACTIVE          20

Status for last dynamic reconfig: 0x04
```

Aggr-policy

Use the **aggr-policy** command to display the status and the route filter policy for the specified aggregate route.

Syntax:

aggr-policy

Example:

```
IP>aggr-policy
IP Address []? 10.2.0.0
IP Mask []? 255.255.0.0

         Aggregate route 10.2.0.0/255.255.0.0 Policy
```

```
Status: ACTIVE      Metric:      20
Sufficient Route: 10.2.1.0/255.255.255.0

Checksum 0x4859  Longest-Match Application

IP Address      IP Mask           Match  Index  Type
-----------------------------------------------------
10.2.0.0        255.255.0.0       Range  1      Include
    Match Conditions:  Protocol: Static
    Policy Actions:    Set Metric: 20
```

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
   Eth/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 zeros).

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.

Dscache

Use the **dscache** command to list the actions, stats, and order of the DiffServ flow cache.

Example: dscache actions

```
IP>dscache actions
Source          Destination     Pro ProtocolInf Net TosIn/Out Action
10.1.100.1      9.1.140.1         1 T:x08 C:x00   0 x05->x05   DROP
9.1.140.1       10.1.100.1        1 FrgId:x0008  -1 x00->x15   PASS
10.1.100.1      9.1.140.1         1 FrgId:x0008  -1 x03->x15   PASS
10.1.100.1      9.1.140.1         6  1024>   23   0 xFE->x15   PASS
9.1.140.1       10.1.100.1        1 T:x03 C:x03   1 x00->x15   PASS
10.1.100.1      9.1.140.1        17 12585>33437   0 x00->x15   PASS
10.1.100.1      9.1.140.1         1 FrgId:x0010  -1 x05->x05   DROP
9.1.140.1       10.1.100.1        6    23> 1024   1 x00->x15   PASS
9.1.140.1       10.1.100.1        1 T:x00 C:x00   1 x00->x15   PASS
10.1.100.1      9.1.140.1         1 FrgId:x0009  -1 x05->x05   DROP
```

Example: dscache stats

```
IP>dscache stats
Source          Destination     Pro ProtocolInf Net Tos     RxPkts    RxBytes
10.1.100.1      9.1.140.1         1 T:x08 C:x00   0 x05          2       4088
9.1.140.1       10.1.100.1        1 FrgId:x0008  -1 x00          1         26
10.1.100.1      9.1.140.1         1 FrgId:x0008  -1 x03          1         26
10.1.100.1      9.1.140.1         6  1024>   23   0 xFE          9        383
9.1.140.1       10.1.100.1        1 T:x03 C:x03   1 x00          1         56
10.1.100.1      9.1.140.1        17 12585>33437   0 x00          1         84
10.1.100.1      9.1.140.1         1 FrgId:x0010  -1 x05          1         26
9.1.140.1       10.1.100.1        6    23> 1024   1 x00          8        879
9.1.140.1       10.1.100.1        1 T:x00 C:x00   1 x00          8       6552
10.1.100.1      9.1.140.1         1 FrgId:x0009  -1 x05          1         26
```

Example: dscache order

```
IP>dscache order
Source          Destination     Pro ProtocolInf Net Tos
10.1.100.1      9.1.140.1         6  1024>   23   0 xFE
9.1.140.1       10.1.100.1        6    23> 1024   1 x00
9.1.140.1       10.1.100.1        1 T:x03 C:x03   1 x00
10.1.100.1      9.1.140.1        17 12585>33437   0 x00
10.1.100.1      9.1.140.1         1 FrgId:x0010  -1 x05
10.1.100.1      9.1.140.1         1 T:x08 C:x00   0 x05
10.1.100.1      9.1.140.1         1 FrgId:x0009  -1 x05
9.1.140.1       10.1.100.1        1 FrgId:x0008  -1 x00
9.1.140.1       10.1.100.1        1 T:x00 C:x00   1 x00
10.1.100.1      9.1.140.1         1 FrgId:x0008  -1 x03
```

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 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 that the route was learned through the RIP protocol.

Del - Indicates that 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 route.

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 that 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.

IGMP

Use the **igmp** command to display IGMP counters and operational parameters for IGMP.

Syntax:

igmp counters

parameters

counters

Displays the counts of IGMP messages sent and received.

Example:

```
IP+ igmp counters
     Net      Querier      Polls Sent      Polls Rcvd      Reports Rcvd
     ---      -------      ----------      ----------      ------------
      0         Y            4973              0              0
      2         N               1            4921              0
      5         Y            4972              0              0
```

Net Specifies the network number.

Querier
Specifies whether the device is the querier on the specified network.

Polls Sent
Number of IGMP queries sent.

Polls Rcvd
Number of IGMP queries received.

Reports Rcvd
Number of IGMP reports received.

parameters

Displays the operational IGMP parameters of the device's attached interfaces.

Example:

```
IP+ igmp parameters

        Net      Robustness      Query       Response     Leave Query
                  Variable      Interval     Interval      Interval
                                 (secs)       (secs)        (secs)
        ---      ----------     --------     --------     -----------
          0          2            125          10             1
          2          2            125          10             1
          5          2            125          10             1
```

Net The network number of the IGMP interface.

Robustness variable
The robustness variable of the specified interface.

Query interval
The number of seconds between IGMP general queries on that network if this device is the designated IGMP querier.

Response interval
The maximum response time inserted into IGMP general queries on that network if this device is the designated IGMP querier.

Leave query interval
The maximum response time inserted into IGMP specific queries on that network if this device is the designated IGMP querier.

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. If the bridge interface used for bridging and routing on the same interface has been assigned an IP address, it will also be listed. The bridge interface is identified by *BDG/0*.

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)           Address- MTU
   TKR/0   133.1.169.2      255.255.252.0   Unspecified
    FR/0     133.1.167.2      255.255.254.0   Unspecified
```

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. Interfaces are identified by interface numbers, except for the bridge interface used for routing and bridging on the same interface. It is identified by *BDG/0*.

Syntax:

packet-filter [*name*]

IPv4 Example: `packet-filter pf-in-0`

```
Name               Direction     Interface  State  SRC-Addr-Check #Access-Controls
pf-in-0            Out           0          On     N/A            3

Access Control is: enabled
Access Control run 563 times, 271 cache hits

List of access control records:

0  Type=IN   Source=10.1.1.1      Dest=10.1.1.2          Prot=0-255
             Mask=255.255.255.255 Mask=255.255.255.254   Use=71
             Sports=  N/A         Dports=  N/A
                                  Log=Yes ELS=N SNMP=Y SLOG=L(Emergency)
             Trace=Enabled

1  Type=I    Source=9.67.1.5      Dest=9.37.192.1        Prot=6-255
             Mask=255.255.255.255 Mask=255.255.255.255   Use=15
             Sports=  N/A         Dports=  N/A
                                  Log=Yes ELS=L SNMP=N SLOG=L(Debug)

2  Type=I    Source=0.0.0.0       Dest=0.0.0.0           Prot=0-255
             Mask=255.255.255.255 Mask=255.255.255.255   Use=477
             Sports=  0-65535     Dports=  1-65535
                                  Log=N
```

IPv6 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
  Trace=Enabled
```

Parameters

Use the **parameters** command to list the values of various parameters.

Example:

```
IP> parameters
ARP-SUBNET-ROUTING     : disabled
ARP-NET-ROUTING        : disabled
CLASSLESS              : disabled
DIRECTED-BROADCAST     : enabled
DSCACHE-SIZE           : 64 entries
ECHO-REPLY             : enabled
FRAGMENT-OFFSET-CHECK  : disabled
PER-PACKET-MULTIPATH   : disabled
REASSEMBLY-SIZE        : 12000 bytes
RECORD-ROUTE           : enabled
ROUTING TABLE-SIZE     : 768 entries (52224 bytes)
(Routing) CACHE-SIZE   : 64 entries
SAME-SUBNET            : disabled
SOURCE-ROUTING         : enabled
TIMESTAMP              : enabled
TTL                    : 64

IP>
```

Ping

Use the **ping** command to have the router send ICMP Echo messages to a given destination (that is, “pinging”) and watch for a response. This command can be used to isolate trouble in the internetwork.

Syntax:

ping *dest-addr [src-addr data-size ttl rate tos data-value]*

The ping 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 ping 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 broadcast or 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.

You can specify the size of the ping (number of data bytes in the ICMP message, excluding the ICMP header), value of the data, time-to-live (TTL) value, rate of pinging, and TOS bits to set. You can also specify the source IP address. If you do not specify the source IP address, 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.

Only the destination parameter is required; all other parameters are optional. By default the size is 56 bytes, the TTL is 64, the rate is 1 ping per second, and the TOS setting is 0. The first 4 bytes of the ICMP data are used for a timestamp. By default the remaining data is a series of bytes with values that are incremented by 1, starting at X'04', and rolling over from X'FF' to X'00' (for example, X'04 05 06 07 . . . FC FD FE FF 00 01 02 03 . . .'). These values are incremented only when the default is used; if the data byte value is specified, all of the ICMP data (except for the first 4 bytes) is set to that value and that value is not incremented. For example, if you set the data byte value to X'FF', the ICMP data is a series of bytes with the value X'FF FF FF . . .'.

Example:

```
   IP> ping
   Destination IP address [0.0.0.0]? 192.9.200.1
   Source IP address [192.9.200.77]?
   Ping data size in bytes [56]?
   Ping TTL [64]?
   Ping rate in seconds [1]?
   Ping TOS (00-FF) [0]? e0
   Ping data byte value (00-FF) [ ]?
   PING 192.9.200.77-> 192.9.200.1:56 data bytes,ttl=64,every 1 sec.
   56 data bytes from 192.9.200.1:icmp_seq=0.ttl=255.time=0.ms
   56 data bytes from 192.9.200.1:icmp_seq=1.ttl=255.time=0.ms
   56 data bytes from 192.9.200.1:icmp_seq=2.ttl=255.time=0.ms


   ----192.9.200.1 PING Statistics----
   3 packets transmitted, 3 packets received, 0% packet loss
   round-trip min/avg/max=0/0/0 ms
   IP>
IP>ping
```

Redundant Default Gateway

Use the **redundant default gateway** command to display the redundant Default IP Gateways configured for each interface.

Syntax:

redundant default gateway

Example:

```
Redundant Default IP Gateways for each interface:
   inf  3   22.2.2.6   255.0.0.0   00.00.00.00.00.AB   backup standby
   inf  4   11.1.1.6   255.0.0.0   00.00.00.00.00.BA   primary active
```

Note: Type can be "Primary" or "Backup". Status can be "Active" or "Standby".

Reset IP

Use the **reset IP** command to make effective certain IP and RIP configuration changes.

Syntax:

reset ip

Example:

```
IP>interface
Interface  IP Address(es)   Mask(s)
  Eth/0    30.1.1.2         255.255.255.0
           30.1.1.1         255.255.255.0
           153.2.2.25       255.255.255.240
 IP>
*talk 6

IP config>add address 0 5.1.1.1 255.255.0.0
IP config>
*talk 5

IP>reset ip

IP>interface
Interface  IP Address(es)   Mask(s)
  Eth/0    5.1.1.1          255.255.0.0
           30.1.1.2         255.255.255.0
           30.1.1.1         255.255.255.0
           153.2.2.25       255.255.255.240
 IP>
```

RIP

Use the **rip** command to display the RIP protocol status detail.

Syntax:

rip

Example:

```
IP>rip

                                RIP Interfaces

Interface-Addr  Interface-Mask  Version  In Out  Send-Flags      Receive-Flags
10.69.1.2       255.255.255.0      1      1   0  D,P
200.1.1.2       255.255.255.0      2      1   0  Policy,P        Policy
Send Flags: N=Network S=Subnet H=Host St=Static D=Default O=Outage-Only
            P=PoisonReverse Policy=Send-Policy
Recv Flags: N=Network S=Subnet H=Host OSt=Override-Static OD=Override-Default
            Policy=Receive-Policy

        RIP Policy

Interface-Address  Send Policy        Receive-Policy
10.69.1.2          rip-global-send    rip-global-recv
200.1.1.2          rip-send           rip-receive
RIP global receive policy: rip-global-recv
RIP global send policy: rip-global-send

RIP never originates a default route
```

RIP-Policy

Use the **rip-policy** command to display the RIP policy that is currently applicable to the specified interface.

Syntax:

rip-policy

Example:

```
IP>rip-policy
For which interface [0.0.0.0]? 200.1.1.2
```

```
Interface Send Policy: rip-send for 200.1.1.2
Checksum 0x8637  Longest-Match Application

IP Address      IP Mask           Match  Index  Type
-----------------------------------------------------
0.0.0.0         0.0.0.0           Range  1      Include
    Match Conditions:  Protocol: BGP
    Policy Actions:    Set Manual Tag: 0xACEEACEE
                       Set Metric: 3

Interface Receive Policy: rip-receive for 200.1.1.2
Checksum 0x5049  Longest-Match Application

IP Address      IP Mask           Match  Index  Type
-----------------------------------------------------
0.0.0.0         0.0.0.0           Range  1      Include
    Match Conditions:  Source Gateway IP Address Range: 200.1.1.1/255.255.255.255
```

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)
```

Route-table-filtering

Use the **route-table-filtering** command to display whether or not route table filtering is enabled and list any defined route table filters.

Syntax:

route-table-filtering

Example: **`route-table-filtering`**

```
IP>route-table-filtering
Route Filters
```

```
Destination     Mask            Match Type
10.1.1.0        255.255.255.0   BOTH  E
10.1.1.1        255.255.255.255 EXACT I
50.0.0.0        255.0.0.0       BOTH  E
50.50.0.0       255.255.0.0     BOTH  I

IP>
```

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 following example 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

```
IP>static routes
Net             Mask            Cost  Next hop
1.1.0.0         255.255.0.0     1     10.1.1.1        TKR/0
                                2     20.1.1.1        TKR/1
                                3     30.1.1.1        TKR/2
2.2.0.0         255.255.0.0     10    10.2.2.2        TKR/0
3.3.0.0         255.255.0.0     100   10.3.3.3        TKR/0
                                200   20.3.3.3        TKR/1

IP>
```

Net The destination address of the route.

Mask The destination mask of the route.

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 a default of 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 left-most 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 a default maximum of 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 *dest-addr [src-addr data-size probes wait tos max-ttl]*

dest-addr
The address at the far end of the route.

src-addr
The source address from which the trace originates.

data-size
The size in bytes of the data field of the traceroute message. The data field does not include the UDP header.

probes
Number of UDP traceroute messages sent from each hop.

wait Time in seconds between retries.

tos The setting of the TOS bits in the UDP messages. For example, a value of X'10' (B'00010000') sets the TOS bits to B'1000'. The default is 0, which sets the TOS bits to B'1000'.

max-ttl
Maximum time-to-live in seconds for each message.

Example:

```
IP> traceroute
Destination IP address [0.0.0.0]? 128.185.142.239
Source IP address [128.185.142.1]?
Data size in bytes [56]?
Number of probes per hop [3]?
Wait time between retries in seconds [3]?
Maximum TTL [32]?
Traceroute TOS (00-FF) [0]? 10

TRACEROUTE 128.185.142.1 -> 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 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.

3 * * * 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
```

VRID

Use the **VRID** command to display the detailed status for a specific virtual router identified by an interface address and VRID. Note that three gratuitous ARPs are sent when the master router notifies the hosts that it is using its burned-in hardware MAC address as the VRID virtual MAC address.

Syntax:

vrid

Example:

```
IP>vrid 153.2.2.25 1

              --- Detailed VRID Information ----

             Interface address:      153.2.2.25
             Interface mask:         255.255.255.240
             VRID:                   1
             VRID State:             MASTER
             Virtual MAC Address:    10:00:5A:63:3B:88
             Source MAC Address:     10:00:5A:63:3B:88
             Ethernet V2 Interface:  UP
             Preempt mode active

  Priority:             255       Advertise interval:   1
  Advertise Timer:      1         Skew (in ticks):      0
  Authentication Type:  NONE      Authentication Key:
  State transitions:    2         Advertisements out:   9
  Advertisements in:    0         Advertisements error: 0
  ARPs Modified:        0         Gratuitous ARPs:      3
  VRID Address:         153.2.2.25
```

VRRP

Use the **VRRP** command to display summary information

Syntax:

vrrp

Example:

```
                                  --VRID Summary--
IP address       VRID  State  Advertise Master-Dead  Address(es)
153.2.2.25          1  MASTER         1         N/A  153.2.2.25
                                                     5.1.1.1
```

IP Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

Internet Protocol (IP) supports the CONFIG (Talk 6) **delete interface** command with no restrictions.

GWCON (Talk 5) Activate Interface

IP supports the GWCON (Talk 5) **activate interface** command with the following considerations:

- IP will be activated on a newly activated network interface only if IP is already installed (that is, IP was configured on a network interface at load time).
- VRRP configuration (**add vrid** and **add vr-address**) is installed only if VRRP is already enabled (**enable vrrp**).
- Packet-filters (**add packet-filter**) and associated access control rules (**update packet-filter**, **add access-control**) are installed only if access control is already enabled (**set access-control on**).

All IP interface-specific commands are supported by the GWCON (Talk 5) **activate interface** command.

GWCON (Talk 5) Reset Interface

IP supports the GWCON (Talk 5) **reset interface** command with the following considerations:

- IP will be activated on a newly activated network interface only if IP is already installed (that is, IP was configured on a network interface at load time).
- VRRP configuration (**add vrid** and **add vr-address**) is installed only if VRRP is already enabled (**enable vrrp**).
- Packet-filters (**add packet-filter**) and associated access control rules (**update packet-filter**, **add access-control**) are installed only if access control is already enabled (**set access-control on**).

All IP interface-specific commands are supported by the GWCON (Talk 5) **reset interface** command.

GWCON (Talk 5) Component Reset Commands

IP supports the following IP-specific GWCON (Talk 5) **reset** commands:

GWCON, Protocol IP, Reset IP Command

Description:
Installs all supported IP configuration changes.

Network Effect:
None.

Limitations:
None.

All IP configuration changes are automatically activated except the following:

Commands whose changes are not activated by the GWCON, protocol ip, reset ip command
CONFIG, protocol ip, add filter
CONFIG, protocol ip, add route
CONFIG, protocol ip, change route
CONFIG, protocol ip, delete filter
CONFIG, protocol ip, delete route

CONFIG (Talk 6) Immediate Change Commands

IP supports the following CONFIG commands that immediately change the operational state of the device. These changes are saved and are preserved if the device is reloaded, restarted, or you execute a dynamically reconfigurable command.

Commands
CONFIG, protocol ip, add filter
CONFIG, protocol ip, add route
CONFIG, protocol ip, change route
CONFIG, protocol ip, delete filter
CONFIG, protocol ip, delete route
CONFIG, protocol ip, disable icmp-redirect

CONFIG, protocol ip, enable icmp-redirect
CONFIG, protocol ip, set ttl

Non-Dynamically Reconfigurable Commands

The following table describes the Internet Protocol (IP) configuration commands that cannot be dynamically changed. To activate these commands, you need to reload or restart the device.

Commands
CONFIG, protocol ip, add distributed default gateway
CONFIG, protocol ip, add redundant default gateway
CONFIG, protocol ip, add route-table-filter
CONFIG, protocol ip, delete default network-gateway
CONFIG, protocol ip, delete default subnet-gateway
CONFIG, protocol ip, delete distributed default gateway
CONFIG, protocol ip, delete redundant default gateway
CONFIG, protocol ip, delete route-table-filter
CONFIG, protocol ip, disable arp-net-routing
CONFIG, protocol ip, disable arp-subnet-routing
CONFIG, protocol ip, disable classless
CONFIG, protocol ip, disable per-packet-multipath
CONFIG, protocol ip, disable route-table-filtering
CONFIG, protocol ip, disable simple-internet-access
CONFIG, protocol ip, enable arp-net-routing
CONFIG, protocol ip, enable arp-subnet-routing
CONFIG, protocol ip, enable classless
CONFIG, protocol ip, enable per-packet-multipath
CONFIG, protocol ip, enable route-table-filtering
CONFIG, protocol ip, enable simple-internet-access
CONFIG, protocol ip, set cache-size
CONFIG, protocol ip, set default network-gateway
CONFIG, protocol ip, set default subnet-gateway
CONFIG, protocol ip, set dscache-size
CONFIG, protocol ip, set internal-ip-address
CONFIG, protocol ip, set reassembly-size
CONFIG, protocol ip, set router-id
CONFIG, protocol ip, set routing table-size

RIP Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

Routing Information Protocol (RIP) supports the CONFIG (Talk 6) **delete interface** command with the following consideration:

Route policy records associated with the interface are not automatically deleted when an interface is deleted.

GWCON (Talk 5) Activate Interface

RIP supports the GWCON (Talk 5) **activate interface** command with the following consideration:

RIP must be globally enabled before RIP can be activated on a network interface.

All RIP interface-specific commands are supported by the GWCON (Talk 5) **activate interface** command.

GWCON (Talk 5) Reset Interface

RIP supports the GWCON (Talk 5) **reset interface** command with the following consideration:

RIP must be globally enabled before RIP can be activated on a network interface.

All RIP interface-specific commands are supported by the GWCON (Talk 5) **reset interface** command.

GWCON (Talk 5) Component Reset Commands

RIP supports the following RIP-specific GWCON (Talk 5) **reset** commands:

GWCON, Protocol IP, Reset IP Command

Description:
All RIP configuration is reset including all interface send and receive policy.

Network Effect:
No network disruption.

Limitations:
None.

All RIP commands are supported by the **GWCON, protocol ip, reset ip** command.

Chapter 16. Using 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:

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). 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 the 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 (224.0.0.5);* 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.

Note: For ATM networks, RFC 1577 will allow IP to use the network as a Non-Broadcast Multi-Access network. Thus, OSPF should be configured assuming non-broadcast. If you are using LAN Emulation, the network is treated as a broadcast network, and you should configure OSPF accordingly. If you are using both RFC 1577 and LAN Emulation on a single physical interface, configure OSPF non-broadcast on the RFC 1577 interfaces (IP addresses assigned to the real interface, for example, ATM/0), and configure OSPF broadcast on virtual or emulated interfaces (IP addresses assigned to emulated or virtual interfaces, for example, TKR/0).

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 that are 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.* Allows 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. Emulated LANs over ATM treat the ATM network as a broadcast network.
- *Non-Broadcast Multi-Access (NBMA).* 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 correctly, this network requires extra configuration information about other OSPF routers attached to the non-broadcast network. Classical IP over ATM (RFC 1577) treats the ATM interface as a Non-Broadcast Multiple Access (NBMA) interface.
- *Point-to-Multipoint.* Networks that support more than two attached routers, have no broadcast capabilities, and are non-fully meshed. A Frame Relay network without PVC between all the attached routers is an example of a Point-to-Multipoint network. Like non-broadcast networks, extra configuration information about other OSPF routers attached to the network is required.

Designated Router

Every broadcast or non-broadcast 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 OSPF 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 end-point 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 (224.0.0.5) 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 238 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 type of link state advertisement known as the group-membership-LSA (type 6). 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 data-link multicast at each hop. The ARP protocol is not used. For some network technologies, mapping between IP Class D addresses and data-link multicast occurs while, for others Class D IP addresses are mapped to the data-link 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 either at 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 data-link 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 might produce unexpected failures in multicast routing.
- 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

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 345.

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:

1. Enable the OSPF protocol. In doing so, you must estimate the final size of the OSPF routing domain. (See “Enabling the OSPF Protocol”.)
2. Set the OSPF router ID. For network technologies that do not support data-link multicast or broadcast (for example, Frame Relay), the multicast datagram must be replicated by the router and forwarded as a data-link unicast. (See “Setting OSPF Router IDs” on page 329.)
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 329.)
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 333.)
5. If you want to forward IP multicasts (IP Class D addresses), enable IP multicast routing capability. (See “Multicast Forwarding” on page 336.)
6. If the router interfaces to non-broadcast networks such as (X.25, Classical IP over ATM or Frame-Relay), set additional interface parameters. (See “Setting Non-Broadcast Network Interface Parameters” on page 336 and “Configuring Wide Area Subnetworks” on page 336.)
7. If you want the router to import routes learned from other routing protocols running on this router (BGP, RIP or statically configured routes), 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 338.)
8. If you want to boot via a neighboring router over an attached point-to-point or point-to-multipoint interface, you must configure the neighbor’s IP address. Do this by adding an OSPF neighbor for the point-to-point interface’s destination.

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 might 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[100]? 200
Estimated # OSPF routers [50]? 60
Maximum Size LSA [0]? 2048
```

Normally, 2048 bytes is large enough for any link state advertisement (LSA) generated by the router. However, routers with many OSPF dial links (for example, ISDN dial links) can require a larger LSA. Additionally, in these situations, the **packet-size** may also need to be increased in the general configuration.

Setting OSPF Router IDs

Every router in an OSPF routing domain must be assigned a unique 32-bit router ID. Choose the value used for the OSPF router ID as follows:

- If you use the IP configuration **set router ID** command, the value configured is used as an OSPF router ID. The configured OSPF router ID must be one of the router's IP addresses or the internal address.
- If you use the IP configuration **set internal address** command, 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.

  You can also determine whether the internal address is advertised and set the advertised metric. The default is to advertise any configured internal IP address with a metric of 0.
- 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 34 on page 331 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.

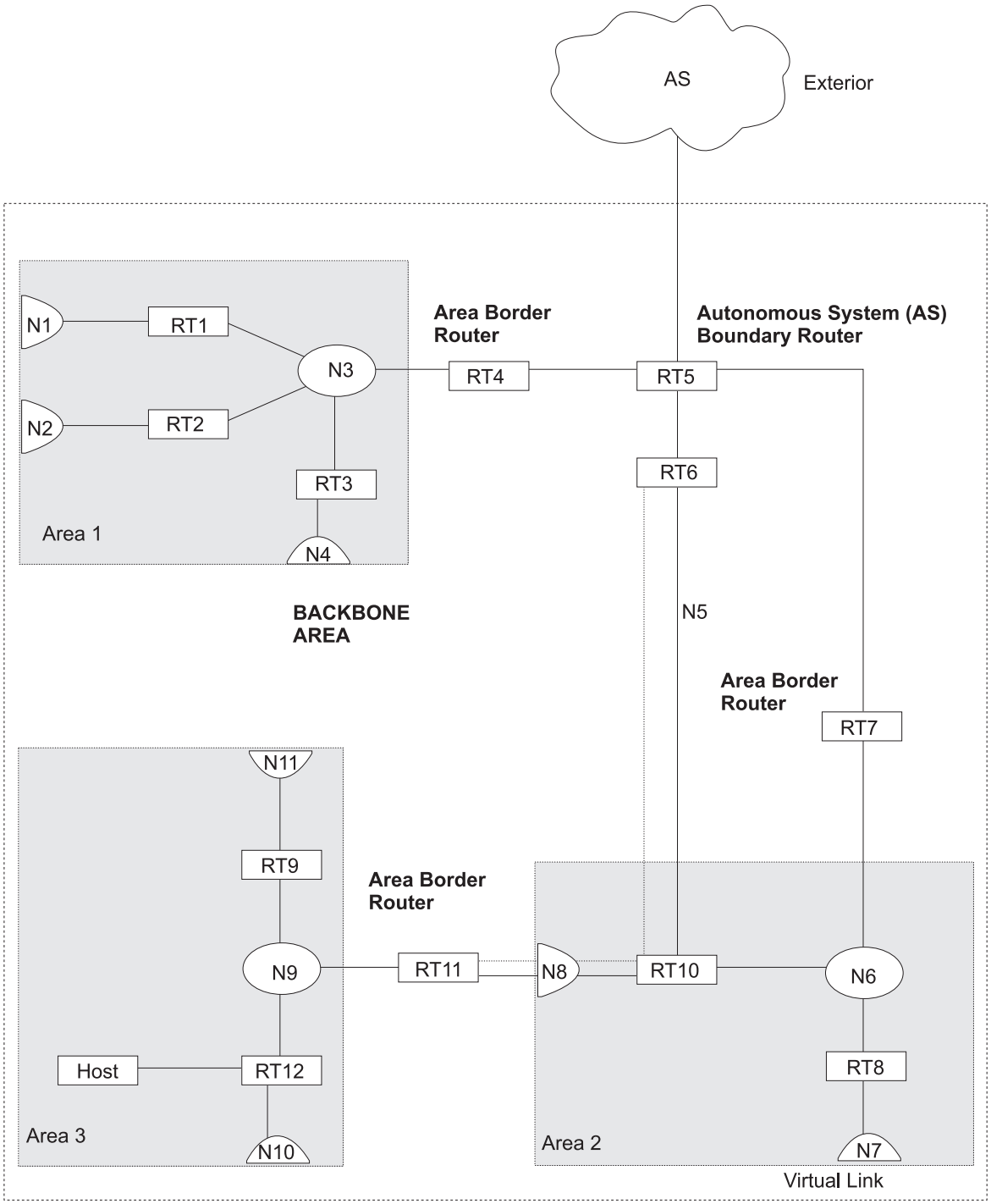


Figure 34. OSPF Areas

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.

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

Defining Attached Areas

Backbone areas, which are the default type, are flooded with all types of LSAs. You can use the **set area** command to define stub areas and not-so-stubby-areas (NSSAs) that should not be flooded with all types of LSAs. If you do not use the **set area** command, the default is that all interfaces of the router are attached to the backbone area.

A stub area is an area that allows no type 5 LSAs to be propagated into the area and instead depends on default routing to external destinations. A common type of stub area has only one router through which traffic from the stub area exits to the other devices of the network.

OSPF 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. A summary advertisement is a type 3 LSA and is used by area border routers to advertise inter-area routes. If you choose to inhibit the advertisement of summary LSAs, then the area border router will advertise a type 3 default route into the stub. As a result, traffic within the stub destined for unknown IP subnets is forwarded to the area border router over the default route. The border router uses its more complete routing information to forward the traffic on an appropriate path toward its destination.

You can 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.
3. There is no requirement for area routers to be AS boundary routers (OSPF routers that advertise routes from external sources as AS external advertisements). An external source is a network running a protocol other than OSPF.

In this case, only the area border routers and backbone routers will have to calculate and maintain AS external routes.

An area cannot be configured as a stub if it is used as a transit area for virtual links.

To set the parameters for an OSPF stub area, use the **set area** command and respond to the following prompts:

```
Area number [0.0.0.0]? 0.0.0.1
Is this a stub area? [No]:  yes
Stub default cost? [0]:
Import summaries? [Yes]:
```

The **set area** command can be also be used to define areas as NSSAs. NSSAs are described in RFC 1587. Physically, NSSAs are stub areas from an area border router's point of view and backbone areas from the point of view of routers within the NSSA. Like stub areas, NSSAs cannot be flooded with external advertisements from other areas. However, unlike stub areas, NSSAs do not depend completely upon default routes to reach external destinations. Instead, NSSA area border routers allow for external advertisements from within the area to be advertised outside the area in some cases as needed. Also unlike stub areas, NSSAs allow external advertisements from within the area to be advertised within the area.

The external advertisements local to the NSSA are advertised as type 7 LSAs , which are a type of LSA used only in NSSAs and are never advertised outside the NSSA. However, when necessary, type 7 LSAs can be summarized and readvertised as type 5 LSAs by the NSSA area border routers.

Applications of the NSSA area type include:

External route aggregation to limit route table size:
Without using NSSAs, there is no mechanism to aggregate external advertisements from many OSPF routers. Nor is there any way to keep external routes from being advertised throughout the entire OSPF routing domain.

Separate default route to firewall per area:
In many enterprise networks, it is desirable to have a separate Internet firewall in each location. This can be accomplished using NSSAs without having the area-specific default route advertised throughout the routing domain.

Implementation of ExtraNets:
Multiple enterprises can use an OSPF backbone with attached NSSAs (one per enterprise) to implement an ExtraNet (network adminstered by multiple cooperating parties) and still have flexible routing within the NSSA.

Defining Subnet Address Ranges for an Attached Area

The propagation of LSAs is also decreased by defining subnet address ranges for an area attached to an area border router. 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, the area 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 to which the border router is attached. Instead, the area border router may originate a single summary advertisement for the range or no advertisement at all, depending upon 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.

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 to which those connections are a part. 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 physical 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 to which the OSPF interface attaches. 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 can 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. To prevent OSPF Hello traffic on these LANs without other attached routers, the network can be defined as a non-broadcast multi-access network. The router priority should also be set to zero because no designated router is required.

The requirements for configuring OSPF interfaces that attach to serial lines vary with the lower layer technology.

For point-to-point lines, only one other router 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 336.

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.

Table 21. Sample Costs for OSPF Links

Interface Bandwidth	Cost
155-Mbps ATM	1
Ethernet	10
16-Mbps Token-Ring	6
4-Mbps Token-Ring	25
serial line	Cost based on bandwidth
Emulated Token-Ring (See note.)	1
Emulated Ethernet (See note.)	1

Note: An Emulated Token Ring or Ethernet will run at the interface speed (for example, 155 Mbps), and should be configured with a cost of 1.

ATM can for attach to networks at a slower rate than the maximum line speed. For example, if the router has a port that is capable of 155 Mbps, and a router connects to it with 25 Mbps, that link will still be treated as a cost of 1. The OSPF weighting is on an interface basis.

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
- Demand Circuit
- Hello Suppression
- Poll 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.

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.

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]?
```

Setting non-broadcast can also be used to force a network without any other OSPF routers to be advertised. The router priority for the interface should be set to zero and no neighbors should be defined.

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 multi-access (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 multi-hop paths with other routers as intermediates (*partial mesh connectivity*).

Non-broadcast multi-access (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, never use the **set non-broadcast** command for any interface to a point-to-multipoint network.

Point-to-multipoint can also be configured for an interface that supports broadcast capability, such as a LAN or an ATM ELAN. This configuration is useful in the following situations:
- Support of re-routing when the broadcast media is unreliable.
- Testing OSPF point-to-multipoint capabilities without having to use an expensive Frame Relay, ATM, or X.25 switch.
- Simple route prioritization using the per-neighbor cost configuration available only in point-to-multipoint topologies.

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.

Select NBMA 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 of 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 the following categories.

- BGP routes
- RIP routes
- Static routes
- Direct routes

For example, you could 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 ENABLED (subnets are imported).

If you do not import subnet routes, OSPF will import only external routes that are network routes. See “Default, Network, Subnet and Host Routes” on page 218.

You can also choose whether to import aggregate routes into the OSPF domain. See “Route Aggregation” on page 230 for more information.

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:

The syntax of the **enable as boundary routing** command is as follows:

```
enable as boundary routing
Use route policy? [No]:
Import BGP routes? [No]
Import RIP routes? [No]
Import static routes? [No]
Import direct routes? [No] yes
Import subnet routes? [Yes]
Import aggregate 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]? 10.1.1.1
```

See the command **enable as boundary routing** on page 351 for information about using a route filter policy to define AS boundary routing parameters.

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 336. Both NBMA and point-to-multipoint configurations are supported.

Configuring OSPF Over ATM (RFC 1577)

OSPF over ATM running RFC 1577 requires the following configuration steps:

1. Assign one or more IP addresses to the ATM interface using the `IP Config>` **add address** command. Each IP address corresponds to an attached Logical IP Subnet (LIS).
2. Use the `OSPF Config>` **set interface** command for each of the IP addresses configured on the ATM interface. Set the OSPF parameters including Designated-Router(DR) eligibility.
3. Use the `OSPF Config>` **set non-broadcast** command for each of the IP addresses configured on the ATM interface. This also needs to be set on all interfaces on every router that is connected to an ATM RFC 1577 LIS.
4. Use the `OSPF Config>` **add neighbor** command to define the other routers on the Logical IP Subnet (LIS) that you wish to share OSPF routing information with.

   Note: All routers that are eligible to be Designated Routers (DRs) need to be configured with the neighbor information. Only one router in every LIS needs to be DR; however, if other routers are also configured to be DR-eligible, the LIS is more capable of recovering when an outage occurs.

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 end-points. Note that you must enter OSPF router IDs in the same form as IP addresses.

```
OSPF Config>set virtual
Virtual endpoint (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 Type (0 - None, 1 - Simple) [0]? 1
Authentication Key []? 41434545
Retype Auth. Key []? 41434545
```

No cost is configured for a virtual link because the cost is the OSPF intra-area cost between the virtual link end-points through the transit area.

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 35 on page 341). 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.

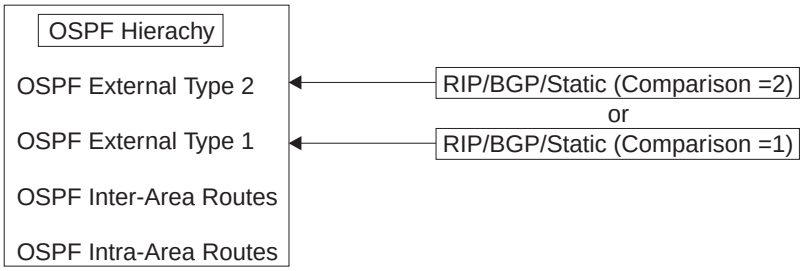


Figure 35. OSPF Routing Hierarchy

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]?
```

Demand Circuit

A demand circuit can be configured for any interface. There is no dependence on physical media or the model used by OSPF for the route calculation. When the demand circuit is configured and there are no compatibility problems:

- Only Link State Advertisements (LSAs) with real changes will be advertised over the interface. Normally, OSPF's reliable flooding algorithm causes LSAs to be refreshed with a new instance every 30 minutes even if topology changes have occurred.
- The DoNotAge bit will be set for LSAs flooded over the interface. This is required since they will not be refreshed over the interface.

Request Hello Suppression

This is an additional parameter that you can use to configure an interface to request Hello suppression. This parameter will have value for point-to-point and point-to-multipoint interfaces. In addition, the subnetwork the interface attaches to must be able to notify OSPF that data cannot be delivered over a connection. Currently, ATM and ISDN dial-on-demand interfaces are the only interface types on which Hello suppression is supported.

Poll Interval

When Hello suppression is not active, the poll interval is used only with non-broadcast multi-access subnetworks and is set with the **set non-broadcast** command. You can configure this parameter after an interface has been configured as a demand circuit and Hello suppression has been requested. This parameter will

be used by OSPF to try to reestablish a connection when a point-to-point line is down because there was a failure to transmit data but the network still appears to be operational.

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. Do this by setting 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 OSPF Configuration Parameters

OSPF configuration parameters can be changed dynamically by updating the configuration through the OSPF configuration facility and subsequently resetting the OSPF protocol through the OSPF console. OSPF neighbors, interfaces, areas, and AS boundary routing policy can be added, deleted, or changed using this technique. In most cases, these changes are completely non-disruptive. For example, adding an OSPF interface will not effect other OSPF interfaces (other than the origination of new OSPF link state advertisements).

Changes that require all of a router's OSPF advertisements to be re-originated will cause OSPF to be restarted. These include:

- Enabling/Disabling OSPF multicast forwarding (MOSPF)
- Enabling/Disabling Demand Circuits (RFC 1793)
- Changing the value of the router's Router-ID

In most cases, this will be transparent to the users as the only outage will be the time for OSPF neighbor adjacencies to be reestablished.

Since router memory is reserved for OSPF prior to allocating input/output buffers, OSPF cannot be enabled dynamically unless it was enabled at the time of the last router restart. Additionally, the amount of memory reserved for OSPF cannot be increased without a system restart. The amount of memory reserved is determined by the estimates for routers and AS external routes specified in the **enable OSPF** command.

Example:

```
  OSPF Config>enable OSPF
Estimated # external routes [100]? 300
Estimated # OSPF routers [50]? 100
Maximum Size LSA [2048]?
```

Migration from the IBM 6611 Nways Network Processor and Multiprotocol Network Program

The following enhancements allow you to migrate from existing IBM 6611 Nways® Network Processors to 2210s:

- **Least-cost area ranges**

  For OSPF summary ranges, the 6611 computes the cost based on the least cost of the component networks, while the 2210 computes the summary range cost based on the greatest cost of the component networks. **Least-cost area ranges** allows the option to compute least-cost ranges.

- **Point-to-multipoint neighbor cost**

  The 6611 supports the concept of logical point-to-point Frame Relay links, but it does not support OSPF point-to-multipoint over Frame Relay. Point-to-multipoint is more efficient, but does not allow you to specify a different cost for each neighbor. **Point-to-multipoint neighbor cost** has been added to allow an alternate TOS 0 cost to be specified for each neighbor.

Chapter 17. Configuring and Monitoring OSPF

This chapter describes how to configure the Open Shortest Path First (OSPF) Protocol. OSPF 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. This chapter includes the following sections:

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 monitoring
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.

Note: Except for the commands noted at "Dynamically Changing OSPF Configuration Parameters" on page 342, which cause OSPF to restart immediately with the changed parameters, the OSPF configuration commands are not effective immediately. They remain pending until you issue the Talk 5 **reset ospf** command.

Enter these commands at the `OSPF config>` prompt. Table 22 shows the commands.

Table 22. OSPF Configuration Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
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, demand circuit capability, or IP multicast routing.
Enable	Enables the entire OSPF protocol, AS boundary routing capability, demand circuit 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.

Table 22. OSPF Configuration Command Summary (continued)

Command	Function
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	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Response to OSPF Configuration Commands

Except for the commands noted at “Dynamically Changing OSPF Configuration Parameters” on page 342, which cause OSPF to restart immediately with the changed parameters, the OSPF configuration (Talk 6) commands do not become effective immediately. They remain pending until you issue the Talk 5 **reset ospf** command.

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	nssa range . . .
	range . . .
	neighbor . .

nssa-range *area# IP-address IP-address-mask inhibit-advertisement external-LSA-tag*

Adds a range for external address summarization to the specified not-so-stubby-area (NSSA). NSSA ranges are defined by an IP address/mask pair and define the set of external routes that are to be aggregated or confined to the NSSA. When the router is an area border router and is elected as the NSSA translator for the NSSA, the ranges will be used to originate type 5 LSAs (this is aggregation).

1. The *area number* indicates the NSSA area to which the range will be added:

   Valid Values: The area number of any configured OSPF NSSA. The **set area** command is used to configure the NSSA.

   Default Value: none

2. The *IP address* is the IP address for the NSSA range. NSSA external routes are encompassed by the range if they equal this address when logically ANDed with the IP address mask.

   Valid Values: Any valid IP address.

   Default Value: none

3. The *IP address mask* is the address mask that identifies the range routes that the range encompasses.

   Valid Values: Any valid IP address mask. However, when the mask is logically ANDed with the range IP address, the resulting value should equal that address.

   Default Value: none

4. The parameter *inhibit advertisement* indicates whether or not an OSPF type 5 advertisement is originated when there are NSSA external routes

that fall within the specified range. The default value **No** indicates that OSPF type 5 advertisements should be originated.

Valid Values: Yes or no

Default Value: No

5. The parameter *external LSA tag value* is a 4-byte hexidecimal value. It allows manual configuration of a tag to be advertised with the type 5 LSA OSPF external routes. OSPF routers receiving the advertisement can then recognize the tag and handle those routes uniquely. This parameter is not applicable if **yes** is specified for *inhibit advertisement*.

   Valid Values: X'0-9', X'A-F', and X'a-f'.

   Default Value: None. The advertised tag will be generated as if the type 5 LSA were originated by the router.

Example: add nssa-range

```
NSSA Area ID [0.0.0.1]?
IP Address [0.0.0.0]? 10.1.0.0
IP Address Mask [0.0.0.0]? 255.255.0.0
Inhibit advertisement? [No]:
External LSA Tag Value [0]? 1CD2
NSSA 0.0.0.1 Range 10.1.0.0/255.255.0.0 added or modified.
```

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. In non-broadcast multi-access networks, neighbors need to be configured only on 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. For point-to-multipoint networks, an alternate TOS 0 cost can be configured. If no cost is configured, the interface cost is used.

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]?
Alternate TOS 0 cost [0]? 100
```

1. The *Interface IP address* has:

   Valid Values: Any valid IP address.

   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 net?` For point-to-multipoint interfaces, this parameter is not applicable and should be set to “No”.

   Valid Values: Yes or No

   Default Value: Yes
4. `Alternate TOS 0 cost` allows an alternate cost to be used.

   Valid Values: 0 - 65534

   Default Value: 0 (indicates that interface cost should be used).

Delete

Use the delete command to delete OSPF information from SRAM.

Syntax:

delete
- area . . .
- interface . . .
- nssa-range . . .
- neighbor . . .
- non-broadcast . . .
- range . . .
- virtual-link

area *area#*

Deletes OSPF areas from the current OSPF configuration.

Example: `delete area 0.0.0.1`

The *area number* has:

Valid Values: Any valid area number.

Default Value: none

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

nssa range *area# IP-address IP-address-mask*

Deletes a range for external address summarization from the specified NSSA area.

1. The *area number* indicates the NSSA area from which the range will be deleted.

   Valid Values: Any configured OSPF NSSA

   Default Value: none

2. The *IP address* is the IP address for the NSSA range to be deleted.

   Valid Values: Any valid IP address.

   Default Value: none

3. The *IP address mask* is the address mask that identifies the range to be deleted.

   Valid Values: Any valid IP address mask.

   Default Value: none

Example: delete nssa-range

```
NSSA Area ID [0.0.0.1]?
IP Address [0.0.0.0]? 10.1.0.0
IP Address Mask [0.0.0.0]? 255.255.0.0
NSSA 0.0.0.1 Range 10.1.0.0/255.255.0.0 0 deleted.
```

neighbor *interface-IP-address neighbor-IP-address*

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

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

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]? 10.1.1.1
Link's transit area [0.0.0.1]? 0.0.0.2
```

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 the entire OSPF protocol, point-to-multipoint topology for an interface, demand circuit capability, or AS boundary routing capability.

Syntax:

disable	as boundary routing
	demand-circuits
	least-cost-ranges
	multicast forwarding
	ospf routing protocol
	p2mp
	rfc1583compatibility
	subnet

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`**

demand-circuits

Disables the demand circuit capability. When disabled, the router will not indicate that it supports demand circuit processing in its router link's Link State Advertisement (LSA) and will not originate any LSAs with the DoNotAge bit set. If one router in the routing domain or OSPF stub area does not support demand circuits, none of the routers in the routing domain or OSPF stub area will originate DoNotAge LSAs.

Example: **`disable demand-circuits`**

least-cost-ranges

Disables the calculation of OSPF area ranges based on the cost of the closest (lowest cost) component network. This option is disabled as a default.

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`

P2MP *IP-interface*

Overrides the point-to-multipoint (P2MP) operation for a broadcast network.

The *IP interface* is the address of an interface to a broadcast network that has been configured as a point-to-multipoint network using the **enable p2mp** command.

Valid Values: valid IP address of any configured OSPF interface

Default Value: none

RFC1583 Compatibility

Disables the AS External route selection that is compatible with RFC 1583. It is recommended that you do not disable RFC1583 compatibility unless you have the same external route accessible through more than one OSPF area and you are experiencing routing loop problems similar to those described in RFC2178. The default is enabled.

Example: `disable rfc1583Compatibility`

subnet

For an interface to a point-to-point serial line, this option disables 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> disable 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

Enable

Use the **enable** command to enable the entire OSPF protocol, the advertisement of a stub to route to a subnet, point-to-multipoint topology for a broadcast interface, demand circuit capability, or AS boundary routing capability.

Syntax:

enable	as boundary routing
	demand-circuits
	least-cost-ranges
	multicast forwarding
	ospf routing protocol
	p2mp
	rfc1583compatibility
	send outage-only
	subnet

as boundary routing

Enables the AS boundary routing capability which allows you to import

routes learned from other protocols, for example, 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 328.

If you do not import subnet routes, OSPF will import only external routes that are network routes. See “Default, Network, Subnet and Host Routes” on page 218.

You can also specify whether to import aggregate routes into the OSPF domain. See “Route Aggregation” on page 230 for more information.

An option within this command allows you to use a route filter policy to determine which routes are imported and the specific details of their advertisement including OSPF external type, metric, tag value (usually the number of the AS), and protocol. See “Route Filter Policy Configuration” on page 298 for information about configuring a route filter policy. Example 1 shows AS boundary routing configuration when not using a route filter policy and Example 2 shows AS boundary routing configuration using a route filter policy.

Example 1:

```
 enable as boundary routing
 Use route policy? [No]:
 Import BGP routes? [No]
 Import RIP routes? [No]
 Import static routes? [No]
 Import direct routes? [No] yes
 Import subnet routes? [Yes]
 Import aggregate 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]? 10.1.1.1
```

Example 2:

```
 enable as boundary routing
Use route policy? [No]: Yes
Router Policy Identifier [1-15 characters] [ ]? ospf-import
Always originate default route? [No]:
```

1. The *Use route policy* question indicates whether a configured route policy is used to determine which non-OSPF routes are imported into OSPF as OSPF external routes. If this question is answered **yes**, many of the questions are no longer displayed because they are not applicable when routing policy is configured. Routing policy provides more granularity by specifying which routes are imported.

   Valid Values: yes or no

   Default Value: no
2. The *Router Policy Identifier* question asks for the character string that identifies a configured route filter policy.

   Valid Values: a 1 to 15-character ASCII string

   Default Value: none
3. The *Import BGP* question indicates whether the BGP routes will be imported into OSPF as OSPF external routes.

   Valid Values: Yes or No

   Default Value: No
4. The *Import RIP* question indicates whether the RIP routes will be imported into OSPF as OSPF external routes.

   Valid Values: Yes or No

   Default Value: No

5. The *Import static* question indicates whether the static routes will be imported into OSPF as OSPF external routes.

   Valid Values: Yes or No

   Default Value: No
6. The *Import direct* question indicates whether the direct routes will be imported into OSPF as OSPF external routes.

   Valid Values: Yes or No

   Default Value: No
7. The *Import subnet* question indicates whether the subnet routes will be imported into OSPF as OSPF external routes.

   Valid Values: Yes or No

   Default Value: Yes
8. The *Always originate default route* question indicates whether to unconditionally originate a default route in the form of an OSPF external advertisement.

   Valid Values: Yes or No

   Default Value: No
9. The *Originate as type 1 or 2* question indicates whether the OSPF-originated default will have an AS external metric type of 1 or 2. Type 1 metrics are considered in the same context as OSPF costs while type 2 metrics are considered higher than any OSPF metric.

   Valid Values: 1 or 2

   Default Value: 2
10. 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
11. 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
```

demand-circuits

Enables demand circuit processing for the router. The router will indicate that it supports demand circuit processing in its router link's Link State Advertisement (LSA). The default is enabled so that demand circuits can be deployed without reconfiguring every router in the OSPF routing domain.

```
OSPF Config> enable demand-circuits
```

least-cost-ranges

Enables the calculation of OSPF area ranges based on the cost of the

closest (lowest cost) component network. Enabling this parameter will be necessary for compatibility with IBM 6611s acting as Area Border Routers (ABR) for the same area. It can also be used in situations where using the lowest cost component network will significantly reduce the number of OSPF LSA re-originations due to cost changes. This option is disabled as a default.

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
- Additionally, you can specify the maximum LSA size. This value may need to be increased if you have a large router with many OSPF dial links (for example, ISDN primary) in the same OSPF area. Normally, 2048 is more than enough space for any single LSA.

  Valid Values: 2048 to 65535

  Default Value: 2048

  Example: `enable OSPF routing protocol`

```
Estimated # external routes[100]? 200
Estimated # OSPF routers [50]? 60
Maximum LSA Size [2048]?
```

P2MP *IP-interface*

Overrides the default network type for a broadcast network to force point-to-multipoint operation. All routers on the subnet should have the same specification and will require neighbor definition as for a network type which is inherently point-to-multipoint, for example, Frame Relay. This option is only applicable to networks which are inherently broadcast-capable, for example, LANs and ATM ELANS.

The *IP interface* indicates the OSPF interface which is to be a P2MP network.

Valid Values: any configured OSPF interface

Default Value: none

RFC1583Compatibility

Enables the AS External route selection that is compatible with RFC 1583. The default is enabled.

Example: `enable rfc1583Compatibility`

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 immediate way (a restart/reload is not required), issue the **join** command from OSPF monitoring. Also, from OSPF monitoring, the join command keeps track of the number of times a particular group is joined. IP multicast groups joined through OSPF monitoring are not retained across router restarts and reloads.

Syntax:

join *multicast-group-address*

Example: `join 224.185.0.0`

The *multicast group address* parameter specifies the IP class D 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 OSPF monitoring. Also, from OSPF monitoring, 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 *multicast group address* parameter specifies the IP class D group/multicast address.

Valid Values: Class D IP address from 224.0.0.1 to 239.255.255.255

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
        Maximum LSA Size:       2048
        External comparison:    Type 2
        RFC 1583 compatibility: Disabled
        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
        Demand Circuits:        Enabled
        Least Cost Ranges:      Disabled
        LSA Max Random Initial Age:    0

                --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.
Maximum LSA size	Displays the maximum size LSA that will be originated by this router.
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.
RFC 1583 compatibility	Indicates whether or not OSPF AS external route is compatible with RFC 1583.
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”, and a non-zero network number is displayed in parentheses. This indicates that the default route will be 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 for the originated default route.
Multicast forwarding	Displays whether IP multicast datagrams will be forwarded.
Demand circuits	Displays whether demand circuit processing is supported.
Least Cost Area Ranges	Displays whether least cost area ranges are computed.
LSA Max Random Initial Age	Displays the maximum initial age for self-originated LSAs. If this value is zero (the default), all LSAs will be originated with an age of 0.

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.
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 end-point. “Virtual endpoint” indicates the OSPF Router ID of the other end-point. “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          Stub/NSSA Default-Cost(Type)  Inhibit-Sum/External  NSSA-Trans
0.0.0.0          Transit        N/A             N/A                   N/A
0.0.0.1          NSSA           0(2)            No /No                Elected
0.0.0.5          NSSA           5(2)            No /No                Elected

                    --NSSA Area ranges--
Area ID         Address         Mask            Advertise             Tag
0.0.0.1         193.1.1.0       255.255.255.0   No                    N/A
0.0.0.5         192.168.0.0     255.255.0.0     Yes                   0xACEEACEE
```

Area-ID	The Area address.
Stub/NSSA	Indicates whether or not the area is a tranist area, stub area, or not-so-stubby area (NSSA).
Default-cost (Type)	The cost of an Area Border Router (ABR) originated default route and, for NSSAs, the configured type.
Inhibit-Sum/External	Whether or not summary and external route advertisement is inhibited. When external route importation is inhibited, an ABR acting as an AS Boundary Router will not advertise type 7 LSAs into the area.
NSSA-Trans	Indicates whether an ABR for NSSA will participation in the NSSA type 7 LSA translation election process or unconditionally translate type 7 LSAs to type 5 LSAs.
NSSA Area ranges-Area ID	The OSPF area address that indicates the NSSA area to which the range belongs.
Address/Mask	The IP address and mask that identify the NSSA.

Advertise — Indicates whether or not an OSPF type 5 advertisement is originated when there are NSSA external routes that fall within the NSSA range.

Tag — This is the tag, if one has been configured for this NSSA range. The tag is a manually configured 4-byte hexadecimal value that is advertised with the type 5 LSA OSPF external routes. A tag exists only if the value of *Advertise* is *Yes*.

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`

```
OSPF Config>list interface

                        --Interface configuration--
IP address        Area            Auth   Cost  Rtrns  Delay  Pri  Hello   Dead
200.1.1.2         0.0.0.2           0      10      5     1     1     10     40
10.69.1.2         0.0.0.0           1       1      5     1     1     10     40
OSPF Config>list virtual-link

                        --Virtual link configuration--
Virtual endpoint     Transit area    Auth   Rtrns  Delay Hello  Dead
4.4.4.4              0.0.0.1           1       10    5     30   180
10.1.1.2             0.0.0.1           1       10    5     30   180
OSPF Config>
OSPF Config>list area

                --Area configuration--
Area ID         Stub? Default-cost Import-summaries?
0.0.0.2          No         N/A           N/A
0.0.0.0          No         N/A           N/A
0.0.0.1          No         N/A           N/A
0.0.0.3          Yes        10            Yes
```

Note: Multicast parameters are not displayed if multicast is disabled. Demand circuit parameters are not displayed if none of the interfaces are configured as demand circuits.

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 and alternate TOS 0 cost for point-to-multipoint networks.

Example: `list neighbors`

```
          --Neighbor configuration--
Neighbor Addr     Interface Address   DR eligible?      Alternate TOS 0 Cost
2.3.4.5           1.2.3.4             yes               0
2.5.6.7           5.6.7.8             no                100
```

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 end-point. “Virtual endpoint” indicates the OSPF router ID of the other end-point. “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
cost-internal-address
interface
non-broadcast
virtual-link
max-random-initial-lsa-age

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). You can set an area as a stub area or as a not-so-stubby-area (NSSA).

Example: `set area`

```
Area number [0.0.0.0]? 0.0.0.1
Is this a stub area? [No]:
Is this an NSSA Area? [No]: yes
Always perform NSSA Translation? [No]:
Import Local Externals? [Yes]:
Stub/NSSA default cost [0]?
Import summaries? [Yes]:
NSSA Type (1 or 2) for Default [2]?
```

- *Area number* - is the OSPF area address. For NSSA areas, this number indicates the NSSA area to which the range will be added.
- *Is this a stub area?* 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.

  Valid Values: yes or no

  Default Value: no

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.

- *Is this an NSSA Area?* indicates whether or not the area is an OSPF NSSA. This question is applicable only if *no* is specified for the question *Is this a stub area?*

  Valid Values: yes or no

  Default Value: no

- *Always perform NSSA translation?* indicates whether or not the router, when acting as an area border router, will unconditionally translate installed NSSA external routes into type 5 link state advertisements. If *no* is specified and the router is an area border router, it will participate in the NSSA translator election as specified in the current IETF specifications for OSPF NSSAs. This question is only applicable if *yes* is specified for the question *Is this an NSSA?*

  Valid Values: yes or no

  Default Value: no

- *Import Local Externals?* indicates whether or not routes being advertised via AS boundary import policy are also flooded into this NSSA as type 7 advertisements. This question is only applicable if *yes* is specified for the question *Is this an NSSA?* and the AS-Boundary importation is enabled.

  Valid Values: yes or no

  Default Value: no

- *Stub/NSSA Default Cost?* specifies the cost for a default flooded into the stub area or NSSA when the router is acting as an area border router. This question is only applicable if you specify *yes* for *Is this a stub area?* or *Is this an NSSA?*

  Valid Values: 0 to 16 777 215

  Default Value: 0

- *Import summaries?* indicates whether or not the router (when acting as an area border router) will inhibit the advertisement of type 3 LSAs. For a stub area, a type 3 default will always be originated by an area border router regardless of the answer to this question. For an NSSA, if you specify *no*, a type 3 default will be originated by an area border router. If *yes* is specified or defaulted, a type 7 default will be flooded into an NSSA. This question appears only if you specify *yes* for *Is this a stub area?* or *Is this an NSSA?*.

  Valid Values: yes or no

  Default Value: yes

- *NSSA Type (1 or 2) for Default* specifies the OSPF external type (1 or 2) for a default advertised into the NSSA when the router is acting as an area border router. This question is only applicable if you specify *yes* for *Is this an NSSA?* and *no* for *Import summaries?*. When summaries are not advertised into the NSSA, a type 3 default is advertised just as with a stub area.

  Valid Values: 1 or 2

  Default Value: 2

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]?
```

cost internal address

Sets the cost for internal address advertisement within the internal address network in the router's type 1 LSA. If *-1* is specified, the internal address is not advertised.

Example:

```
OSPF Config>set cost-internal-address
Internal Address Cost? [10]? 8
```

- *Internal Address Cost?* specifies the cost for the advertisement of the internal address. If *-1* is specified, the internal address will not be included in the router's type 1 advertisements.

  Valid Values: -1 to 65 535

  Default Value: 0

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.
   a. The *retransmission interval* is the interval after which a Link Request for one or more link state advertisements will be resent.

      Valid values: 1 to 65 535 seconds

      Default Value: 5

   b. 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 65 535 seconds

      Default Value: 1

   c. The *Hello Interval* is the interval between Hello packets sent on the interface.

      Valid Values: 1 to 65 535 seconds

      Default Value: 10

   d. 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 ≥ 65 535 seconds

      Default Value: 40 (or four times the configured Hello interval)
4. 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

5. The *Type of service 0 cost* is cost that will be used for the interface when the shortest path routes are computed for the area..

   Valid Values: 1 to 65 534

   Default Value: 1

6. The *Demand Circuit* indicates whether or not the interface will be treated as a demand circuit for purposes of flooding LSAs (Link State Advertisements). Over demand circuits, LSAs will be flooded with the DoNotAge bit set over this interface and will not be flooded unless there is an actual change to the LSA. Refer to RFC 1793 for more information.

   Valid Values: yes or no

   Default Value: no

7. The *Hello Suppression* indicates whether or not Hello packets will be suppressed on the interface once the neighbors reach the full state. Demand circuits must be enabled on the interface for Hello Suppression to be requested or allowed. Currently, Hello Suppression is only supported on ATM and ISDN Dial-on-Demand links. Refer to RFC 1793 for more information.

   Valid Values: allow, request, or disable

   Default Value: allow

Allow	Allows a neighbor to request Hello Suppression.
Request	Requests Hello Suppression from a neighbor.
Disable	Disables Hello Suppression and continues sending Hellos.

8. The *Demand Circuit Down Poll Interval* indicates the duration between hello polls sent when there is a failure to send data on a demand circuit with hello suppression active. Currently, hello suppression is only supported on ATM and ISDN Dial-on-Demand links. Refer to RCF 1793 for more information.

   Valid Values: 1 to 65 535

   Default Value: 60

9. The *Authentication type* defines the authentication procedure to be used for OSPF packets on the interface. The choices are 1, which indicates a simple password; or 0, which indicates that no authentication is necessary to exchange OSPF packets on the interface. When 1 is specified, the authentication key must also be specified.

   Valid Values: 0, 1

   Default Value: 0

10. 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 1-8 characters

    Default Value: a null string

Example: `set interface`

```
Interface IP address [0.0.0.0]? 10.69.1.2
Attaches to area [0.0.0.0]?
Retransmission Interval (in seconds) [5]?
Transmission Delay (in seconds) [1]? 1
Router Priority [1]? 1
Hello Interval (in seconds) [10]?
Dead Router Interval (in seconds) [40]?
Type Of Service 0 cost [1]?
Demand Circuit (Yes or NO) ?[No]:
Authentication Type (0 - none, 1 - simple) [0]? 1
Authentication Key []? AceeOSPF
Retype Auth.  Key []? AceeOSPF
```

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. This parameter specifies the interval that determines the frequency of Hellos sent to neighbors that are inactive. You

must set non-broadcast consistently across all interfaces that attach to the same subnetwork for OSPF to function correctly.

For Frame Relay or ATM networks, however, the **set non-broadcast** command is used to configure an OSPF interface as connecting to a non-broadcast multi-access network. If the **set non-broadcast** command is not used, the interface is assumed to be connected to a point-to-multipoint network. In Frame Relay networks, all OSPF interfaces must be configured as connecting to the same type of network (non-broadcast multi-access or point-to-multipoint), so if the **set non-broadcast** command is used for one router's interface, it must be configured on the interfaces for all routers attaching to the network.

Example: set non-broadcast

```
Interface IP address [0.0.0.0]? 128.185.138.19
Poll Interval [120]
```

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 end-points.

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 end-points.

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 resent.

      Valid Values: 1 to 65 535 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 65 535 seconds

      Default Value: 5

   c. The *Hello Interval* is the interval between Hello packets sent on the interface.

      Valid Values: 1 to 255 seconds

      Default Value: 30

   d. 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 65 535 seconds

      Default Value: 180

4. The *Authentication type* defines the authentication procedure to be used for OSPF packets on the virtual link. The choices are 1, which indicates a simple password; or 0, which indicates that no authentication is necessary to exchange OSPF packets on the interface. When 1 is specified, the authentication key must also be specified.

   Valid Values: 0, 1

   Default Value: 0

5. 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.

Valid Values: any 1-8 characters

Default Value: a null string

Example: **`set virtual-link`**

```
Virtual endpoint  (Router ID) [0.0.0.0]? 10.1.1.2
Link's transit area [0.0.0.1]?
Virtual link already exists - record will be modified.
```

```
Retransmission Interval (in seconds) [10]?
Transmission Delay (in seconds) [5]?
Hello Interval (in seconds) [30]?
Dead Router Interval (in seconds) [180]?
Authentication Type (0 - none, 1 - simple) [0] 1
Authentication Key []? AceeOSPF
Retype Auth. Key []? AceeOSPF
```

max-random-initial-lsa-age

Specifies the maximum initial age for self-originated LSAs. The default is 0 and normally should be modified only if you are experiencing problems with LSA origination synchronization.

Valid Values: 0 - 1770

Default Value: 0

Example:

```
OSPF Config>  set max-random-initial-lsa-age
Maximum initial LSA age [0]?
```

Accessing the OSPF Monitoring Environment

Use the following procedure to access the OSPF monitoring commands. This process gives you access to the OSPF *monitoring* process.

1. At the OPCON prompt, enter **talk 5**. (For more detail on this command, refer to "The OPCON Process and Commands" in *Software User's Guide*.) For example:

   ```
   * talk 5
   +
   ```

   After you enter the **talk 5** command, the GWCON prompt (+) displays on the terminal. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the + prompt, enter the **protocol ospf** command to get you to the `OSPF>` prompt.

   Example:

   ```
   + prot ospf
   OSPF>
   ```

OSPF Monitoring Commands

This section summarizes and then explains all the OSPF monitoring commands. These commands enable you to monitor the OSPF routing protocol. Table 23 lists the OSPF monitoring commands.

Enter the OSPF monitoring commands at the `OSPF>` prompt.

Table 23. OSPF Monitoring Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
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.

Table 23. OSPF Monitoring Command Summary (continued)

Command	Function
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.
Policy	Displays any configured AS boundary router import policy.
Reset	Resets the OSPF configuration dynamically.
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	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

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,DC
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 OSPF object corresponding to the advertisement. These capabilities include:

- **E** — Indicates that type 5 (external advertisements) are supported in area corresponding to the advertisement. This is always set for type 5 (external advertisements).
- **T** — Routing based on IP TOS (Type of Service) is supported.
- **MC** — Multicast forwarding is supported. This will only be set in advertisements originated by routers with MOSPF enabled.
- **DC** — Demand circuits are supported as described in RFC 1793.

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 view a summary display of OSPF area information or a detailed display of information for the specified area. If no area is specified or the area is not found, a summary of configured areas will be displayed. If a configured area is specified, detailed information about that area will be displayed.

Syntax:

area *area-number*

area number

The number that indicates the area for which detailed information is to be displayed.

Valid Values: An OSPF IP address that identifies any configured OSPF NSSA

Default Value: none

Example 1:

```
OSPF>area

Area ID            Type    #ifcs  #nets  #rtrs  #brdrs  DC-Status
0.0.0.1            NSSA        2      0      2       1      On
```

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.

DC-Status — Indicates whether demand circuit processing is active for the area.

Example 2:

```
OSPF>area 0.0.0.1

                         Area 0.0.0.1
      Area Index:                      0 Configured Interfaces:            2
      Type:                        NSSA Inhibit Summaries:              Yes
      Stub Default Cost:               0 NSSA Default Type:                1
      NSSA Translation:               No NSSA Always Translate:           No
      NSSA Type 7 ABR Default Type:    1 NSSA Type 7 Default Originated:  No
      Active Interfaces:               2 Demand Circuit Capability:      Yes
      Area Networks:                   0 Area Routers:                     2
      Reachable ASBRs:                 2 Reachable ABRs:                   1
      Number of LSAs:                  7 Noage LSAs:                       0
      Area Checksum:           0x0003D1CF Dynamic Config Change Flag:  0x0000
```

```
                    --Area ranges--
Address          Mask             Active Advertise Cost
153.2.0.0        255.255.0.0          No       Yes
```

Area Index
The position of the OSPF area relative to other areas. This is used internally by IBM's OSPF implementation.

Stub Default Cost
The cost for a default route flooded into the NSSA when the router is acting as an area border router (ABR).

NSSA Translation
Indicates whether or not an area border router for an NSSA is translating type 7 NSSA advertisements to type 5 external advertisements.

NSSA Type 7 ABR Default Type
When an area border router originates a type 7 default advertisement, this indicates the external metric type (1 or 2).

Active Interfaces
The number of active interfaces within this area.

Area Networks
The number of networks within this area.

Reachable ASBRs
The number of autonomous system boundary routers (ASBRs) that can be reached from this NSSA.

Number of LSAs
Number of LSAs in the area link state database.

Area Checksum
Checksum for all the LSAs in the area link state database.

Configured Interfaces
The number of configured interfaces within this area. This number includes both active and inactive interfaces.

Inhibit Summaries
Indicates whether or not the router (when acting as an area border router) will inhibit the advertisement of type 3 LSAs.

NSSA Default Type
Metric type for a type 7 NSSA default advertisement.

NSSA Always Translate
Indicates whether unconditional NSSA translation was configured for the NSSA. However, a router will only unconditionally translate when it is acting as an area border router.

NSSA Type 7 Default Originated
Indicates whether a type 7 default is currently originated by the area border router.

Demand Circuit Capability
Indicates whether or not the interface will be treated as a demand circuit for purposes of flooding LSAs. Over demand circuits, LSAs will be flooded with the DoNotAge bit set over this interface only when there is an actual change to the LSA. Refer to RFC 1793 for more information.

Area Routers
The number of routers within this area.

Reachable ABRs
The number of area border routers (ABRs) that are reachable from within this area.

Noage LSAs
Number of Noage LSAs in the area (as defined in RFC 1795).

Dynamic Config Change Flag
Flag that represents what was changed during the last dynamic re-configuration (used by development).

Address
IP address of the NSSA range.

Mask IP mask of the address.

Active Indicates whether or not the area border router has discovered type 7 advertisements in the NSSA range.

Advertise Cost
The advertised metric in the translated type 5 LSA.

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   Seq-Number  Age Unreach Xsum   Options
  5  10.13.64.0      10.1.62.1       0x80000385 1422         0x7791 E,DC
  5  10.14.64.0      10.1.62.1       0x80000385 1420         0x6B9C E,DC

                # advertisements:       2
                Checksum total:         0xE32D
```

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.

Unreach
Indicates how long the destination associated with a Link State Advertisement (LSA) that is DoNotAge has been unreachable. If the LSA is DoNotAge, *DA* will appear after the Age column before the Unreach column. If the LSA is ***not*** DoNotAge, there will be blanks.

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.

Options
These are the Link State Options, which are the optional OSPF capabilities supported by the OSPF object corresponding to the advertisement. These capabilities include:

E Indicates that type 5 (external advertisements) are supported in the area corresponding to the advertisement. This is always set for type 5 (external advertisements).

T Routing based on IP TOS (Type of Service) is supported.

MC Multicast forwarding is supported. This will only be set in advertisements originated by routers with MOSPF enabled.

DC Demand circuits are supported as described in RFC 1793.

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   Seq-Number  Age Unreach Xsum   Options
  1  10.1.62.1       10.1.62.1       0x80004963  496         0xBC15 E,DC
  1  10.1.62.2       10.1.62.2       0x800250FF    6         0xCA6F E,DC
   ⋮

                # advertisements:       99
                Checksum total:         0x2CD102
```

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.

Unreach
Indicates how long the destination associated with a Link State Advertisement (LSA) that is DoNotAge has been unreachable. If the LSA is

DoNotAge, *DA* will appear after the Age column before the Unreach column. If the LSA is ***not*** DoNotAge, there will be blanks.

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.

Options
These are the Link State Options, which are the optional OSPF capabilities supported by the OSPF object corresponding to the advertisement. These capabilities include:

E Indicates that type 5 (external advertisements) are supported in the area corresponding to the advertisement. This is always set for type 5 (external advertisements).

T Routing based on IP TOS (Type of Service) is supported.

MC Multicast forwarding is supported. This will only be set in advertisements originated by routers with MOSPF enabled.

DC Demand circuits are supported as described in RFC 1793.

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. Also, if demand circuit processing is active in the OSPF routing domain or OSPF stub area, the database checksum will most likely be different among routers with demand circuits. Refer to RFC 1793 for more information.

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 monitoring'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 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 monitoring'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
Authentication Type:    None
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
Demand Circuit   off  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

Network Capabilities:    Broadcast    Real Network
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, for example, an Ethernet interface), PP (a point-to-point network, for example, a synchronous serial line), P-2-MP (point-to-multipoint, for example, a Frame-Relay network), Multi (non-broadcast, multiaccess, for example, an X.25 connection) and 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), 64 (Designated router) or 128 (Full).

Authentication Type
Indicates the type of authentication active for the interface. Supported types are none or simple.

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.

Demand circuit
Indicates whether or not demand circuit processing is active on the 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 acknowledgments 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).

Network Capabilities
Displays the network capabilities for the interface.

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 (that is, 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 monitoring with two exceptions:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (that is, a restart/reload is not required). Similarly, the IP groups joined are not retained across router restarts and reloads.
- 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 monitoring with two differences:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (that is, a restart/reload is not required).
- The command will not delete group membership until the “leaves” executed equals the number of “joins” previously executed. Similarly, the IP multicast groups left are not retained across router restarts and unloads.

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 (for example, 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.

There is more information in a multicast forwarding cache entry. A cache entry can be displayed in detail by providing the source and destination of a matching datagram on the command line. If a matching cache entry is not found, one is built. A sample of this command is shown in Example 2.

Example 2: `mcache 128.185.182.9 224.0.1.2`

```
source Net:     128.185.182.0
Destination:    224.0.1.2
Use Count:      472
Upstream Type:  Transit Net
Upstream ID:    128.185.184.114
Downstream:     128.185.177.11 (TTL = 2)
```

In addition to the information shown in the short form of the mcache command, the following fields are displayed:

Upstream Type
Indicates the type of node from which the datagram must be received in order to be forwarded. Possible values for this field are “none” (indicating that the datagram will not be forwarded), “router” (indicating that the datagram must be received over a point-to-point connection), “transit network,” “stub network,” and “external” indicating that the datagram is expected to be received from another Autonomous System).

Downstream
Prints a separate line for each interface or neighbor to which the datagram will be sent. A TTL value is also given, indicating that datagrams forwarded out of or to this interface must have at least the specified TTL value in their IP header. When the router is itself a member of the multicast group, a line specifying “internal Application” appears as one of the downstream interfaces/neighbors.

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.

Syntax:

mstats

Example:

mstats

```
              MOSPF forwarding:       Disabled
              Inter-area forwarding:  Disabled
              DVMRP forwarding:       Enabled
              PIM forwarding:         Disabled

Datagrams received:           10143  Datagrams fwd (multicast):  10219
Datagrams fwd (unicast):          0  Locally delivered:              0
Unreachable source:               0  Unallocated cache entries:      0
Off multicast tree:               0  Unexpected DL multicast:        0
Buffer alloc failure:             0  TTL scoping:                    0
Administrative filtering:       235

# DVMRP routing entries:          5  # 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:            0
```

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.

Administrative filtering.
Displays the number of datagrams discarded because of outbound filtering.

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 entries 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 entries 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

Use the **neighbor** 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

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 2178).

```
            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
            Alternate TOS 0 cost:   5

DB summ qlen:      0  LS rxmt qlen:      0  LS req qlen: 0
Last hello:        7  No Hello         Off

# 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
Indicates 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.

Alternate TOS 0 cost
For point-to-multipoint interfaces, indicates an alternate TOS 0 cost for this neighbor. In the router's type 1 (router links) LSA, this cost will be advertised rather than the interface's TOS 0 cost.

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 acknowledgments received.

 # Nbr losses
Indicates the number of times the neighbor has changed to Down state.

 # Adj. resets
Counts entries to state ExStart.

Ping

See “Ping” on page 313 for an explanation of the **Ping** command.

Policy

Use the OSPF **policy** command to display the OSPF AS boundary route import policy for the router.

Syntax:

policy

Example:

```
  AS Boundary Importation Policy - ospf
Checksum 0x9A23  Longest-Match Application

IP Address      IP Mask           Match  Index  Type
-----------------------------------------------------
9.0.0.0         255.0.0.0         Range  1      Include
10.0.0.0        255.0.0.0         Range  2      Exclude
    Match Conditions:  Protocol: BGP
10.1.1.0        255.255.255.0     Range  4      Include
0.0.0.0         0.0.0.0           Range  0      Include
0.0.0.0         0.0.0.0           Range  3      Include
    Match Conditions:  Protocol: Static
                       Gateway IP Address Range: 153.2.2.20/255.255.255.255
0.0.0.0         0.0.0.0           Range  7      Include
    Policy Actions:    Set Manual Tag: 0xACEEACEE
0.0.0.0         0.0.0.0           Range  8      Include
    Match Conditions:  Protocol: RIP
    Policy Actions:    Set Metric: 999
```

Reset

Use the OSPF **reset** command to dynamically modify the OSP routing configuration without restarting the router. For more information see “Dynamically Changing OSPF Configuration Parameters” on page 342.

Note: During a restart, OSPF routes will be retained in the routing table to maintain IP forwarding.

Syntax:

reset ospf

Example:

```
OSPF>interface

Ifc Address      Phys    assoc. Area     Type    State   Auth   #nbrs  #adjs
153.2.2.25       Eth/0   0.0.0.1         Brdcst   16     None       3      2
10.69.1.1        FR/0    0.0.0.0         P-2-MP    8     None       1      1

OSPF>
*t 6

OSPF Config>delete interface 10.69.1.1
OSPF Config>
*t 5

OSPF>reset ospf
OSPF>interface

Ifc Address      Phys    assoc. Area     Type    State   Auth   #nbrs  #adjs
153.2.2.25       Eth/0   0.0.0.1         Brdcst   16     None       3      2
```

Traceroute

See “Traceroute” on page 318 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:

```
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:

```
# 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:

```
OSPF Router ID:         17.17.17.17
                    External comparison:    Type 2
                    RFC 1583 compatibility: Yes
                    Multicast OSPF [MOSPF]: Yes [Inter-Area Multicast Forwarder]
                    Demand circuit support: Yes
                    Least Cost Area Ranges: No
                    AS boundary capability: Yes
                    Import external routes: POLICY ospf
```

```
                    Orig. default route:    No [0,0.0.0.0]
                    Default route cost:     [1, Type 2]
                    Default forward. addr:  0.0.0.0

Attached areas:                   2  Estimated # external routes:    400
Estimated # OSPF routers:       100  Estimated heap usage:        104000
OSPF packets rcvd:            16971  OSPF packets rcvd w/ errs:    16269
Transit nodes allocated:        286  Transit nodes freed:            283
LS adv. allocated:             1439  LS adv. freed:                 1421
Queue headers alloc:             32  Queue headers avail:             32
Maximum LSA size:              2048

# Dijkstra runs:                 12  Incremental summ. updates:        0
Incremental VL updates:           0  Buffer alloc failures:            0
Multicast pkts sent:          16982  Unicast pkts sent:               10
LS adv. aged out:                 0  LS adv. flushed:                  5
Ptrs To Invalid LS adv:           0  Incremental ext. updates:        29
LSA Max Random Initial Age:       0  LSA MINARRIVAL rejects:           1

External LSA database:
Current state:               Normal
Number of LSAs:                  11  Number of overflows:              0
```

OSPF Router ID
Displays the router's OSPF ID.

External comparison
Displays the external route type used by the router when importing external routes.

RFC 1583 compatibility
Indicates whether or not OSPF AS external route computation will be compatible with RFC 1583.

Import external routes
Displays which external routes will be imported. If a route filter import policy has been configured for AS Boundary Routing, that policy is displayed.

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.

Maximum LSA size
The maximum size LSA that can be originated by this router. This is the minimum of the value configured through OSPF configuration and the maximum packet size computed or configured through general configuration.

Incremental summ updates, incremental VL updates
Indicates that new summary link advertisements have caused the routing table to be partially rebuilt.

Buffer alloc failures.
Indicates 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.

Ptrs to Invalid LS adv
Displays number of advertisements in the database which were malformed and could not be interpreted.

Incremental ext. updates.
Displays number of changes to external destinations that are incrementally installed in the routing table.

LSA Max Random Initial Age
Displays number of maximum initial random age for self-originated LSAs.

LSA MINARRIVAL Rejects
Displays number of LSAs rejected due to receiving a new instance in less than MINARRIVAL (1 second).

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`

OSPF Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

OSPF supports the CONFIG (Talk 6) **delete interface** command with no restrictions.

GWCON (Talk 5) Activate Interface

OSPF supports the GWCON (Talk 5) **activate interface** command with the following consideration:

If OSPF was not running before, but has now been configured, you need to reboot to enable OSPF and activate the new interface.

All OSPF interface-specific commands are supported by the GWCON (Talk 5) **activate interface** command.

GWCON (Talk 5) Reset Interface

OSPF supports the GWCON (Talk 5) **reset interface** command with the following considerations:

- If OSPF was not running before, but has now been configured, you need to reboot to enable OSPF and activate the new interface.
- The2210 may try to allocate memory.

All OSPF interface-specific commands are supported by the GWCON (Talk 5) **reset interface** command.

GWCON (Talk 5) Component Reset Commands

OSPF supports the following OSPF-specific GWCON (Talk 5) **reset** commands:

GWCON, Protocol OSPF, Reset OSPF Command

Description:

Re-reads the SRAM and reinitializes OSPF.

Network Effect:

If multicast support is enabled or disabled, or if demand circuit support is enabled or disabled, or if the router ID is changed, OSPF will be restarted. All adjacencies will be lost.

Limitation:

If the amount of heap memory required is larger than that currently available, because of an increased number of routes or routers, the router must be restarted.

All OSPF commands are supported by the **GWCON, protocol ospf, reset ospf** command.

GWCON, Protocol IP, Reset IP Command

Description:

Re-reads the SRAM and reinitializes OSPF.

Network Effect:

If multicast support is enabled or disabled, or if demand circuit support is enabled or disabled, or if the router ID is changed, OSPF will be restarted. All adjacencies will be lost.

Limitation:

If the amount of heap memory required is larger than that currently available, because of an increased number of routes or routers, the router must be restarted.

All OSPF commands are supported by the **GWCON, protocol ip, reset ip** command.

GWCON (Talk 5) Temporary Change Commands

OSPF supports the following GWCON commands that temporarily change the operational state of the device. These changes are lost whenever the device is reloaded, restarted, or you execute any dynamically reconfigurable command.

Commands
GWCON, protocol ospf, weight
GWCON, protocol ospf, join
GWCON, protocol ospf, leave

Chapter 18. Using BGP4

This chapter describes how to use the Border Gateway Protocol (BGP) using the BGP configuration commands.

This chapter contains the following sections:

Border Gateway Protocol Overview

BGP is an exterior gateway routing protocol used to exchange network reachability information among autonomous systems. An AS is essentially a collection of routers and end nodes that operate under a single administrative organization. Within each AS, routers and end nodes share routing information using an interior gateway protocol. The interior gateway protocol may be either RIP or OSPF.

BGP was introduced in the Internet in the loop-free exchange of routing information between autonomous systems. Based on Classless Inter-Domain Routing (CIDR), BGP has since evolved to support the aggregation and reduction of routing information.

In essence, CIDR is a strategy designed to address the following problems:
- Exhaustion of Class B address space
- Routing table growth

CIDR eliminates the concept of address classes and provides a method for summarizing *n* different routes into single routes. This significantly reduces the amount of routing information that BGP routers must store and exchange.

Note: IBM only supports the latest version of BGP, BGP4, which is defined in RFC 1654. All references to BGP in this chapter and on the interface of IBM’s routers are to BGP4, and do not apply to previous versions of BGP.

How BGP4 Works

BGP is an inter-autonomous system routing protocol. In essence, BGP routers selectively collect and advertise reachability information to and from BGP neighbors in their own and other autonomous systems. Reachability information consists of the sequences of AS numbers that form the paths to particular BGP speakers, and the list of IP networks that can be reached via each advertised path. An AS is an administrative group of networks and routers that share reachability information using one or more Interior Gateway Protocols (IGPs), such as RIP or OSPF.

Routers that run BGP are called BGP speakers. These routers function as servers with respect to their BGP neighbors (clients). Each BGP router opens a passive TCP connection on port 179, and listens for incoming connections from neighbors at this well-known address. The router also opens active TCP connections to enabled BGP neighbors. This TCP connection enables BGP routers to share and update reachability information with neighbors in the same or other autonomous systems.

Using BGP4

Connections between BGP speakers in the same AS are called internal BGP (IBGP) connections, while connections between BGP speakers in different autonomous systems are called external BGP (EBGP) connections.

A single AS may have one or many BGP connections to outside autonomous systems. Figure 36 shows two autonomous systems. The BGP speaker in AS1 is attempting to establish a TCP connection with its neighbor in AS2. Once this connection is established, the routers will be able to share reachability information.

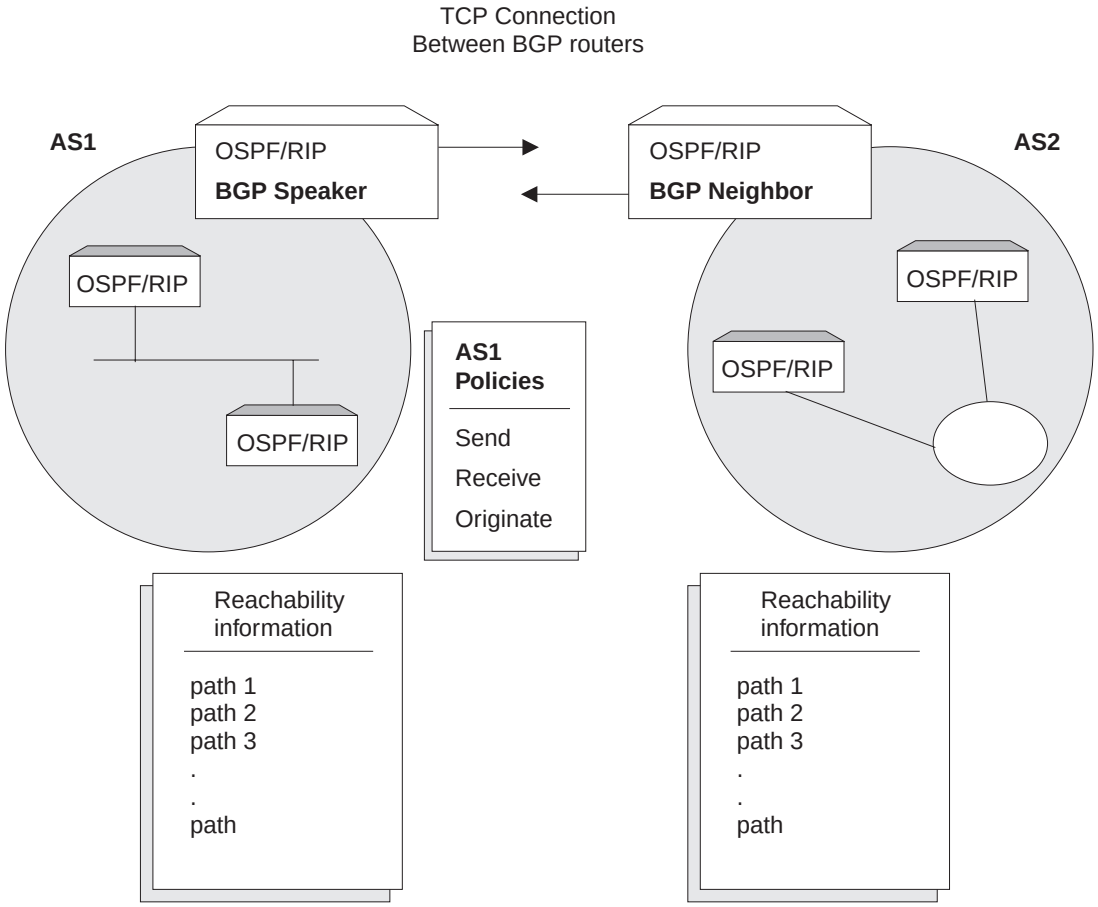


Figure 36. BGP Connections between Two Autonomous Systems. Once the BGP speaker in AS1 establishes a TCP connection with its BGP neighbor in AS2, the two routers can selectively exchange reachability information. The information each router sends or accepts is determined by policies defined for each router.

While the autonomous systems shown in Figure 36 have only one BGP router, each could have multiple connections to other autonomous systems. As an example of this, Figure 37 on page 395 shows three interconnected autonomous systems. AS1 has three BGP connections to outside autonomous systems: one to AS2, one to AS3 and one to ASx. Similarly, AS3 has connections to AS1, AS2 and to ASy.

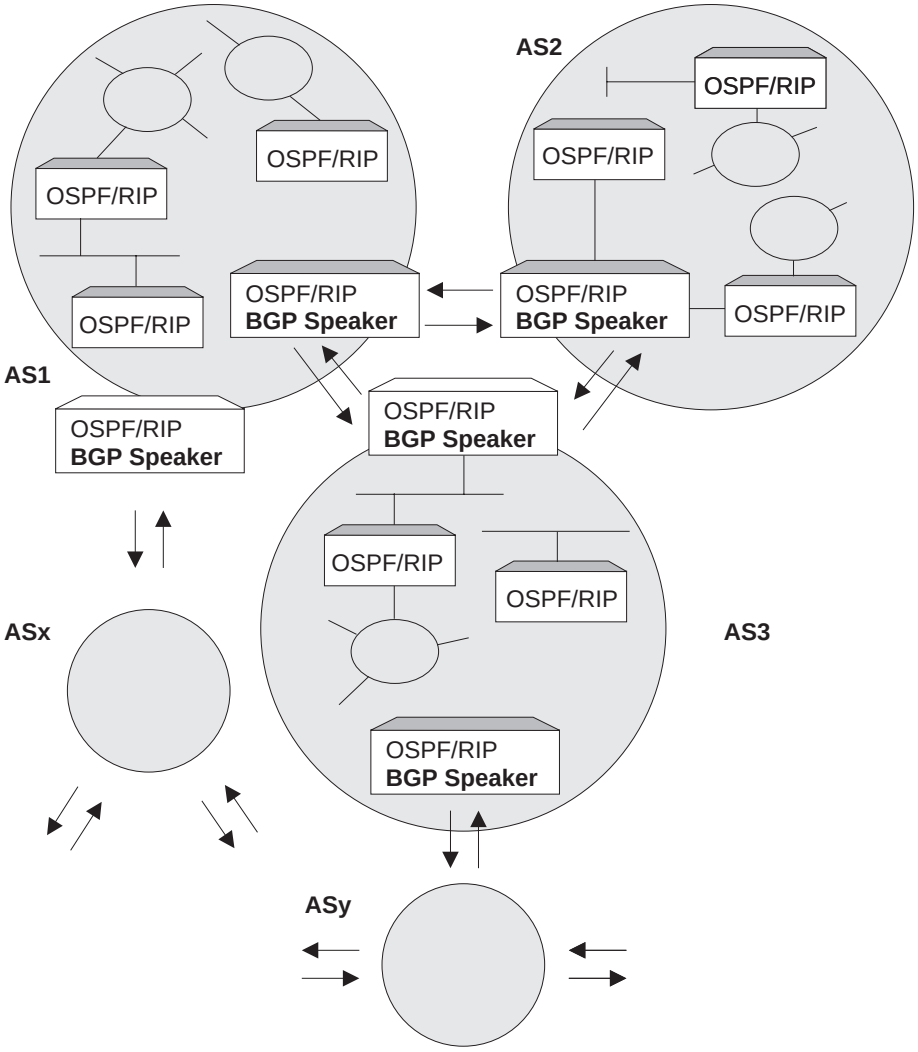


Figure 37. BGP Connections among Three Autonomous Systems. Note that AS1 and AS3 have two BGP speakers.

Once a TCP connection is established, the BGP speaker shown in Figure 36 on page 394 can send its entire routing table to its BGP neighbor in AS2. However, for security or other reasons, it may not be desirable to send reachability information on each network to AS2. Similarly, it may not be desirable for AS2 to receive reachability information on each network in AS1.

Originate, Send, and Receive Policies

Decisions on which reachability information to advertise (send), and which to accept (receive) are made on the basis of explicitly defined policy statements. IBM's BGP implementation supports three types of policy statements:

- Originate Policies
- Send Policies - There are two types of send policies
  - AS based send policies are applied only to a particular AS or all ASs. If no send policies are configured then the destination address is dropped.
  - Neighbor based send policies are applied only to a particular neighbor or neighbors. If there is no neighbor based send policies configured for a particular neighbor, then AS based send policies are applied. If a neighbor based send policy is configured, then AS based send policy is ignored.

Each send policy statement contains the destination network advertisement classifier and a set of associated actions.

The destination network classification is based on:

- Exact destination network
- Range of destination networks
- Originating AS number
- Any AS number found in AS path attribute

The possible actions are:

- Exclude destination network for advertisement
- Include destination network for advertisement to specific AS or all ASs (using AS based policy) or to a specific neighbor (using neighbor based policy)
- Set the MED value
- ASpath padding

Note: MED and ASpath padding are only applicable to a neighbor based policy.

MED attribute value hints to external BGP neighbors about its route preference. Routes with the lowest MED attribute value will be preferred. See “Route Preference Process” on page 401 for more information.

- ASpath padding allows you to add additional local AS numbers multiple times (1 to 10) to the BGP route’s ASpath. Route with the lowest ASpath will be preferred. See “Route Preference Process” on page 401 for more information.
- Receive Policies - there are two types of receive policies.
  - AS based receive policies are applied only to a particular AS or all ASs. If no receive policies are configured then the destination address is dropped.
  - Neighbor based receive policies are applied only to a particular neighbor or neighbors. If there is no neighbor- based receive policy configured for a particular neighbor, AS based receive policies are applied. If neighbor based receive policies are configured, AS based receive policies are ignored.

  Each receive policy statement contains the destination network advertisement classifier and a set of associated actions.

  The destination network classification is based on:

  - Exact destination network
  - Range of destination networks
  - Originating AS number
  - Any AS number found in AS path attribute

  The possible actions are:

  - Exclude destination network
  - Include destination network from a specific AS or all ASs (using AS based policy) or from a specific neighbor (using neighbor based policy)
  - Reset the MED value
  - Set weight value
  - Set IGP metric value
  - Set local preferences value.

  Note: MED, weight, and local preferences are only applicable to a neighbor based policy.

Weight value hints to local BGP routers to select the route based on highest weight value and ignores the route preference algorithm.

BGP Messages

BGP routers use four kinds of messages to communicate with their neighbors: OPEN, KEEP ALIVE, UPDATE, and NOTIFICATION messages.

OPEN

Open messages are the first messages transmitted when a link to a BGP neighbor comes up and establishes a connection.

KEEP ALIVE

Keep alive messages are used by BGP routers to inform one another that a particular connection is alive and working.

UPDATE

Update messages contain the interior routing table information. BGP speakers send update messages only when there is a change in their routing tables.

NOTIFICATION

Notification messages are sent whenever a BGP speaker detects a condition that forces it to terminate an existing connection. These messages are advertised before the connection is transmitted.

Setting Up BGP4

Setting up BGP involves three basic steps:

1. “Enabling BGP”.

   Enabling BGP requires you to specify the BGP router’s unique AS Number. AS numbers are assigned by Stanford Research Institute Network Information Center.

2. “Defining BGP Neighbors” on page 398.

   BGP Neighbors are BGP routers with which a BGP speaker establishes a TCP connection. Once neighbors are defined, connections to them are established by default.

3. “Adding Policies” on page 398.

   The *policies* you establish determine which routes will be imported and exported by the BGP speaker. You can set up policies for different purposes. See “Sample Policy Definitions” on page 398 for more information.

Enabling BGP

You enable BGP using the **enable BGP speaker** command as shown.

```
BGP Config> enable BGP speaker
AS [0]? 167
TCP segment size [1024]?
```

The *AS number* must be in the range 1 to 65535. The *TCP segment* size must be in the range 1 to 65535. The default value for *TCP segment* is 1024. This number represents the maximum segment size BGP will use for passive TCP connections.

After you have issued the **enable bgp** command you must reboot the device to enable BGP.

Defining BGP Neighbors

After enabling a BGP speaker, you must define its neighbors. BGP neighbors can be internal or external. Internal neighbors exist in the same AS and do not need to have a direct connection to one another. External neighbors exist in different autonomous systems. These must have a direct connection to one another.

To define internal or external BGP neighbors, use the **add neighbor** command. You must specify the IP address of the neighbor, and assign an AS number to the neighbor as shown below. Internal neighbors must have the same AS number as the BGP speaker.

```
BGP Config> add neighbor 192.0.190.178
AS [0]? 178
Init timer [12]? 30
Connect timer [120]?
Hold timer [90]? 30
TCP segment size [1024]? 512
```

Use the **reset neighbor** command to activate the specified BGP neighbor, based on the neighbor configuration parameters stored in the configuration memory.

Adding Policies

IBM's BGP implementation supports three policy commands:

- *Originate Policy*. This enables you to select the interior gateway protocol (IGP) networks to export.
- *Receive Policy*. This enables you to select the route information to import from BGP peers.
- *Send Policy*. This enables you to select the route information to export to peers. Note that exportable route information can include information collected from neighboring autonomous systems, as well as the routes that originate in the IGP.

If you added or modified a neighbor based policy use the **reset neighbor** command to activate the neighbor policy. If you added or modified an AS-based policy you must reboot the device.

Sample Policy Definitions

This section provides a set of examples of some specific policies you can set up for a BGP speaker. All policies are defined using the BGP **add** command. See "Add" on page 403 for the syntax of the **add** command.

Originate Policy Examples

Include All Routes for Advertisement

This example includes all routes in the BGP speaker's IGP routing table for advertisement. In this sense, you can view this command as the "default" originate policy statement for BGP.

Notice that the command specifies a range of addresses, rather than a single (exact) address.

```
BGP Config> add originate-policy inclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Tag [0]?
```

Exclude a Range of Routes

This example also specifies a range, but in this case the goal is to prevent the BGP Speaker from advertising addresses in this range to its neighbors.

This example excludes all routes in the range 194.10.16.0 to 194.10.31.255 from the IGP routing table, which in turn prevents them from being advertised.

```
BGP Config> add originate-policy exclusive
Network Prefix [0.0.0.0]? 194.10.16.0
Network Mask [0.0.0.0]? 255.255.240.0
Address Match (Exact/Range) [Exact]? range
Tag [0]?
```

The tag is the received RIP information. You can select networks based on a particular tag value for advertisement. See the description of the **Set** command in “Chapter 15. Configuring and Monitoring IP” on page 243 for information on setting the tag value.

Include All Classless Route for Advertisements

By default, only classfull routes from the BGP speaker’s IGP routing table will be selected for advertisement. To select both a classless and classful route for subnet advertisement, use the **enable classless-bgp** or **patch bgp-subnets** command.

AS Based Receive Policy Examples

Import all Routes from all BGP Neighbors

This example ensures that the BGP speaker will import all routes from all of its neighbors into its IGP routing table.

```
BGP Config> add receive-policy inclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Originating AS# [0]?
Adjacent AS# [0]?
IGP-metric [0]?
```

IGP-metric specifies the metric value with which the accepted routes are imported into the speaker’s IGP routing table. You are only prompted to enter a value for IGP-metric only when setting up a policy for route inclusion.

If *IGP-metric* is -1, these routes will not be imported into IGP; thus, routes are not re-advertisable.

Block Specific Routes from an Originating AS

This example will prevent the BGP speaker from importing any routes originating at AS 168 from neighboring AS 165. You might use this command if you do not want the BGP speaker to receive any routes from AS 168 for security reasons.

```
BGP Config> add receive-policy exclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Originating AS# [0]? 168
Adjacent AS# [0]? 165
```

Block Specific ASpath

This example will prevent the BGP speaker from importing any route that has AS 175 in its ASpath list.

```
BGP Config> add no-receive
Enter AS: [0]? 175
```

Neighbor Based Receive Policy Examples

Import all routes from a specific BGP neighbor, set weight = 100

This example will allow you to import all routes from BGP neighbor 192.0.190.178. All routes will have a weight value of 100 and IGP-metric value of 1.

Define the policy list name for receive policy.

```
BGP Config> add policy-list
Name[]?S1_100_r
Policy Type(Receive/Send)[Receive]?Receive
```

Attach the defined receive policy list name to a specific neighbor.

```
BGP Config> attach policy-to-neighbor
Neighbor address [0.0.0.0]?192.0.190.178
First receive policy list name (none for global AS based policy)[]?S1_100_r
Second receive policy list name (none for exit)[]?
```

Add receive policies for neighbor using **update** and **add** command.

```
BGP Config>update policy S1_100_r
Policy-list S1_100_r Config>add
Policy type (Inclusive/Exclusive) [Exclusive]? inclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Range]?
Originating AS# [0]?
Any AS# [0]?
MED [0]?
Weight [0]? 100
Local-Pref [0]?
IGP-metric [0]? 1
```

AS Based Send Policy Examples

Restrict Route Advertisement to a Specific AS

This example restricts the BGP speaker. The speaker cannot advertise routes in the address range 143.116.0.0 to 143.116.255.255, that originate from AS 165, to autonomous system 168.

```
BGP Config> add send exclusive
Network Prefix [0.0.0.0]? 143.116.0.0
Network Mask [0.0.0.0]? 255.255.0.0
Address Match (Exact/Range) [Exact]? range
Tag [0]? 165
Adjacent AS# [0]? 168
```

Advertise All Known Routes

This example ensures that the BGP speaker will advertise all routes originated from its IGP, and all routes learned from its neighboring autonomous systems.

```
BGP Config> add send policy inclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Tag [0]?
Adjacent AS# [0]?
```

Neighbor Based Send Policy Examples

Advertise All Known Routes to a Specific Neighbor with MED Attribute value = 100

This example will allow you to advertise all routes to a BGP neighbor 192.0.190.178. All advertise routes will have a MED value of 100.

Define the policy list name for send policy.

```
BGP Config> add policy-list
Name[]?S1_100_s
Policy Type(Receive/Send)[Receive]?Send
```

Attach the defined send policy list name(s) to a specific neighbor.

```
BGP Config> attach policy-to-neighbor
Neighbor address [0.0.0.0]?192.0.190.178
First send policy list name (none for global AS based policy)[]?S1_100_s
Second send policy list name (none for exit)[]?
```

Add the send policies for neighbor using the **update** and **add** commands.

```
BGP Config>update policy S1_100_s
Policy-list S1_100_s Config>add
Policy type (Inclusive/Exclusive) [Exclusive]?
Network prefix [0.0.0.0]?
Network mask [0.0.0.0]?
Address match (exact/range) [range]?
Originating AS# [0]?
TAG [0]?
MED [0]? 100
# of AS to pad [0]?
```

Route Preference Process

When the BGP speaker receives a path for particular destination from its peer, BGP goes through the following process for selecting a best possible path:

- Applies receiving policies based on configuration.
- If a destination is permitted by receiving policies, then it calculates Degree of Preference for the received destination, based on shorter ASpath length and Origin type.
- If there are several paths to the same destination then, it executes the path selection process. It selects the best possible path by comparing the new path with the existing selected best path. If the new path is selected as the best path, then it installs the new path in the IP forwarding table.
- BGP advertises the selected best path to its External and Internal BGP peers, subject to send policies.

Path Selection Process

The best path is selected based on the following order:

- Prefer the path that has been originated by this router.
- If path is not originated by this router, then prefer the path which has highest configured Weight value.
- If paths have same weight value then, prefer the path which has highest configured local-preference value.
- If paths have same local-preference value, then prefer the path which has highest Degree of Preference.
  - The path which has shortest ASpath length is given higher degree of preference.
  - If paths have same ASpath length, then Origin type IGP is preferred over EGP and Incomplete.
- If paths have same Degree of Preference, then prefer the path which has the lowest MED attribute value.
- If paths have same MED attribute value, then prefer External(EBGP) over internal (IBGP) route.
- If paths are still same, then prefer the path with lowest BGP-ID.

Chapter 19. Configuring and Monitoring BGP4

This chapter describes the BGP configuring and monitoring commands and includes the following sections:

Accessing the BGP4 Configuration Environment

To access the BGP configuration environment, enter the following command at the `Config>` prompt:

```
Config> Protocol BGP
BGP Config>
```

BGP4 Configuration Commands

This section describes the BGP configuration commands. These commands allow you to modify the BGP protocol behavior to meet your specific requirements. Some amount of configuration is necessary to produce a fully functional BGP router. Enter BGP configuration commands at the `BGP config>` prompt.

Table 24. BGP Configuration Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
Add	Add BGP neighbors and policies.
Attach	Attaches receive and send policy-list to a particular neighbor.
Change	Modifies information that was originally entered with the **add** command.
Delete	Deletes BGP configuration information that had been entered with the **add** command.
Disable	Disables certain BGP features that have been turned on by the **enable** command.
Enable	Enables BGP speakers, BGP neighbors or Classless BGP.
List	Displays BGP configuration items.
Move	Changes the order in which policies and aggregates are defined.
Set	Sets the IP-route-table-scan-timer.
Update	Manipulates a policy in a configured policy-list name using the submenu **add**, **delete**, **change** and **move** commands.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

Add

Use the **add** command to add BGP information to your configuration.

Syntax:

add

aggregate . . .
neighbor . . .
no-receive asnum . . .
originate-policy . . .
policy-list . . .
receive-policy . . .
send-policy. . .

aggregate *network prefix network mask*

The **add aggregate** command causes the BGP speaker to aggregate a block of addresses, and advertise a single route to its BGP neighbors. You must specify the network prefix common to all the routes being aggregated and its mask. The following example illustrates how to aggregate a block of addresses from 194.10.16.0 through 194.10.31.255.

1. The *Network Prefix* is the addresses being affected. The prefix is the first address in a range of addresses specified in a BGP policy.
   Valid Values: Any valid IP address.
   Default Value: none
2. The *Network Mask* applies to the address specified in Network Prefix to generate an address used in a BGP policy.
   Valid Values: Any valid IP address.
   Default Value: none

Example:

```
add aggregate
Network Prefix [0.0.0.0]? 194.10.16.0
Network Mask [0.0.0.0]? 255.255.240.0
```

When you add an aggregate definition, remember to define a policy to block the aggregated routes from being exported. If you do not, the router will advertise both the individual routes and the aggregate you have defined. This does not apply when you are aggregating the routes, which are originated from it's IGP routing table.

neighbor *neighbor IP address as# init timer connect timer hold timer keep alive timer tcp segment size*

Use the **add neighbor** command to define a BGP neighbor. The neighbor can be internal to the BGP speaker's AS, or external. To activate this neighbor dynamically use the **reset neighbor** command from BGP monitoring.

1. The IP address is the address of the neighbor you wish to peer with. It could be within your own autonomous system or in another autonomous system. If it is an external neighbor, both BGP speakers must share the same network. There is no such restriction for internal neighbors. The address has:
   Valid Values: Any valid IP address.
   Default Value: none
2. The AS number is your own autonomous system number for internal neighbor or neighbor's autonomous system number. The AS number of the neighbor has:
   Valid Values: An integer in the range of 0 - 65535
   Default Value: none
3. The *Init timer* to specifies the amount of time the BGP speaker waits to initialize resources and reinitiate transport connection with the neighbor in case the speaker has previously changed to IDLE state due to an error. If the error persists, this timer increases exponentially.
   Valid Values: 0 to 65535 seconds.

Default Value: 12 seconds

4. The *Connect timer* specifies the amount of time the BGP speaker waits to reinitiate transport connection to its neighbor, if the TCP connection fails while in either CONNECT or ACTIVE state. In the mean time, the BGP speaker continues to listen for any connection that may be initiated by its neighbor.
   Valid Values: 0 to 65535 seconds.
   Default Value: 120 seconds
5. Enter the *Hold timer* to specify the length of time the BGP speaker waits before assuming that the neighbor is unreachable. Both neighbors exchange the configured information in OPEN message and choose the smaller of the two timers as their negotiated Hold Timer value.

   Once neighbors have established BGP connection, they exchange Keepalive messages at frequent intervals to ensure that the connection is still alive and the neighbors are reachable. The Keep-Alive timer interval is calculated to be one-third of the negotiated hold timer value. Hence the hold timer value must be either zero or at least three seconds.

   Note that on switched lines, you may wish to have the Hold Timer value of zero to save bandwidth by not sending Keepalives at frequent intervals.
   Valid Values: 0 to 65535 seconds.
   Default Value: 90 seconds
6. The *TCP segment size* specifies the maximum data size that may be exchanged on the TCP connection with a neighbor. This value is used for active TCP connection with the neighbor.
   Valid Values: 0 to 65535 bytes.
   Default Value: 1024 bytes

Example:

```
add neighbor

Neighbor address [0.0.0.0]? 192.0.251.165
AS [0]? 165
Init timer [12]?
Connect timer [120]?
Hold timer [90]?
TCP segment size [1024]?
```

no-receive *asnum*

Use the **add no-receive asnum** to exclude AS-paths if the particular AS number appears anywhere inside the AS-path list.

The *AS number* has:
Valid Values: 0 to 65535
Default Value: none

Example:

```
add no-receive

Enter AS: [0]? 178
```

originate-policy *(exclusive/ inclusive) network prefix network mask address match (Exact/Range) tag*

Use the **add originate-policy** command to create a policy that determines whether a specific address, or range of addresses, can be imported to the BGP speaker's routing table from the IGP routing table.

Exclusive

Exclusive policies prevent route information from being included in the BGP speaker's routing table.

Inclusive
Inclusive policies ensure that specific routes will be included in the BGP speaker's routing table.

Network prefix
The network prefix for the addresses being affected.
Valid Values: Any valid IP address.
Default Value: none

Address match
The address, or range of addresses, that will be affected by the policy statement.Enter the *Network Mask* to be applied to the address specified in Network Prefix to generate an address used in a BGP policy.
Valid Values: Any valid IP address.
Default Value: none

Tag The value that has been set for a particular AS. All tag values match that of the AS from which they were learned.
Valid Values: 0 to 65535
Default Value: none

The following example includes all routes in the BGP speaker's IGP routing table to be advertised.

Example:
add originate-policy exclusive

```
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Exact]? range
Tag [0]?
```

See "Originate Policy Examples" on page 398 for detailed examples of this policy command.

policy-list
Use the **add policy-list** command to configure a group of policy, which can be attached to a specific neighbor using the **attach policy-to-neighbor** command.

Example: add policy-list

```
Name[]? nbr1-rcv
Policy Type(Receive/Send)[Receive]?Receive
```

Example: add policy-list

```
Name[]? nbr1-snd
Policy Type(Receive/Send)[Receive]?Send
```

Note: See "Neighbor Based Receive Policy Examples" on page 400 and "Neighbor Based Send Policy Examples" on page 400 for detailed examples of this policy command.

receive-policy *(exclusive/ inclusive) network prefix network mask address match originating as# adjacent as# igpmetric (inclusive only)*

Use the **add receive-policy** command to determine what routes will be imported to the BGP speaker's routing table.

Exclusive policies prevent route information from being included in the BGP speaker's routing table.

1. The *Network Prefix* is the addresses being affected.
   Valid Values: Any valid IP address.

Default Value: none

2. The *Network Mask* applies to the address specified in Network Prefix to generate an address used in a BGP policy.
   Valid Values: Any valid IP mask.
   Default Value: none
3. The *Address match* is a range of addresses or an exact address.
4. An *Originating AS#* has:
   Valid Values: 0 to 65535
   Default Value: none
5. The *Adjacent AS#* to specifies the neighboring AS number.
   Valid Values: 0 to 65535
   Default Value: none

Example:

```
add receive-policy exclusive

Network Prefix [0.0.0.0]? 10.0.0.0
Network Mask [0.0.0.0]?  255.0.0.0
Address Match (Exact/Range) [Exact]? range
Originating AS# [0]? 168
Adjacent AS# [0]? 165
```

See “AS Based Receive Policy Examples” on page 399 for detailed examples of this policy command.

send-policy *(exclusive/ inclusive) network prefix network mask address match tag adjacent as#*

Use the **add send-policy** command to create policies that determine which of the BGP speaker’s learned routes will be readvertised. These routes could be internal or external to the BGP speaker’s AS.

Exclusive policies prevent route information from being included in the BGP speaker’s routing table.

1. The *Network Prefix* is for the addresses being affected.
   Valid Values: Any valid IP address.
   Default Value: none
2. The *Network Mask* applies to the address specified in Network Prefix to generate an address used in a BGP policy.
   Valid Values: Any valid IP address.
   Default Value: none
3. The *Address match* is a range of addresses or an exact address.
4. A *TAG*. is the value that has been set for a particular AS. Tag values match that of the AS from which they were learned.
   Valid Values: 0 to 65535
   Default Value: none
5. The *Adjacent AS#* specifies the neighboring AS number.
   Valid Values: 0 to 65535
   Default Value: none

Example:

```
add send exclusive

Network Prefix [0.0.0.0]? 180.220.0.0
Network Mask [0.0.0.0]? 255.255.0.0
Address Match (Exact/Range) [Exact]? range
Tag [0]?
Adjacent AS# [0]? 25
```

See “AS Based Send Policy Examples” on page 400 for detailed examples of this policy command.

Attach

Use the **attach policy-to-neighbor** command to attach a configured policy-list name to a specific neighbor. You can attach up to three receive and three send policy-list names.

Syntax:

attach	policy-to-neighbor

Example: attach policy-to-neighbor

```
Neighbor address [0.0.0.0]? 192.0.251.165
First receive policy list name (none for global AS based policy)[]? nbr1-rcv
Second receive policy list name (none for exit)[]?
First send policy list name (none for global AS based policy)[]? nbr1-snd
Second send policy list name (none for exit)[]?
```

Note: See “Neighbor Based Receive Policy Examples” on page 400 and “Neighbor Based Send Policy Examples” on page 400 for detailed examples of this policy command.

Change

Use the **change** command to change a BGP configuration item previously installed by the add command.

Syntax:

change	aggregate . . .
	neighbor . . .
	originate-policy . . .
	policy-to-neighbor
	receive-policy . . .
	send-policy. . .

aggregate *index# network prefix network mask*

This example changes the current aggregate (aggregate 1). The change causes aggregate 1 to use a different network prefix and mask to aggregate all routes in the address range from 128.185.0.0 to 128.185.255.255.

Example:

```
change aggregate 1

Network Prefix [128.185.0.0]? 128.128.0.0
Network Mask [255.255.0.0]? 255.192.0.0
```

neighbor *neighbor IP address as# init timer connect timer hold timer keep alive timer tcp segment size*

The following example changes the value of the hold timer to zero for neighbor 192.0.251.165.

The *neighbor address* to be modified has:
Valid Values: Any valid IP address.
Default Value: none

To reactivate this neighbor dynamically use the **reset neighbor** command from BGP monitoring.

Example:

```
change neighbor 192.0.251.165
```

```
AS [165]?
Init timer [12]?
Connect timer [60]?
Hold timer [12]? 0
TCP segment size [1024]?
```

originate-policy *index# (exclusive/ inclusive) network prefix network mask address match tag*

Use the **change originate-policy** command to alter an existing originate policy definition.

This example alters the BGP speaker's originate policy. Rather than excluding networks with prefix 194.10.16.0 from the IGP routing table, the policy will now include all routes.

Example:

```
change originate-policy

Enter index of originate-policy to be modified [1]?
Policy Type (Inclusive/Exclusive) [Exclusive]? inclusive
Network Prefix [194.10.16.0]? 0.0.0.0
Network Mask [255.255.240.0]? 0.0.0.0
Address Match (Exact/Range) [Range]?
Tag [0]?
```

policy-to-neighbor

Use the **change policy-to-neighbor** command to change a policy-list attachment to a particular neighbor.

Example:

```
change policy-to-neighbor

Neighbor address [0.0.0.0]? 192.0.251.165
First receive policy list name to be changed[nbr1-rcv]?
Second receive policy list name to be changed[]?
Third receive policy list name to be changed[]?
First send policy list name to be changed[nbr1-snd]?
Second send policy list name to be changed[]?
Third send policy list name to be changed[]?
```

receive-policy *index# (exclusive/inclusive) network prefix network mask address match originating as# adjacent as# igpmetric (inclusive only)*

Use the **change receive-policy** command to alter an existing receive policy definition.

This example adds a restriction to the BGP speaker's receive-policy. Rather than import route information from every BGP peer into its IGP routing table, it will now prevent routes from AS 165 from being imported.

Example:

```
change receive-policy

Enter index of receive-policy to be modified [1]?
Policy Type (Inclusive/Exclusive) [Inclusive]? exclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Range]?
Originating AS# [0]?
Adjacent AS# [0]? 165
```

send-policy *index# (exclusive/ inclusive) network prefix network mask address match tag adjacent as#*

Use the **change send-policy** command to alter an existing send policy to one that is more inclusive, or more exclusive.

This example adds a restriction to the BGP speaker's send policy. The restriction ensures that all routes in the address range 194.10.16.0 to 194.10.31.255 will be excluded when advertising to autonomous system 165.

Example:

```
change send-policy

Enter index of send-policy to be modified [1]?
Policy Type (Inclusive/Exclusive) [Inclusive]? exclusive
Network Prefix [0.0.0.0]? 194.10.16.0
```

```
Network Mask [0.0.0.0]? 255.255.240.0
Address Match (Exact/Range) [Range]?
Tag [0]?
Adjacent AS# [0]? 165
```

Delete

Use the **delete** command to delete a BGP configuration item previously installed by the **add** command.

Syntax:

delete aggregate . . .
neighbor . . .
no-receive . . .
originate-policy . . .
policy-list . . .
policy-to-neighbor
receive-policy . . .
send-policy. . .

aggregate *index#*

You must specify the index number of the aggregate you want to delete.

Example: `delete aggregate 1`

neighbor *neighbor IP address*

Use this command to delete a BGP neighbor. You must specify the neighbor's network address.

The *neighbor's network address to be deleted* has:
Valid Values: Any valid IP address.
Default Value: none

To deactivate this neighbor dynamically use the **reset neighbor** command from BGP monitoring.

Example: `delete neighbor 192.0.251.165`

no-receive *as*

Use this command to delete the no-receive policy set up for a particular AS. You must specify the AS number.

The *AS number* has:
Valid Values: 0 to 65535
Default Value: none

Example: `delete no-receive 168`

originate-policy *index#*

Use this command to delete a specific originate policy. You must specify the index number associated with the policy.

Example: `delete originate-policy 2`

policy-list

Use the **delete policy-list** command to delete a policy-list.

Example: `delete policy-list`

```
Name of policy-list to delete []? nbr1-rcv
All policies defined for 'nbr1-rcv' will be deleted.
Are you sure you want to delete (Yes or [No]? Yes
Policy-list 'nbr1-rcv' is deleted.
```

The policy-to-neighbor attachment will be adjusted accordingly.

policy-to-neighbor
Use the **delete policy-to-neighbor** command to delete an existing policy-list name attachment to a particular neighbor.

Example: `delete policy-to-neighbor`

```
Neighbor address [192.0.251.165]?
Remove first receive policy-list name [nbr1-rcv]
Are you sure you want to remove (Yes or [No])? yes
Remove first send policy-list name [nbr1-snd]
Are you sure you want to remove (Yes or [No])? yes
```

receive-policy *index#*
Use this command to delete a specific receive policy. You must specify the index number associated with the policy.

Example: `delete receive-policy`

```
Enter index of receive-policy to be deleted [1]?
```

send-policy *index#*
Use this command to delete a specific send policy. You must specify the index number associated with the policy.

Example: `delete send-policy 4`

Disable

Use the **disable** command to disable a previously enabled BGP neighbor or speaker. Note that neighbors are implicitly enabled whenever added with the **add** command.

Syntax:

disable	BGP speaker
	classless-bgp
	compare-med-from-diff-AS
	neighbor . . .

bgp speaker
Use the **disable bgp speaker** command to disable the BGP protocol.

Example: `disable bgp speaker`

classless-bgp
Use this command to disable a classless route for advertisement.

Example: `disable classless-bgp`

Note: Be sure that the **patch bgp-subnets** command is disabled.

compare-med-from-diff-AS
Use this command to disable a MED comparison between different ASs.

Example: `disable compare-med-from-diff-AS`

neighbor *neighbor IP address*
The *neighbor address* has:
Valid Values: Any valid IP address.
Default Value: none

Example: `disable neighbor 192.0.190.178`

Enable

Use the **enable** command to activate the BGP features, capabilities, and information added to your BGP configuration.

Syntax:

enable	BGP speaker
	classless-bgp
	compare-med-from-diff-AS
	neighbor . . .

bgp speaker *as# tcp segment size*

Use the **enable bgp speaker** command to enable the BGP protocol.

Note: IBM only supports the latest version of BGP - BGP4, which is defined in RFC 1654.

1. The *AS number* is associated with this collection of routers and nodes.
   Valid Values: 0 to 65535
   Default Value: none
2. Enter the *TCP segment size* to specify the maximum segment size that BGP should use for passive TCP connections.
   Valid Values: 0 to 65535 bytes.
   Default Value: 1024 bytes

Example:

```
enable bgp speaker

AS [0]? 165
TCP segment size [1024]?
```

classless-bgp neighbor

Use this command to enable a classless route for advertisement.

Example: `enable classless-bgp`

compare-med-from-diff-AS

Use this command to enable MED comparison between different ASs.

Example: `enable compare-med-from-diff-AS`

neighbor *neighbor IP address*

Use this command to enable a BGP neighbor.

The *neighbor address* has:
Valid Values: Any valid IP address.
Default Value: none

Example: `enable neighbor 192.0.190.178`

List

Use the **list** command to display various pieces of the BGP configuration data, depending on the particular subcommand invoked.

Syntax:

list	aggregate
	all

BGP speaker

neighbor

no-receive

originate-policy

policy-list . . .

policy-to-neighbor

receive-policy

send-policy

aggregate

Use the **list aggregate** command to all aggregated routes defined with the **add aggregate** command.

Example: `list aggregate`

```
Aggregation:
Index    Prefix              Mask
1         194.10.16.0        255.255.240.0
```

all

Use the **list all** command to list the BGP neighbors, policies, aggregated routes, and no-receive-as records in the current BGP configuration.

Example: `list all`

```
                  BGP Protocol:          Enabled
                  AS:                    167
                  TCP-Segment Size:      1024
                  Neighbors and their AS:

                                    Init    Conn    Hold    TCPSEG
Address           State     AS      Timer   Timer   Timer   Size
128.185.250.168   ENABLD    168     12      60      12      1024
192.0.251.165     ENABLD    165     12      60      12      1024

Receive-Policies:
Index  Type  Prefix       Mask      Match  OrgAS  AdjAS IGPmetric
1      INCL  0.0.0.0      0.0.0.0   Range  0      0      0

Send-Policies:
Index  Type  Prefix       Mask      Match  Tag    AdjAS
1      INCL  0.0.0.0      0.0.0.0   Range  0      0

Originate-Policies:
Index  Type  Prefix       Mask             Match  Tag
1      EXCL  194.10.16.0  255.255.240.0    Range  0

Aggregation:

Index  Prefix             Mask
1      194.10.16.0        255.255.240.0
No no-receive-AS records in configuration.
```

bgp speaker

Use the **list bgp speaker** command to derive information on the BGP speaker. The information provided is as follows:

Example:

`list BGP speaker`

```
BGP Protocol:          Enabled
AS:                    165
TCP-Segment Size:      1024
```

neighbor

Use the **list neighbor** command to derive information on BGP neighbors.

Example: `list neighbor`

```
Neighbors and their AS:

                                    Init    Conn    Hold    TCPSEG
Address           State     AS      Timer   Timer   Timer   Size
```

```
128.185.252.168    ENABLD    168    12      60      12      1024
192.0.190.178      DISBLD    178    12      60      12      1024
192.0.251.167      ENABLD    167    12      60      12      1024
```

no-receive

Use the **list no-receive** command to derive information on no-receive-AS definitions that have been added to the BGP configuration.

Example: `list no-receive`

```
AS-PATH with following autonomous systems will be discarded:
AS  178
AS  165
```

originate-policy *all index prefix*

Use the **list originate-policy** command to derive information on the originate policies that have been added to the BGP configuration.

Example: `list originate-policy`

```
Originate-Policies:
Index  Type  Prefix           Mask             Match  Tag
1      EXCL  194.10.16.0      255.255.240.0    Range  0
2      INCL  0.0.0.0          0.0.0.0          Range  0
```

policy-list

Use the **list policy-list** command to list configured policy-list names.

Example: `list policy-list`

```
BGP Config>li policy list
Policy list:
nbr1-rcv  Receive
nbr1-snd  Send
```

policy-to-neighbor

Use the **list policy-to-neighbor** command to list policies attached to neighbors.

Example: `list policy-to-neighbor`

```
Neighbor addrs  receive          send
192.0.251.165   nbr1-rcv         nbr1-snd
```

receive-policy adj-as-number *all* **or** *index* **or** *prefix*

Use the **list receive-policy** command to derive information on the receive policies that have been added to the BGP configuration. You can display all receive policies defined for an AS, or display policies by index or prefix number.

Example: `list receive-policy`

```
Receive-Policies:
Index  Type  Prefix      Mask        Match  OrgAS  AdjAS  IGPmetric
1      EXCL  0.0.0.0     0.0.0.0     Range  178    165
2      INCL  0.0.0.0     0.0.0.0     Range  0      0      0
```

send-policy adj-as-number *all* **or** *index* **or** *prefix*

Use the **list send-policy** command to display information on send policies defined for specified autonomous systems. You can display all send policies defined for an AS, or display policies by index or prefix number.

Example: `list send-policy`

```
Send-Policies:
Index  Type  Prefix      Mask           Match Tag    AdjAS
1      EXCL  194.10.16.0 255.255.240.0  Range   0    165
2      INCL   0.0.0.0    0.0.0.0        Range   0    0
```

Move

Use the **move** command to change the order in which policies and aggregates have been defined. This changes the order in which the router applies existing

policies to route information. Before using this command, it is advisable to use the **list** command to see what policies have been defined.

Syntax:

move	*aggregate* or *originate-policy* or *receive-policy* or *send-policy*

Example:

```
move originate-policy
Enter index of originate-policy to move [1]? 3
Move record AFTER record number [0]?
```

Set

Use the **set** command to set the IP-route-table-scan-timer. The IP-route-table-scan-timer value is used to set the IP forwarding table scanning time interval for BGP updates.

Syntax:

set	ip-route-table-scan-timer

Example:

```
set ip-route-table-scan-timer
```

Update

Use the **update** command and sub-commands to manipulate policies.

Syntax:

update	*policy-list*

Receive Policy Example:

```
update policy-list
Name[]? nbr1-rcv
```

Add

Use the **Add** command to add receive policies within the **update** command.

```
BGP nbr1-rcv: Receive Config>add
Policy type (Inclusive/Exclusive) [Exclusive]? inclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Range]?
Originating AS# [0]?
Any AS# [0]?
MED [0]?
Weight [0]?
Local-Pref [0]?
IGP-metric [0]?
```

Note: There will be no prompting for MED, Local-pref, Weight, and IGP-metric parameters for exclusive receive policy. MED and Local-pref values will be used from received advertisement if they are configured as value ’0’. The value ’0’ for the weight parameter indicates to ignore the weight value in the route selection process.

Change

Use the **Change** command to change policies within the **update** command.

Example:

```
Enter index of receive-policy to be modified [1]?
```

Delete

Use the **delete** command to delete policies within the **update** command.

Example:

```
Enter index of receive-policy to be deleted [1]?
```

Move

Use the **move** command to move policies within the **update** command.

Example:

```
Enter index of receive-policy to move [1]?
Move record after record number [0]?
```

List

Use the **list policy-list** command to list receive policies within the **update** command.

Example: list policy-list

```
Receive policy list for 'name':
                    T Prefix            Match OrgAS AnyAS MED   Weight Lpref IGPmetric
                1   I 0.0.0.0/0         Range 0     0     0     0      0     1
```

Send Policy Example:

```
update policy-list
Name[]? nbr1-rcv
```

Add

Use the **Add** command to add send policies within the **update** command.

```
BGP nbr1-rcv: Send Config>add
Policy type (Inclusive/Exclusive) [Exclusive]? inclusive
Network Prefix [0.0.0.0]?
Network Mask [0.0.0.0]?
Address Match (Exact/Range) [Range]?
Originating AS# [0]?
Any AS# [0]?
TAG [0]
MED [0]?
# of AS to pad[0]?
```

Note: There will be no prompting for MED and ASpad parameters for exclusive send policy. The value 0 for the MED parameter indicates that MED attribute is not included in advertisement. The value 0 for the ASpad parameter indicates that there will be no additional local AS number inserted in the ASpath.

Change

Use the **Change** command to change policies within the **update** command.

Example:

```
Enter index of send-policy to be modified [1]?
```

Delete

Use the **delete** command to delete policies within the **update** command.

Example:

```
Enter index of send-policy to be deleted [1]?
```

Move

Use the **move** command to move policies within the **update** command.

Example:

```
Enter index of send-policy to move [1]?
Move record after record number [0]?
```

List

Use the **list policy-list** command to list send policies within the **update** command.

Example: list policy-list

```
Send policy list for 'name':
                   T Prefix            Match OrgAS AnyAS Tag   MED    ASpad
                1  I 0.0.0.0/0         Range 0     0     0     0      0
```

Accessing the BGP Monitoring Environment

To access the BGP monitoring environment, enter the following command at the + prompt:

```
+ Protocol BGP
BGP>
```

BGP4 Monitoring Commands

This section describes the BGP monitoring commands. These commands allow you to modify the BGP protocol behavior to meet your specific requirements. Some amount of configuration is necessary to produce a fully functional BGP router. Enter BGP monitoring commands at the `BGP>` monitoring prompt.

Table 25. BGP Monitoring Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
Destinations	Displays all entries in the BGP routing table.
Disable neighbor	Disables a particular neighbor or all neighbors.
Dump routing tables	Lists the contents of the IP routing table.
Enable neighbor	Enables a particular neighbor or all neighbors.
Neighbors	Displays currently active neighbors.
Parameter	Displays installed BGP globals in the BGP system.
Paths	Displays all available paths in the database.
Ping	Sends ICMP Echo Requests to another host once a second and watch for a response. This command can be used to isolate trouble in an internetwork environment.
Policy-list	Displays the current installed policy for specific neighbor and usage statics of each policy.
Reset neighbor	Resets a particular neighbor.
Traceroute	Displays the complete path (hop-by-hop) to a particular destination.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Destinations

Use the **destinations** command to dump all BGP routing table entries, or to display information on routes advertised to, or received from, specified BGP neighbor addresses (destinations).

Syntax:

destinations

net address/net address net mask
advertised-to *network address*
received-from *network address*

Example: destination

```
Network/MaskLen    NextHop         MED   Weight LPref AAG AGRAS ORG AS-Path
142.4.0.0/16       192.0.251.165   100   0      0     No  0     IGP seq[165-178]
```

destinations *net address*

Displays detailed information on the specified route or destination network. The command shows how a specific route was learned, the best path to a specific destination, the metric associated with the route, and other information.

Example: `destinations 142.4.0.0`

```
Network/MaskLen   NextHop        MED  Weight LPref     AAG AGRAS ORG ASPath
142.4.0.0/16      192.0.251.165  100  0      0         No  0     IGP seq[165-178]

Dest:142.4.0.0/16, Age:180, Upd#:13,LastSent:0001:53:32

Eligible paths: 2
PathID: 8  (Best Path)
        ASpath: seq[165-178]
        Origin: IGP, Pref: 507, LocalPref: 0
        Metric: 0, Weight: 0, MED: 100
        NextHop: 192.0.251.165, Neighbor: 192.0.251.165
        AtomicAggr: No

PathID: 21
        ASpath: seq[168-165-178]
        Origin: IGP, Pref: 505, LocalPref: 0
        Metric: 0, Weight: 0, MED: 0
        NextHop: 128.185.250.168, Neighbor: 128.185.250.168
        AtomicAggr: No
```

ASpath
Enumeration of autonomous systems along the path.

-seq: Sequence of autonomous systems in order in the path

-set: Set of autonomous systems in the path.

Origin The originator of the destination. This is EGP, IGP, or Incomplete (originated by some other means not known).

LocalPref
The originating router's degree of preference for the destination.

Metric The path metric with which the route is imported.

Weight
The path weight.

MED A multi-exit discriminator value, used to discriminate among multiple entry/exit points to the same AS.

NextHop
The address of the router to use as the forwarding address for destinations reachable via the given path.

AtomicAggr
Indicates whether the router advertising the path has included the path in an atomic-aggregate.

destinations *net address net mask*

Displays detailed information on the specified route or destination network.

The command shows how a specific route was learned, the best path to a specific destination, the metric associated with the route, and other information.

This command is useful in cases where multiple network addresses have the same prefix and different masks. In such cases, specifying the network mask narrows the scope of the information presented.

Example: `destinations 194.10.16.0 255.255.240.0`

```
Dest:194.10.16.0/21, Age:0, Upd#:3, LastSent:0002:00:00

Eligible paths: 1
PathID: 0 - (Best Path)
        ASpath:
        Origin: IGP, Pref: 0, LocalPref: 0
        Metric: 0, Weight: 0, MED: 0
        NextHop: 194.10.16.167, Neighbor: 194.10.16.167
        AtomicAggr: No, Aggregator AS167/194.10.16.167
```

destinations advertised-to *net address*

Lists all routes advertised to the specified BGP neighbor.

Example: `destinations advertised-to`

```
BGP neighbor address [0.0.0.0]? 192.0.251.165

        Destinations advertised to BGP neighbor 192.0.251.165

Network         NextHop         MED  Weight  LPref      AAG AGRAS ORG ASPath
194.10.16.0/20  194.10.16.167   0    0       0          No  167   IGP
192.0.190.0/24  192.0.251.165   0    0       0          No  0     IGP seq [165]
142.4.0.0/16    192.0.251.165   0    0       0          No  0     IGP seq [165-178]
143.116.0.0/16  128.185.250.168 0    0       0          No  0     IGP seq [168]
```

destinations received-from *net address*

Lists all routes received from the specified BGP neighbor.

Example: `destinations received-from`

```
BGP neighbor address [0.0.0.0]? 128.185.250.167

        Destinations obtained from BGP neighbor 128.185.250.167

Network              NextHop         MED Weight  LPref      AAG AGRAS ORG ASPath
194.10.16.0/20       128.185.250.167 0   0       0          No  167   IGP seq[167]
192.0.190.0/24       128.185.250.167 0   0       0          No  0     IGP seq[167-165]
142.4.0.0/16         128.185.250.167 0   0       0          No  0     IGP seq[167-165-178]
```

Disable Neighbor

Use the **disable neighbor** command to disable a particular neighbor or all neighbors that have been enabled. This command brings down the BGP session and removes the routes learned from that neighbor.

Syntax:

disable neighbor *internet address*

Example: **`disable neighbor`**

```
Neighbor address (255.255.255.255 for all) [0.0.0.0]? 128.185.250.167
```

Dump Routing Tables

For a complete explanation of the **dump routing tables** command, refer to “Dump Routing Table” in the “Monitoring IP” chapter of *Protocol Configuration and Monitoring Reference Volume 1*

Enable Neighbor

Use the **enable neighbor** command to enable a particular neighbor or enable all neighbors that have been disabled. This command starts the BGP session with neighbor.

Syntax:

enable neighbor *internet address*

Example:

```
Neighbor address (255.255.255.255 for all) [0.0.0.0]? 128.185.250.167
```

Neighbors

Use the **neighbors** command to display information on all active BGP neighbors.

Syntax:

neighbors *internet address*

Example: **neighbors**

```
IP-Address       Status   State        DAY-HH:MM:SS  BGPID           AS     Upd#

128.185.252.168  ENABLD  Established  00000:48:52  128.185.142.168  168     16
192.0.190.178    ENABLD  Established  00002:01:49  142.4.140.178    178     16
192.0.251.167    DISBLD  Established  00002:01:45  194.10.16.167    167     16
```

IP-Address
Specifies the IP address of the BGP neighbor.

State Specifies the state of the connection. Possible states are:

Connect
Waiting for the TCP connection to the neighbor to be completed.

Active In the event of TCP connection failure, the state is changed to Active, and the attempt to acquire the neighbor continues.

OpenSent
In this state OPEN has been sent, and BGP waits for an OPEN message from the neighbor.

OpenConfirm
In this state a KEEPALIVE has been sent in response to neighbor's OPEN, and waits for a KEEPALIVE/NOTIFICATION from the neighbor.

Established
A BGP connection has been successfully established, and can now start to exchange UPDATE messages.

BGP-ID
Specifies the neighbor's BGP Identification number.

AS Specifies the neighbor's AS number.

Upd# Specifies the sequence number of the last UPDATE message sent to the neighbor.

internet-address
Use the **neighbor** command to display detailed data on a particular BGP neighbor.

Example: **neighbor 192.0.251.167**

```
Active Conn: Sprt:1026  Dprt:179    State: Established KeepAlive/Hold
Time: 4/12
Passve Conn: None
TCP connection errors: 0          TCP state transitions: 0

BGP Messages:     Sent       Received                     Sent
Received
Open:             1          1          Update:           11         11
Notification:     0          0          KeepAlive:        1828       1830
```

```
Total Messages:   1840       1842

Msg Header Errs:  Sent       Received                    Sent
Received
Conn sync err:    0          0          Bad msg length:   0          0
Bad msg type:     0          0

Open Msg Errs:    Sent       Received                    Sent
Received
Unsupp versions:  0          0          Unsupp auth code: 0          0
Bad peer AS ident:0          0          Auth failure:     0          0
Bad BGP ident:    0          0          Bad hold time:    0          0

Update Msg Errs:  Sent       Received                    Sent
Received
Bad attr list:    0          0          AS routing loop:  0          0
Bad wlkn attr:    0          0          Bad NEXT_HOP atr: 0          0
Mssng wlkn attr:  0          0          Optional atr err: 0          0
Attr flags err:   0          0          Bad netwrk field: 0          0
Attr length err:  0          0          Bad AS_PATH attr: 0          0
Bad ORIGIN attr:  0          0

Total Errors:     Sent       Received                    Sent
Received
Msg Header Errs:  0          0          Hold Timer Exprd: 0          0
Open Msg Errs:    0          0          FSM Errs:         0          0
Update Msg Errs:  0          0          Cease:            0          0
```

Parameter

Use the BGP **parameter** command to display installed BGP globals in the BGP system.

Syntax:

parameter

Example:

```
BGP> parameter

classless-bgp is enabled.
compare-med-from-diff-as is enabled.
IP-route-table-scan-timer value is 5 seconds.
```

Paths

Use the BGP **paths** command to display the paths stored in the path description data base.

Syntax:

paths

Example:

```
paths
PathId   NextHop   MED  AAG   AGRAS  RefCnt   ORG   ASPath
0       10.2.0.3    0   No     0      2      IGP
4       192.2.0.2   0   No     0      2      IGP    seq[2]
5       192.2.0.2   0   No     2      1      IGP    seq[2]
6       192.2.0.2   0   No     0      1      IGP    seq[2-1]
7       10.2.0.168  0   No     0      4      IGP
8       192.3.0.1   0   No     0      2      IGP    seq[1]
9       192.2.0.2   0   No     2      1      IGP    seq[2]
10      10.2.0.3    0   No     0      1      IGP
```

PathId
: Path identifier

NextHop
: The address of the router to use as the forwarding address for the destinations that can be reached via the given path.

MED The multi-exit discriminator used to discriminate among multiple entry/exit points to the same AS.

AAG Indicates if the path has been atomic-aggregated that is the router that is advertising the given path has selected less specific route over the more specific one when presented with overlapping routes.

AGRAS
Indicates the AS number of the BGP speaker that aggregated the routes.

RefCnt
Indicates the number of path entities referring to the descriptor.

ORG Specifies the originator of the advertised destinations in the given path: either EGP, IGP, or Incomplete (originated by some other means not known).

AS Path
Enumeration of autonomous systems along the path.

seq: Sequence of autonomous systems in order in the path.

set: Set of autonomous systems in the path.

Ping

For a complete explanation of the **ping** command, see the Ping command in the "Configuring and Monitoring IP" chapter in *Protocol Configuration and Monitoring Reference Volume 1*.

Policy-List

Use the **policy-list** command to display the current installed policy for specific neighbor and usage statistics of each policy.

Example: policy-list

```
Neighbor address[0.0.0.0]? 192.0.251.167
Policy Type(Receive/Send/Origin)[All]?Receive
```

Display for neighbor based policy configuration:

```
Receive policy list for neighbor '192.0.251.167':
Idx T Prefix            Match OrgAS AnyAS MED   Weight LPref IGPmet Usage
1   I 0.0.0.0/0         Range 0     0     0     0      0     1      1
```

Display for AS based policy configuration:

```
Receive policy :
Idx Type  Prefix            Match OrgAS AdjAS IGPmetric Usage
1   INCL  0.0.0.0/0         Range 0     0     1         1
```

Example: policy-list

```
Neighbor address[0.0.0.0]? 192.0.251.167
Policy Type(Receive/Send/Origin)[All]?Send
```

Display for neighbor based policy configuration:

```
send policy list for neighbor '0.0.0.0': 192.0.251.167
Idx T Prefix            Match OrgAS AnyAS TAG   MED   ASpad Usage
1   I 0.0.0.0/0         Range 0     0     0     0     0     1
```

Display for AS based policy configuration

```
send policy :
Idx Type  Prefix            Match OrgAS AdjAS TAG   Usage
1   INCL  0.0.0.0/0         Range 0     0     0     1
```

Example: policy-list

```
Neighbor address[0.0.0.0]? 192.0.251.167
Policy Type(Receive/Send/Origin)[All]?Origin

Origin policy list for neighbor '0.0.0.0':
Idx T Prefix            Match TAG   Usage
1   I 0.0.0.0/0         Range 0     1
```

Reset Neighbor

Use the **reset neighbor** command to reset the specified BGP neighbor, based on the neighbor configuration parameters stored in the configuration memory.

Syntax:

reset neighbor *ip address*

Example: **reset neighbor**

```
Neighbor address[0.0.0.0]? 128.185.250.167
```

Sizes

Use the BGP **sizes** command to display the number of entries stored in the various data bases.

Syntax:

sizes

Example:

sizes

```
# Paths:                                11
# Path descriptors:                      7
  Update sequence#:                     22
# Routing tbl entries (allocated):       6
# Current tbl entries (not imported):    0
# Current tbl entries (imported to IGP): 3
```

Paths Total number of eligible paths for all the routes in the BGP routing table.

Path descriptors
Total number of path descriptors in the database used to hold common path information.

Update sequence#
Indicates the current update sequence number.

Routing tbl entries (allocated)
Indicates the number of entries in BGP routing table.

Current tbl entries (not imported)
Indicates the number of BGP routes not imported into IGP.

Current tbl entries(imported to IGP)
Indicates the number of BGP routes imported into IGP.

Traceroute

For a complete explanation of the **traceroute** command, refer to “Configuring and Monitoring IP” in *Protocol Configuration and Monitoring Reference Volume 2*.

BGP4 Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

Border Gateway Protocol Version 4 (BGP4) supports the CONFIG (Talk 6) **delete interface** command with the following consideration:

Deletes configured BGP external neighbors if the neighbor address has a common net number with an IP address deleted on that interface. This means that the BGP peer relationship is formed using a deleted interface.

GWCON (Talk 5) Activate Interface

The GWCON (Talk 5) **activate interface** command is not applicable for BGP4. BGP has no SRAM records associated with an interface.

GWCON (Talk 5) Reset Interface

The GWCON (Talk 5) **reset interface** command is not applicable for BGP4. BGP has no SRAM records associated with an interface.

GWCON (Talk 5) Component Reset Commands

BGP4 supports the following BGP4-specific GWCON (Talk 5) **reset** commands:

GWCON, Protocol BGP, Reset Neighbor Command

Description:
Adds or deletes a BGP neighbor. Changes neighbor parameters and policies.

Network Effect:
BGP neighbor connection and learned routes are updated based on configuration change.

Limitations:
None.

The following table summarizes the BGP4 configuration changes that are activated when the **GWCON, protocol bgp, reset neighbor** command is invoked:

Commands whose changes are activated by the GWCON, protocol bgp, reset neighbor command
CONFIG, protocol BGP, add neighbor
CONFIG, protocol BGP, change neighbor
CONFIG, protocol BGP, delete neighbor
CONFIG, protocol BGP, attach policy-to-neighbor
CONFIG, protocol BGP, change policy-to-neighbor
CONFIG, protocol BGP, delete policy-to-neighbor
CONFIG, protocol BGP, add policy-list
CONFIG, protocol BGP, update policy-list

GWCON (Talk 5) Temporary Change Commands

BGP4 supports the following GWCON commands that temporarily change the operational state of the device. These changes are lost whenever the device is

reloaded, restarted, or you execute any dynamically reconfigurable command.

Commands
GWCON, protocol BGP, enable neighbor
GWCON, protocol BGP, disable neighbor

Non-Dynamically Reconfigurable Commands

The following table describes the BGP4 configuration commands that cannot be dynamically changed. To activate these commands, you need to reload or restart the device.

Commands
CONFIG, protocol BGP, enable/disable bgp
CONFIG, protocol BGP, add/delete no-receive
CONFIG, protocol BGP, add/change/delete/move aggregate
CONFIG, protocol BGP, add/change/delete/move originate-policy
CONFIG, protocol BGP, add/change/delete/move receive-policy
CONFIG, protocol BGP, add/change/delete/move send-policy
CONFIG, protocol BGP, enable/diable compare-med-from-diff-as
CONFIG, protocol BGP, enable/disable classless-bgp
CONFIG, protocol BGP, set ip-route-table-scan-timer

Chapter 20. Configuring and Monitoring DVMRP

This chapter describes configuring and monitoring for DVMRP (Distance Vector Multicast Routing Protocol) protocol activity. It includes the following sections:

Accessing the DVMRP Configuration Environment

To access the DVMRP configuration environment, enter the following command at the `Config>` prompt:

```
Config> protocol dvmrp
Distance Vector Multicast Routing Protocol config monitoring
DVMRP Config>
```

DVMRP Configuration Commands

This section describes the DVMRP configuration commands. The commands are entered at the `DVMRP Config>` prompt.

Table 26. DVMRP Configuration Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
Add	Adds to already existing DVMRP information. You can add a physical interface or an IP-IP tunnel interface.
Change	Changes DVMRP information in SRAM. You can change the cost or threshold of a physical interface, IP-IP tunnel, the MOSPF interface, or the endpoints of an IP-IP tunnel.
Delete	Deletes DVMRP information from the static configuration.
Disable	Disables the entire DVMRP protocol or the MOSPF interface.
Enable	Enables the entire DVMRP protocol or the MOSPF interface.
List	Displays the DVMRP configuration.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

Add

Use the **add** command to add to existing DVMRP information. You can add a physical interface or an IP-IP tunnel.

Syntax:

add interface *ip-address cost threshold*

tunnel *tunnel-source tunnel-destination cost threshold*

interface

Adds or updates a DVMRP interface

ip-address

Specifies the IP address of the DVMRP interface.

Valid Values: Any valid IP address

Default Value: None

cost Specifies the cost (in terms of hop-count) incurred for using the interface.

Valid Values: Any integer greater than 0

Default Value: 1

threshold
Specifies the time-to-live needed to reach the nearest neighbor on the interface.

Valid Values: Any integer greater than 0

Default Value: 1

tunnel Adds or updates an IP-IP tunnel across a non-multicast network. Tunnels need to be configured when multicast traffic needs to traverse a network which does not support multicast datagrams or are not running a multicast routing protocol.

source-address
Specifies the IP address of the tunnel source.

Valid Values: Any valid IP address

Default Value: None

destination-address
Specifies the IP address of the tunnel destination.

Valid Values: Any valid IP address

Default Value: None

cost Specifies the cost (in terms of hop-count) incurred for using the tunnel.

Valid Values: Any integer greater than 0

Default Value: 1

threshold
Specifies the time-to-live needed to reach the nearest neighbor on the interface.

Valid Values: Any integer greater than 0

Default Value: 1

Change

Use the **change** command to modify existing DVMRP information. You can modify the cost or threshold values of physical interface, IP-IP tunnels, or the MOSPF interface.

Syntax:

change interface *ip-address cost threshold*

tunnel *tunnel-source tunnel-destination cost threshold*

mospf *cost threshold*

interface
Changes a DVMRP interface

ip-address

Valid Values: Any valid IP address

Default Value: None

cost Specifies the cost (in terms of hop-count) incurred for using the interface.

Valid Values: Any integer greater than 0

Default Value: 1

threshold
Specifies the time-to-live needed to reach the nearest neighbor on the interface.

Valid Values: Any integer greater than 0

Default Value: 1

tunnel Changes an IP-IP tunnel.

source-address

Valid Values: Any valid IP address

Default Value: None

destination-address

Valid Values: Any valid IP address

Default Value: None

cost Specifies the cost (in terms of hop-count) incurred for using the interface.

Valid Values: Any integer greater than 0

Default Value: 1

threshold
Specifies the time-to-live needed to reach the nearest neighbor on the interface.

Valid Values: Any integer greater than 0

Default Value: 1

mospf Changes a MOSPF interface.

cost Specifies the cost (in terms of hop-count) incurred for using the interface.

Valid Values: Any integer greater than 0

Default Value: 1

threshold
Specifies the time-to-live needed to reach the nearest neighbor on the interface.

Valid Values: Any integer greater than 0

Default Value: 1

Delete

Use the **delete** command to remove existing DVMRP information from static memory.

Syntax:

delete interface *ip-address*

tunnel *tunnel-source tunnel-destination*

interface
Deletes a DVMRP interface.

ip-address

Valid Values: Any valid IP address

Default Value: None

tunnel Deletes an IP-IP tunnel.

source-address

Valid Values: Any valid IP address

Default Value: None

destination-address

Valid Values: Any valid IP address

Default Value: None

Disable

Use the **disable** command to disable the entire DVMRP protocol or the MOSPF interface.

Syntax:

disable dvmrp

mospf

dvmrp
Disables the DVMRP protocol. When disabled, the device will not participate as a DVMRP multicast router.

mospf Disables the interface to the MOSPF routing protocol. When disabled, the DVMRP protocol will not forward/receive multicast datagrams to/from the MOSPF routing protocol.

Enable

Use the **enable** command to enable the entire DVMRP protocol or the MOSPF interface.

Syntax:

enable dvmrp

mospf *cost threshold*

dvmrp
Enables the DVMRP protocol. All interfaces configured for IP and do not have MOSPF enabled on them, and the MOSPF interface are enabled.

mospf Enables the interface to the MOSPF routing protocol for DVMRP. This interface allows DVMRP to forward multicast datagrams to the MOSPF routing protocol. This interface is treated as a physical interface.

cost Specifies the cost (in terms of hop-count) incurred for using the interface.

Valid Values: Any integer greater than 0

Default Value: 1

threshold
Specifies the time-to-live needed to reach the nearest neighbor on the interface.

Valid Values: Any integer greater than 0

Default Value: 1

List

Use the **list** command to display the current DVMRP configuration. The output displays the current DVMRP state (disabled or enabled), physical interface configuration information, tunnel configuration information, and MOSPF configuration information.

Syntax:

list

Example:

```
DVMRP config> list

          DVMRP on
          phyint 128.185.138.19 1 1
          phyint 128.185.177.19 2 4
          tunnel 128.185.138.19 128.185.138.21 4 4
```

The following information are displayed for each listed interface:

DVMRP protocol
Displays whether DVMRP is enabled or disabled

DVMRP physical interfaces
For each physical interface, its IP address and values for cost and threshold are displayed.

DVMRP tunnel interfaces
For each tunnel interface, the configured tunnel endpoints, cost and threshold are displayed.

DVMRP MOSPF interface
For the MOSPF interface, cost and threshold are displayed.

DVMRP Monitoring Commands

The DVMRP monitoring commands allow you to view the parameters and statistics of networks that have enabled DVMRP.

DVMRP Monitoring Commands (Talk 5)

Enter the DVMRP monitoring commands at the **DVMRP>** prompt.

Table 27. DVMRP Monitoring Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
Dump routing tables	Displays the DVMRP routes contained in the routing table.
Interface summary	Displays DVMRP 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.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Dump Routing Tables

Use the **dump routing tables** command to display the set of known DVMRP multicast sources. Each source is listed together with the DVMRP router it was learned from, an associated cost, and the number of seconds since the routing table entry was refreshed.

Syntax:

dump

Example: **dump**

```
Multicast Routing Table
Type   Origin-Subnet   From-Gateway    Metric  Age  In  Out-Vifs
Direct 18.26.0.0       192.35.82.97      10     30   1    0  2*
Direct 18.58.0.0       192.35.82.97       4     30   1    0  2*
DVMRP  18.85.0.0       192.35.82.97       4     30   1    0  2*
DVMRP  18.180.0.0      192.35.82.97       3     30   1    0  2*
DVMRP  36.8.0.0        192.35.82.97       9     30   1    0  2*
DVMRP  36.56.0.0       192.35.82.97       7     30   1    0  2*
DVMRP  36.103.0.0      192.35.82.97       9     30   1    0  2*
DVMRP  128.61.0.0      192.35.82.97       8     30   1    0  2*
DVMRP  128.89.0.0      192.35.82.97      10     30   1    0  2*
DVMRP  128.109.0.0     192.35.82.97       4     30   1    0  2*
DVMRP  128.119.0.0     192.35.82.97       4     30   1    0  2*
DVMRP  128.150.0.0     192.35.82.97       6     30   1    0  2*
```

Type Displays the type of multicast sources (i.e., DVMRP)

Origin-Subnet
Displays the IP address of the originating subnet.

From-Gateway
Displays the IP address of the gateway from which the entry came.

Metric Displays the associated cost of that route.

Age Displays the age of routing table entry as the number of seconds since the routing table entry was refreshed.

In Displays the DVMRP VIF that multicast datagram from the source must be received on.

Out-Vifs
Displays those VIFs that will send the multicast datagrams. VIFs marked with an asterisk indicate that a datagram will only be forwarded if there are group members on the attached network.

Interface Summary

Use the **interface summary** command to display current list of DVMRP interfaces (or VIFs).

Syntax:

interface *interface-ip-address*

Example: **`interface`**

```
Virtual Interface Table
Vif  Local-Address                          Metric  Thresh  Flags
0   10.1.153.22      subnet: 10.1.153.0       1       1    querier
1   10.1.154.22      subnet: 10.1.154.0       1       1    down
```

Vif Displays the number assigned to DVMRP interfaces (or VIFs). Each VIF is assigned a number, which is used to identify the VIF in other commands.

Local Address
Displays the local IP address of the DVMRP interface.

Metric The associated cost of the route.

Threshold
Reflects the ability of a network to control external flow of multicast packets outside of the network.

Flags Displays whether the VIF is down or that the router is the sender of IGMP Host Membership Queries on the interface.

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 monitoring with two differences:

- The effect on group membership is immediate when the commands are given from the 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 224.185.00.00`**

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 monitoring with two differences:

- The effect on group membership is immediate when the commands are given from the 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 224.185.00.00`

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 DVMRP system going up or down), and on group membership changes.

Note: The numbers displayed in the legend at the top of the output do NOT refer directly to VIFs, but instead refer to physical interfaces (which may be running either DVMRP or MOSPF) and tunnels.

Note:

Syntax:

mcache

Example:

```
mcache
0: Eth/0           1: TKR/0           2: Internal
3: 128.185.246.17  4: 192.35.82.97

Source          Destination     Count   Upst    Downstream
128.185.146.0   239.0.0.1       1       0       2,4
128.119.0.0     224.2.199.198   9       4       3
128.9.160.0     224.2.127.255   1       4       3
13.2.116.0      224.2.0.1       27      4       3
140.173.8.0     224.2.0.1       31      4       3
128.165.114.0   224.2.0.1       25      4       3
132.160.3.0     224.2.158.99    11      4       3
132.160.3.0     224.2.170.143   56      4       3
128.167.254.0   224.2.199.198   27      4       3
129.240.200.0   224.2.0.1       21      4       3
131.188.34.0    224.2.0.1       28      4       3
131.188.34.0    224.2.199.198   28      4       3
```

Source
Source network/subnet of matching datagrams.

Destination
Destination group of matching datagrams.

Count Displays the number of entries processed for that multicast group.

Upstream
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 *none*, the datagram will not be forwarded.

There is more information in a multicast forwarding cache entry. A cache entry can be displayed in detail by providing the source and destination of a

matching datagram on the command line. If a matching cache entry is not found, one is built. A sample of this command is shown below:

Example:

```
mcache 128.185.182.9  224.0.1.2
source Net:     128.185.182.0
Destination:    224.0.1.2
Use Count:      472
Upstream Type:  Transit Net
Upstream ID:    128.185.184.114
Downstream:     128.185.177.11 (TTL = 2)
```

In addition to the information shown in the short form of the mcache command, the following fields are displayed:

Upstream Type
Indicates the type of node from which the datagram must be received to be forwarded. Possible values for this field are “none” (indicating that the datagram will not be forwarded), “router” (indicating that the datagram must be received over a point-to-point connection), “transit network”, “stub network”, and “external” (indicating that the datagram is expected to be received from another Autonomous System).

Downstream
Prints a separate line for each interface or neighbor to which the datagram will be sent. A TTL value is also given, indicating that datagrams forwarded out of or to this interface must have at least the specified TTL value in their IP header. When the router is itself a member of the multicast group, a line specifying *internal application* appears as one of the downstream interfaces/neighbors.

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 an 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.

Syntax:

mstats

Example:

```
mstats

              MOSPF forwarding:       Disabled
              Inter-area forwarding:  Disabled
              DVMRP forwarding:       Enabled
              PIM forwarding:         Disabled

Datagrams received:           10143  Datagrams fwd (multicast):  10219
Datagrams fwd (unicast):          0  Locally delivered:              0
Unreachable source:               0  Unallocated cache entries:      0
Off multicast tree:               0  Unexpected DL multicast:        0
Buffer alloc failure:             0  TTL scoping:                    0
Administrative filtering:       235

# DVMRP routing entries:          5  # 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:            0
```

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 will forward IP multicast datagrams.

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 datalink 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 datalink 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 datalink multicasts on those interfaces that have been configured for datalink 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.

Administrative filtering
Displays the number of datagrams discarded because of outbound filtering.

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 entries 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 entries 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”+.

DVMRP Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

Distance Vector Multicast Routing Protocol (DVMRP) supports the CONFIG (Talk 6) **delete interface** command with no restrictions.

GWCON (Talk 5) Activate Interface

Distance Vector Multicast Routing Protocol (DVMRP) supports the GWCON (Talk 5) **activate interface** command with the following consideration:

DVMRP must be globally enabled before DVMRP can be activated on a network interface.

DVMRP interface-specific commands are supported by the GWCON (Talk 5) **activate interface** command.

GWCON (Talk 5) Reset Interface

DVMRP supports the GWCON (Talk 5) **reset interface** command with the following consideration:

DVMRP must be globally enabled before DVMRP can be activated on a network interface.

All DVMRP interface-specific commands are supported by the GWCON (Talk 5) **reset interface** command.

Non-Dynamically Reconfigurable Commands

All Distance Vector Multicast Routing Protocol (DVMRP) configuration parameters can be changed dynamically.

Chapter 21. Using RSVP

Resource ReSerVation Protocol (RSVP) is an IP signaling protocol used by applications to signal their quality of service (QoS) requirements. RSVP is designed to support sessions from multiple senders to multiple receivers. When RSVP signaling triggers traffic management, the result is dynamic reservation of network resources (for example, bandwidth and buffers) that achieve a desired QoS for packet delivery. RSVP is receiver-oriented, that is, the application that receives the QoS flow is responsible for initiating the RSVP signaling that reserves network resources. Thus, QoS in RSVP is accomplished by establishing reservations along each hop in the path from the receiver to the sender. A reservation consists of a set of parameter values that determine the QoS for a traffic flow. The sender and receiver, which are host applications that are enabled for RSVP, create the reservation by sending RSVP messages to one another. An IBM enhancement allows some non-RSVP enabled applications to have their first-hop router perform RSVP signaling on their behalf. RSVP runs on IPv4 in the IBM routers and supports both unicast and multicast IP traffic. A full description of RSVP is found in RFC 2205.

For each IP traffic flow for which a reservation has been established, RSVP, as implemented on the 2210, provides Controlled Load quality of service. Controlled Load QoS is defined in the Integrated Services model of the Internet Engineering Task Force (IETF) (RFC 2211). Even when the network becomes congested, Controlled Load QoS continues to provide the level of service that the traffic flow receives when the network is not congested.

This chapter includes the following sections:

How RSVP Works

Figure 38 shows the sequence of messages that RSVP uses to establish a reservation that provides QoS to a particular traffic flow. For this example, assume that Best Effort IP traffic flows are already established among the routers.

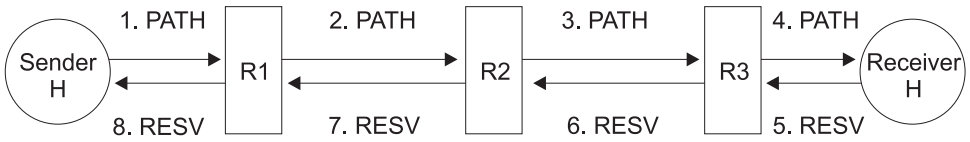


Figure 38. RSVP Reservations-All Routers Support RSVP

The establishment of an RSVP reservation begins when an RSVP-enabled sender sends *PATH* messages towards the receivers of the data traffic flow. The PATH message contains traffic information that describes the flow. When routers receive a PATH message (by looking at the ALERT Option field of the IP header), they establish and maintain a soft state for that PATH message. An RSVP router will also mark the PATH message that it sends on toward the destination with its own IP address, called the previous-hop or p-hop. An RSVP-enabled receiver can respond to one of the PATH messages by sending back a *RESV* message. The RESV message requests network resources such as bandwidth to be reserved along each

link in the path. The RESV message is sent along the reverse path that the PATH message traversed. The RESV message is received by the first router (Router R3) on the reverse path. This router attempts to reserve resources on the outbound interface, that is, on the link between R3 and the receiver host. If the requested resources are available, then they are reserved for this flow, and the amount of available resources is decreased by the corresponding amount. If the requested resources are not available, then the reservation fails at that node and a RESVERR message may flow back to the receiver host. For now, assume that the reservation is successful.

Router R3 sends the RESV message on to the next router (R2) on the path back toward the sender. R2 sets up a reservation on the link between itself and R3 and sends the RESV message on to R1. R1 sets up a reservation on the link between itself and R2 and sends the RESV message to the sender host. In this example, the sender host supports RSVP. It sets up a reservation on the link between itself and R1. A path of reserved links now forms a reservation that has been established from the sender to the receiver.

Now consider a network where not all nodes support RSVP, as shown in Figure 39. In particular, suppose that R2 does not support RSVP. When R2 receives the PATH

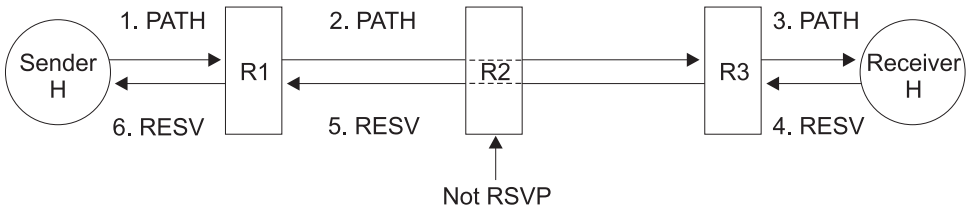


Figure 39. RSVP Reservations-Not All Routers Support RSVP

message, it treats it as an ordinary packet and forwards it toward R3. R2 does not change the p-hop contained in the PATH message

As before, when the PATH message reaches the receiver host, it starts the reservation process by sending a RESV message to R3. The previous-hop that R3 sees on the RESV message is the address of R1 because R2 did not supply its previous hop in the PATH message. R3 sends the RESV message to R1 and makes the reservation on the link between itself and the receiver host. When R1 receives the RESV message from R3, it makes the reservation from itself to R3. Now, reservations (in the direction of the sender) exist in the sender, R1, and R3. Packets will pass through R2 as ordinary best effort packets. In this way, RSVP can be used in a network in which not all routers support RSVP.

Virtual Circuit Resource Manager

Virtual Circuit Resource Manager (VCRM) is a feature that is enabled whenever RSVP is enabled. Based upon the reservation request from RSVP, VCRM creates the connection for the data flow over the physical interface. To do this, VCRM must first determine whether enough bandwidth exists to accommodate the reservation.

Note: If you are using WAN interfaces such as frame relay or X.25, you need to set the line speed so that VCRM knows how much bandwidth is available. The procedure for setting the line speed is described in the frame relay and X.25 interface configuration chapters of the *Software User's Guide*.

If an underlying link in the network supports QoS traffic, such as ATM support of QoS SVCs, VCRM takes advantage of this link capability to establish an ATM SVC to be used by the data traffic for this flow. When the underlying link is not QoS-capable, traffic scheduling and buffering between the network and data link layers aggregate the QoS flows and differentiate them from the best-effort traffic.

If an underlying link is a DiffServ-enabled WAN link, VCRM will request DiffServ to allocate the link resource for the DOS flows, and add the DiffServ TOS markings as traffic flows through the device.

For more information about VCRM, see “Configuring and Monitoring VCRM” in *Using and Configuring Features*.

Traffic Flows and RSVP Sessions

A router’s path and reserve soft state defines the existence of an RSVP reservation and that the traffic flow is being transmitted according to that reservation. An RSVP session consists of all the traffic flows from one or more senders that are being routed over reserved paths to the same IP session address, which can be either a unique or a multicast IP address. For example, in Figure 41 on page 442 the session includes the traffic flows from sender S1 to the receiver Rec 1 as well as the traffic flows from sender S2 to the receiver Rec 1. This session is identified by the IP address of the receiver Rec 1.

The senders and receivers maintain the existence of each path and reservation in the session by sending refresh messages that reaffirm the existence of the reserved traffic flow. These refresh messages are just copies of the PATH and RESV messages. Configurable timers will time out and cause the nodes maintaining soft state to tear down the reservation if the node does not receive a refresh message within a certain amount of time.

There are two types of teardown messages—RSVTEAR and PATHTEAR. The RSVTEAR message, which is sent by the receiver, tears down the reservation but not the traffic flow, which continues with best-effort service. The PATHTEAR message tears down the path from the sender to the session address. PATHTEAR tears down both the reservation and the path soft state. Best effort traffic can still flow.

Reservation Styles

Figure 38 on page 439 shows the establishment of one RSVP reservation, which reserves links for a stream of traffic from one particular sender to one particular receiver. If multiple senders send to the same receiver, multiple IP traffic flows will exist—one from each sender to the receiver. In this situation, the different senders can share reservations over some of the links to the receiver or not, depending upon the selected *reservation style*.

Figure 40 on page 442 shows two senders S1 and S2 for which the receiver has requested the fixed-filter (FF) reservation style. In this reservation style, each sender is provided with its own individual reservation. Host S3 is not participating in RSVP, but is receiving Best Effort traffic.

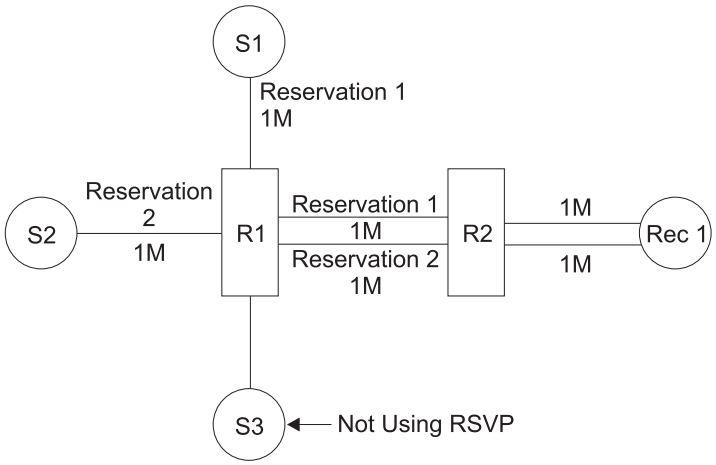


Figure 40. Fixed-Filter Reservation Style

In the shared-explicit (SE) reservation style, the senders identified as members of a particular group can share some of the reserved links. The senders within a group are defined by the receiver according to information that the senders send in the PATH message, such as the senders' IP addresses. In Figure 41, sender S1 and sender S2 have been included in the RSVP session that is identified by the destination address of receiver Rec 1. The senders in the group share the reservation as soon as the senders' paths to the receiver merge. In this case, the common reservation extends from router R1 to the receiver.

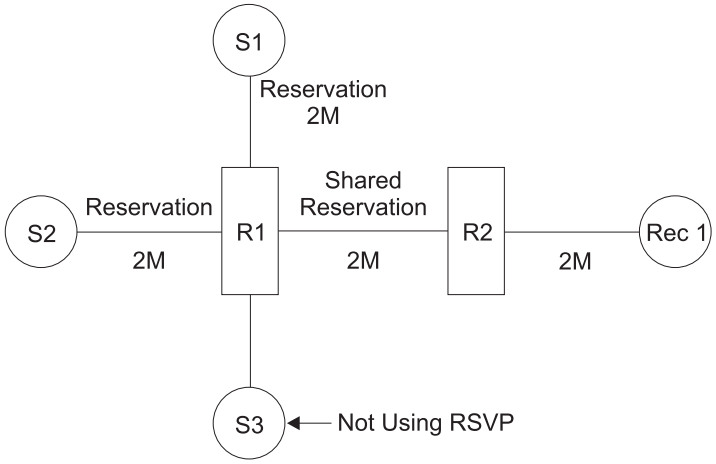


Figure 41. Shared-Explicit Reservation Style

In the third reservation style, wildcard-filter (WF), all senders that send *PATH* messages to the session address share the same reservation, as illustrated in Figure 42 on page 443.

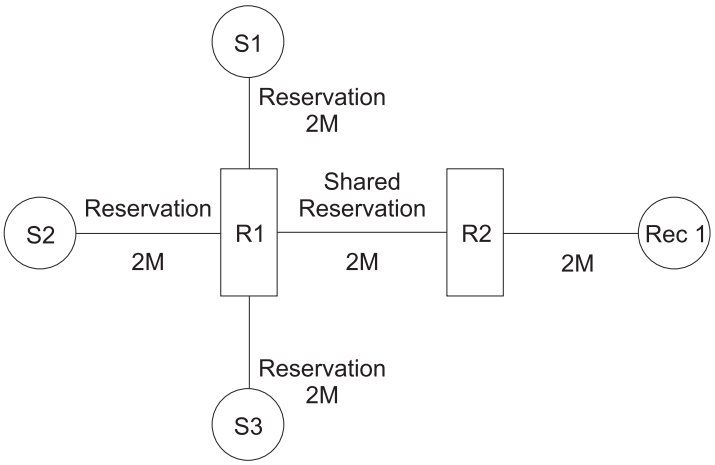


Figure 42. Wildcard-Filter Reservation Style

OPWA

One-Path With Advertising (OPWA) is an optional feature of RSVP. It allows the receiver to obtain a record of the QoS values, such as bandwidth, that are available from each link along the reservation path. For example, if Routers R1 and R3 that are shown in Figure 38 on page 439 are configured for OPWA, they will be told about the characteristics of each link. This information allows them to adjust the information in the PATH message according to the capacity of the link with the least resources.

For example, in the context of Figure 38 on page 439, suppose that a sender starts sending PATH messages toward a receiver with an average rate of 1 Mbps and a peak rate of 10 Mbps. Further suppose that the link between R2 and R3 is a PPP link with a line rate of 2 Mbps. OPWA in R2 will alter the peak rate in the PATH message to be decreased to 2 Mbps, since there is no reason for any nodes downstream to reserve for a peak rate higher than 2 Mbps.

Link Types Supported by RSVP

The link types that RSVP supports include:

- ATM point-to-point SVC.
- PPP links. RSVP supports PPP over all supported link types, such as V.35, T1/E1, and ISDN, that are on a permanent-connection basis. Links that are used in dial-on-demand, WAN-restoral, short-hold mode, or load-balancing configurations should not be used for RSVP.
- Frame relay PVC. As with PPP, all supported link types will support RSVP, but only links that are on a permanent-connection basis should be used for RSVP. Links that are used in dial-on-demand, WAN-restoral, short-hold mode, or load-balancing configurations should not be used with RSVP.
- Frame relay SVC. This is supported in the same way as frame relay PVC; that is, RSVP cannot set up separate DLCIs for QoS traffic, but will use part of the default DLCI for QoS bandwidth allocation.
- All LAN links:
  - Ethernet
  - Token Ring

Note: For shared media networks such as LANs, other methods, such as traffic engineering, are needed to coordinate the sharing of LAN bandwidth. RSVP controls the bandwidth usage of one particular router, but does not coordinate the usage of the LAN bandwidth by multiple routers and hosts.

- X.25. Supported in the same way as PPP or frame relay PVC. RSVP cannot set up separate VCs for QoS traffic; it uses part of the default VC for QoS bandwidth allocation.

Notes:

1. RSVP is routed according to normal IP routing tables. It does not take advantage of Next Hop Routing Protocol (NHRP) in ATM because NHRP uses IP's fast path forwarding cache, not IP routing, to track its routes.
2. To avoid conflict, RSVP is disabled on PPP or FR links that have been configured for Bandwidth Reservation System (BRS).
3. RSVP can use DiffServ queueing and scheduling functions with PPP or FR links that have been configured for DiffServ.

Sample Configuration

As guidance for the configuration of RSVP, a sample of the talk 6 command line interface configuration is included. See "Chapter 22. Configuring and Monitoring RSVP" on page 447 for descriptions of the RSVP commands and parameters. The following steps describe a sample configuration of RSVP:

1. Enable RSVP in the router using the talk 6 **enable rsvp** command from the `RSVP config>` prompt. RSVP can be enabled only on interfaces that are configured for IP. This command sets the RSVP router parameters to default values, including 0 as the default bandwidth on the interfaces. You will need to enable particular interfaces and set bandwidth on them before RSVP can run over those interfaces.
2. Use the **enable interface** command to enable each particular interface for RSVP.
3. Use the talk 5 **reset interface** command if you want RSVP to take effect immediately on this interface.
4. You will be prompted to set the bandwidth for each interface. If the bandwidth for a particular interface is left at 0 (the default), RSVP reservations cannot be made over that interface.
5. Use the command **enable opwa-all** if you want to enable OPWA on all the RSVP-enabled interfaces. Use the command **enable opwa** and the interface number if you want to enable OPWA on one interface. Be sure to enable RSVP over the interface before you enable OPWA. If you attempt to enable OPWA over an interface that has not been enabled for RSVP, you will see the message `Cannot find RSVP i/f rec`.
6. Other parameters are optional and RSVP can run with the default values.
7. If desired, you can use the **add sender** and **add receiver** commands to create static senders or receivers for the router. The static sender and receiver will generate RSVP signaling for a host application that does not use RSVP. The IP address and port configured for the static sender and receiver identify the source and destination of the IP traffic flow for which the router will send RSVP messages. If no static sender or receiver is configured, the router forwards RSVP messages and establishes reservation links, but does not originate RSVP messages. See "Sample Configuration of a Static Sender and Receiver" on page 445 for more information.

Example:

```
Config> protocol rsvp
Resource ReSerVation Protocol config console
RSVP Config> enable rsvp
RSVP Config> enable interface
Interface [0]?
Creating RSVP i/f record...
Set Link Reservable Bandwidth (bits) [0]? 5000000

Interface enabled.
  To take effect immediately, use talk-5 RSVP's 'reset interface'
RSVP Config> enable interface
Interface [0]? 1
Creating RSVP i/f record...
Set Link Reservable Bandwidth (bits) [0]? 1024000

Interface enabled.
    To take effect immediately, use talk-5 RSVP's 'reset interface'
RSVP Config>enable opwa
Interface [0]?
Controlled Load installed on interface 0
take effect immediately?(Yes or [No]): y
RSVP Config>enable opwa
Interface [0]? 1
Controlled Load installed on interface 1
take effect immediately?(Yes or [No]): y
Interface enabled.

RSVP Config>list interface

RSVP Interfaces:

If      IP address   RSVP-enabled  Encaps.  max_res_bw SRAM_rec
0       5.0.31.5        Y          IP       5000000      1
1       5.0.31.3        Y          IP       1024000      2

RSVP Config>list opwa

OPWA configuration:

Network OPWA    CTL-LOAD
0       Y       Y
1       Y       Y
```

After you have completed your configuration, you can activate RSVP by using the talk 5 **reset rsvp** or **reset interface** commands or by restarting the router.

Sample Configuration of a Static Sender and Receiver

If you configure RSVP as described in “Sample Configuration” on page 444, RSVP traffic flows and sessions will be established dynamically by RSVP-enabled applications in hosts that are connected to the router. When there is a host application that is not enabled for RSVP and that sends packets to a known IP address and port, a static sender and receiver can be configured to cause the router to generate RSVP signaling for that flow.

First, configure the sender using the **add sender** command from the `RSVP config>` prompt.

```
Config> protocol rsvp
Resource ReSerVation Protocol config console
RSVP Config> add sender
Session> IP Address: [0.0.0.0]? 5.0.31.1  1
Session> Port Number: [1]? 5004
Session> Protocol Type (UDP/TCP): [UDP]?
Sender> IP Address: [0.0.0.0]? 5.0.27.27  2
Sender> Src Port: [1]? 5005
Tspec> Peak Rate (in byte/sec) [250000]? 25000
Tspec> Average Rate (in byte/sec) [200000]? 20000
Tspec> Burst Size (in bytes) [2000]?
Tspec> Max. Pkt Size [1500]?
Tspec> Min Pkt Size [53]?
```

1 If the traffic flow is unicast, the session IP address is the unicast address of the receiver of the IP traffic flow. If the traffic flow is multicast, the session IP address is the multicast address of the destination of the IP traffic flow.
2 The sender IP address is the unicast address of the sender of the IP traffic flow. If the sender and the receiver are not routers, they are hosts that are connected to routers. The routers in this case act as proxies for the hosts.

After using the **list sender** command to check that the correct values have been configured, you can configure a static receiver in a second remote router that will act as a receiver. In the example, the sending router has the IP address 5.0.27.27 and the receiving router has the IP address 5.0.31.1. To configure the static receiver, use the **add receiver** command.

```
RSVP Config>add receiver
RESV requestor IP Address: [0.0.0.0]? 5.0.31.1
Session> IP Address: [5.0.31.1]?  1
Session> Port Number: [1]? 5004
Session> Protocol Type (UDP/TCP): [UDP]?
Style> (WF, FF, SE): [FF]? wf  2
Need confirmation?(Yes or [No]):
Service Type: CTL-LOAD
Tspec> Peak Rate (in byte/sec) [250000]? 5000
Tspec> Average Rate (in byte/sec) [200000]? 20000
Tspec> Burst Size (in bytes) [2000]?
Tspec> Max. Pkt Size [1500]?
Tspec> Min Pkt Size [53]?
```

1 Note that the IP session address, port, and protocol of the receiver match the IP session address, port, and protocol of the sender. The sender and receiver must identify the same traffic flow. The receiver, not the sender, determines what bandwidth the routers along the path will attempt to establish on each link.
2 The letters *wf* stand for wildcard-filter. This is one of the three reservation styles of RSVP. See “Reservation Styles” on page 441 for more information.

Chapter 22. Configuring and Monitoring RSVP

This chapter describes how to configure and monitor Resource ReSerVation Protocol (RSVP) and how to use the RSVP monitoring commands. It includes the following sections:

Accessing the RSVP Configuration Environment

To access the RSVP configuration environment, enter the following command at the `Config>` prompt:

```
Config> protocol rsvp
Resource ReSerVation Protocol config console
RSVP Config>
```

RSVP Configuration Commands

This section describes the RSVP configuration commands. Enter these commands at the `RSVP Config>` prompt.

Table 28. RSVP Configuration Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
Add	Adds sender and receiver.
Delete	Deletes sender and receiver.
Disable	Disables RSVP or One-Path With Advertising (OPWA).
Enable	Enables RSVP or One-Path With Advertising (OPWA).
List	Lists information about the RSVP configuration.
Set	Sets RSVP system parameters.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

Add

Use the **add** command to add static RSVP senders and receivers to the router. Static senders or receivers enable the router to send or receive RSVP messages. In most cases, if the router sends and receives RSVP messages, it is acting as a proxy on behalf of a host application that is not configured for RSVP. The sender IP address, in such a case, is the address of the host application and the session IP address is the destination address of the data flow. If no static sender or receiver is configured for the router, it dynamically forwards RSVP messages, sets up reservations, and provides QoS, but does not originate RSVP messages.

Definitions of senders and receivers are saved in the configuration as numbered SRAM records. The Talk 5 **activate** command can be used to activate each record.

Syntax:

add	sender ...
	receiver ...

sender
Keyword to specify that the parameters following this term apply to the sender of the RSVP *path* message.

receiver
Keyword to specify that the parameters following this term apply to the receiver, which returns the RSVP *resv* message to the sender.

Most of the following parameters are specified for both the sender and the receiver. Parameters that are unique to the sender or to the receiver are identified in their descriptions.

session-ip-address
This is the unicast or multicast destination IP address of the IP data flows from one or more senders. When the traffic flows are unicast, this address is the receiver's address; when the traffic flows are multicast, this address is a multicast address; the receiver must be a member of the group identified by the multicast address. The senders and the receiver use the session IP address along with the session port number and the protocol to identify the RSVP session for which QoS is established.

Valid values: Valid IPv4 address. Cannot be 0.0.0.0. When RSVP is activated, this address must be accessible to the sender and the receiver.

Default value: none

session-port
The IP port number of the session to be reserved by RSVP. This is the UDP port number or the TCP socket number of the destination application.

Valid values: 0 - 65535

Default value: 1

session-protocol
Either UDP or TCP.

Valid values: UDP or TCP

Default value: UDP

sender-ip-address
The address of the sender, which is the sending application that originates the data flow to be reserved. This parameter must be a unicast address.

Valid values: Valid IPv4 address.

Default value: none

sender-port
The IP port number of the sender of the IP flow to be reserved for QoS. This is the UDP port number or the TCP socket number of the sending application.

Valid values: 0 - 65535

Default value: 1

receiver-ip-address
The IP address of the receiver that issues the *resv* message. In the case of a unicast session, this address is the same as the session IP address. In the case of a multicast session, this address is the unicast address of the

application that makes the reservation for the multicast session address. If it is a multicast session, the receiver must belong to the multicast group represented by this multicast address.

Valid values: Valid IPv4 address.

Default value: none

peak-rate

Specifies the peak data rate on the IP session. This rate is set to the sender's peak traffic generation rate, if known and controlled, the physical interface line rate, if known, or infinity (X'FFFFFFFF', decimal 4 294 967 295) if no better value is available. The peak traffic rate should be set to a value greater than or equal to the average traffic rate.

If the receiver requests a peak data rate different from the rate offered by the sender, the router attempts to honor the receiver's request.

Valid values: 1 - 4 294 967 295 bytes/second

Default value: 250 000

average-rate

Specifies the average data rate that the sender should send or the receiver should receive on the IP session. This rate is set to the sender's average traffic generation rate, if known and controlled, or to the physical interface line rate, if known, or to 200 000 bytes/ second by default.

If the receiver requests a different average rate from that offered by the sender, the router attempts to honor the receiver's request.

Valid values: 1 - 4 294 967 295 bytes/second

Default value: 200 000

data-burst-size

Specifies the number of bytes that can be sent without regard to the peak or average rate. For example, if the peak rate is 50 000 bytes/second, and the data burst size is 2000, 2000 bytes can be sent in one particular instance even if the burst may cause the peak rate to exceed 50 000 bytes/second at that instance.

If the receiver requests a different rate than the sender, the router attempts to honor the receiver's request.

Valid values: 1 - 4 294 967 295 bytes

Default value: 2000

max-packet-size

Specifies the maximum packet size that the sender will send on the IP flow or that the receiver will receive from the IP flow. For the sender, this value should be set to the size of the largest packet generated by the sending application. For the receiver, it should be set to the smallest path MTU, which the receiver learns either from information arriving in RSVP One-Path With Advertisement (OPWA) packets or in other ways.

If the maximum packet size is larger than the MTU of a link on the path, the reservation request will be rejected at that point. For example, if one link along the path of reservations has an MTU of 1500 and the maximum packet size requested is 2000, the reservation request will be rejected.

If the receiver requests a different maximum packet size than the sender, the router attempts to honor the receiver's request.

The maximum packet size should be configured with a value no smaller than the minimum packet size. For example, if the minimum packet size is 64 bytes, the maximum packet size must be equal to or greater than 64 bytes.

Valid values: 1 - 4 294 967 295 bytes

Default value: 1500

min-packet-size

Specifies the minimum packet size that the sender will send on the IP flow or that the receiver will receive from the IP flow. For the sender, this value should be set to the size of the smallest packet generated by the sending application.

This packet size must be no greater than the maximum packet size. For example, if the maximum packet size is 1500 bytes, the minimum packet size must be equal to or less than 1500. This packet size includes the application data and all protocol headers at or above IP level, such as IP, TCP, or UDP, but does not include any link-level headers.

Note: This value is used to estimate the overhead in resource reservation. The smaller the minimum packet size, the larger the reservation overhead.

Valid values: 1 - 4 294 967 295 bytes

Default value: 48

reservation-style

This parameter is configured only for receivers. It specifies the reservation style that the receiver will receive on the IP flow. An RSVP reservation guarantees special handling of the packets in an IP traffic flow to provide a particular QoS over each link or a series of links that form a path from the sender to the receiver. The three reservation styles offered are defined as follows:

Fixed-Filter (FF)

Specifies that the receiver will receive one particular sender's data traffic on the IP flow. One reservation is established per sender.

Shared-Explicit (SE)

Specifies that the receiver will receive data traffic from a group of senders in the same group, which is defined by the receiver. The members of this group share the reservation. Each sender in the group can share the reservation as soon as its link merges into a common path to the receiver.

Wildcard-Filter (WF)

Specifies that the receiver will receive data traffic from all senders. Each sender can share the reservation as soon as its link merges into a common path to the receiver.

See "Reservation Styles" on page 441 for more information.

Valid values: FF, SE, and WF

Default value: FF

confirm-reservation
Specifies whether the receiver wishes to receive a *reservation confirm* message. This message is sent back to the receiver that sent the resv message when the request is merged into an existing larger reservation or is delivered to the sender application.

Valid values: Yes or No

Default value: No

Delete

Use the **delete** command to delete a sender or a receiver.

Syntax:

delete	sender *sram-record*
	receiver *sram-record*

sender or receiver *sram-record*
Each sender or receiver is identified by a SRAM record that is displayed when you use the **delete** command. Entering the SRAM record number of the sender or receiver to be deleted deletes that sender or receiver from the configuration.

Disable

Use the **disable** command to disable RSVP or OPWA on an interface or on all interfaces.

Syntax:

disable	interface
	opwa
	opwa-all
	rsvp

interface *interface-number*
Disables the RSVP function on a particular interface. RSVP control messages can flow over this interface, but no RSVP reservations will be made on this interface. This command also disables the ability of this interface to set up QoS.

Valid Values: Any valid interface number.

Default Value: 0

OPWA *interface-number*
Disables OPWA on a particular interface.

Valid Values: Any valid interface number.

Default Value: 0

OPWA-all
Disables OPWA on all interfaces.

RSVP Disables the RSVP function within the router. By default, RSVP is disabled .

Enable

Use the **enable** command to enable RSVP or OPWA on an interface or on all interfaces.

Syntax:

enable	interface opwa opwa-all rsvp

interface *interface-number*

Enables the RSVP function on a particular interface. This command enables this interface to respond to RSVP messages and to forward them, but not to originate them. You need to configure static senders and receivers to originate RSVP messages.

You will be prompted to set the bandwidth on the enabled interface. You can also use the **set bandwidth** command later to change the bandwidth setting. This command operates only if the router is enabled for RSVP and the specified interface is enabled and configured for IP.

See “Link Types Supported by RSVP” on page 443 for a list of the links that support RSVP.

Valid Values: Any valid interface number.

Default Value: 0

OPWA *interface-number*

Enables OPWA on a particular interface. OPWA tells the receiver whether the path between the sender and the receiver can be reserved on every hop and how much bandwidth is available at each hop along the path. This operation is allowed only if the interface is enabled for RSVP.

Valid Values: Any valid interface number.

Default Value: 0

OPWA-all

Enables OPWA on all interfaces. RSVP must be enabled in the router for this command to take effect.

RSVP Enables the RSVP function within the router. If this is the first time that RSVP is enabled, a set of default parameters for RSVP will also be initialized.

Enabling RSVP does not activate it. To activate RSVP in this router, you have to use the **set bandwidth** command to set bandwidth on at least one interface that will use RSVP. Then, you have to restart the router for RSVP. To do this, you can use the Talk 5 command **reset rsvp** or reboot the router. See the Talk 5 **reset rsvp** command for more information.

List

Use the **list** command to list RSVP parameters. These groups of parameters can be separately listed:

- All parameters
- Interface parameters
- OPWA settings for all interfaces
- Sender or receiver records
- System-level RSVP parameters

Note: The **list** command lists the sender and receiver records that have been configured. These records do not identify the active RSVP traffic flows, which

are defined by the address of the sender and the address of the receiver. Use the Talk 5 **show rsvp flows** command to see the RSVP flows that are currently active.

Syntax:

list ...
- all
- interface
- opwa
- receiver
- sender
- system

Example:

```
RSVP Config>list all

Software Version:

RSVP Control: IBM RSVP Router Release 1.0 (RFC 2205)


RSVP Configuration:

RSVP Status:                    Enabled
Maximum RSVP Msg Size:          1500 (bytes)
Refresh Interval:               30 (sec)
Allowed Successive Msg Loss:    3 (frame)
Flow Life-Time:                 158 (sec)
Refresh Slew Max:               30 (percent)
Total system reservable b/w:    4294967 (kbps)

RSVP Interfaces:

If      IP address   RSVP-enabled  Encaps.  max_res_bw SRAM_rec
0       5.0.27.2        Y          IP       5000000      1
5       5.0.28.2        Y          IP       8000000      2
4       5.0.25.101      Y          IP       1024000      3
2       5.0.45.2        Y          IP       1024000      4

OPWA configuration:

Network OPWA    CTL-LOAD
0       Y       Y
5       Y       Y
4       Y       Y
2       Y       Y

Following senders/receivers are defined in SRAM:
Rec.No   Type     DestAddr [1]      Dest Port   Protocol   Src Addr        Src Port

1    Sender(PATH) 5.0.25.100      25         17         5.0.25.101      25
2    Receiv(RESV) 5.0.25.101      26         17         0.0.0.0         0
```

[1] The destination address displayed is the IP session address. See the Talk 6 **add session-ip-address** command for the definition of the IP session address.

Set

Sets the RSVP system parameters. See the example under the Talk 6 **list all** command for a view of some typical values for these parameters.

Syntax:

set ...
- allowed-successive-msg-loss ...
- bandwidth ...
- default
- encapsulation ...
- lifetime ...
- max-msg-size ...
- refresh-interval ...

slew ...
total ...

allowed-successive-msg-loss *msg-losses*

This parameter defines the number of successive path and matching resv refresh messages that can be lost before RSVP times out the path and reserve state that is defined for the RSVP traffic flow. When RSVP times out the path and reserve state for a particular traffic flow, that flow no longer provides QoS. The sender and receiver have to re-establish the reservation.

Valid Values: 1 - 9999

Default Value: 3

bandwidth *interface bps*

This parameter defines the reservable bandwidth of an interface. Normally the reservable bandwidth should be a small portion of the total link bandwidth. A good target is no more than 30%. The reservable bandwidth can be set only on an interface that is enabled for RSVP.

This Talk 6 command can optionally take effect immediately and dynamically without affecting the values of other parameters.

interface

Network interface number.

Valid Values: Any valid network interface number.

Default Value: 0

bps Bps of bandwidth that can be reserved on this interface.

Valid Values: 1 - 4 294 967 295 bps (represents unlimited)

Default Value: 0

default

This parameter sets all RSVP parameters to the original defaults that exist when you use the command **enable rsvp**. The **set default** command overwrites any parameter values that you have previously configured on the individual interfaces. Because the default value for bandwidth on each interface is 0, meaning that RSVP reservations will not be established on that interface, you have to use the **set bandwidth** command for each interface that uses RSVP to prepare RSVP to run again.

encapsulation *interface style*

This parameter sets the RSVP message encapsulation style on an interface to IP, UDP, or both. Normally, the RSVP control messages, such as path and resv messages, are encapsulated in native IP frames with protocol type 46. In case a host that is connected to this router can use only UDP packets to send the RSVP messages, the encapsulation style over the interface that connects to that host should be set to UDP. If some hosts that use IP and some that use UDP are sending RSVP messages over the same link, then you should set the encapsulation style to Both. This operation is permitted only if RSVP is enabled on the specified interface.

This Talk 6 command can optionally take effect immediately and dynamically without affecting the values of other parameters.

interface

Network interface number.

Valid Values: Any valid network interface number.

Default Value: 0

style Encapsulation style of the RSVP messages.

Valid Values: IP, UDP, or both

Default Value: IP

lifetime

This parameter defines the lifetime in seconds of a path and reserve state, which maintains an established RSVP traffic flow. This time must be long enough for RSVP to observe the number of refresh message losses that is specified by the value of the allowed successive message loss parameter. To roughly calculate this time, use this formula: 1.5 x refresh-interval x (allowed-successive-msg-losses + 0.5).

If the reserve state times out, but not the path state, the reservation is torn down and the IP traffic flow continues with best effort service. If the path state times out, both the reservation and the IP traffic flow are ended.

This Talk 6 command can optionally take effect immediately and dynamically without affecting the values of other parameters. It is expected that the default value for this parameter should work without modification.

Valid Values: 1 - 2 147 483 647 seconds

Default Value: 158 seconds

max-msg-size

This parameter defines the overall maximum RSVP control message size in the router. This value must be no greater than the smallest of MTU sizes that are supported by the RSVP-enabled interfaces along the path. It is expected that the default value for this parameter should work without modification.

Valid Values:64 - 2 147 483 647 bytes (represents unlimited)

Default Value: 1500 bytes

refresh-interval

This parameter defines the time interval in seconds that elapses between refresh messages to maintain a path and reserve state (an RSVP traffic flow) between the receiver and the sender.

Valid Values: 10 - 600 seconds

Default Value: 30 seconds

slew-max

This parameter limits how much the refresh interval can be changed within one refresh cycle. It is expected that the default value for this parameter should work without modification. However, you may need to modify the value of this parameter to prevent timing errors.

For example, if the slew-max is 30% and the refresh interval is 30 seconds, you can change the refresh interval a maximum of 9 seconds (30% of 30) within one refresh interval. To make a larger change, you must change the refresh interval a second time. For example, once the refresh interval is 39, you can change it plus or minus 11 within one refresh interval. Alternatively, you can increase the slew-max and then make the change. For example, if the refresh interval is 30 and you want to change it to 50, you can first increase the slew-max to 70% (giving you the ability to change 30 by plus or minus 21) and then increase the refresh interval to 50.

This Talk 6 command can optionally take effect immediately and dynamically without affecting the values of other parameters.

Valid Values: 0 - 100%

Default Value: 30%

total Because the aggregate of link bandwidths of all the interfaces can be larger than the total router throughput, you may need to set a limit on the router's total reservable bandwidth. For example, the sum of the bandwidth of the aggregate links might add up to 250 000 000 bps, while the total router throughput might be 200 000 000 bps. If the total reservable bandwidth is set to 200 000 000 bps and 200 000 000 bps are currently reserved across all the interfaces, no more RSVP IP reservations can be established until some are torn down.

This Talk 6 command can optionally take effect immediately and dynamically without affecting the values of other parameters.

Valid Values: 1 to 4 294 967 295 bps

Default Value: 4 294 967 295 bps (represents unlimited)

Accessing the RSVP Monitoring Environment

To access the RSVP monitoring environment type **t 5** at the OPCON prompt (*):

```
* t 5
```

Then, enter the following command at the **+** prompt:

```
+ protocol rsvp
RSVP>
```

RSVP Monitoring Commands

This section describes the RSVP monitoring commands. Enter these commands at the `RSVP>` prompt.

Table 29. RSVP Monitoring Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
Activate	Activates a statically defined sender or receiver.
List	Lists RSVP information.
Reset	Dynamically resets RSVP and characteristics of RSVP.
Send	Sends various RSVP messages, including *data-packet, ip ping, path, ptear, resv,* and *rtear*.
Show	Shows information about the active RSVP flows.
Stop-RSVP	Stops the RSVP function in the router.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

Activate

Use the **activate** command to dynamically activate a configured sender or receiver.

Syntax:

activate *record-number*

This command enables you to dynamically activate senders or receivers that you have defined using the Talk 6 **add sender** and **add receiver** commands and that have been enabled using the appropriate Talk 6 **enable** commands.

record-number

When you use the **activate** command, the currently enabled and configured senders and receivers will be displayed and each will be identified with a record number. When you specify a record number, that receiver or sender will be dynamically activated. An activated sender or receiver can be stopped in Talk 5 by issuing a **send ptear**, **send rtear**, or **reset rsvp** command, or by restarting the router.

To learn how to configure static senders and receivers, see "RSVP Configuration Commands" on page 447 for a description of the Talk 6 **add sender**, **add receiver**, and **enable** commands.

List

Use the **list** command to display information about the running RSVP configuration.

Note: Use the Talk 5 **show rsvp flow** command to see existing RSVP traffic flows.

Syntax:

list	interface opwa sender/receiver-records-in-sram system

interface

This command shows RSVP interfaces and their current status. The state *bwCtrl* designates a link that is under RSVP bandwidth control; bandwidth can be reserved on this interface for RSVP QoS. The state *notCnf* indicates a link that is not configured for RSVP. The state *up* indicates that a link is configured fro RSVP, but the bandwidth is under the control of a link-level QoS function (such as the Differentiated Services feature).

Example:

```
RSVP> list int

RSVP Interfaces:

If      IP address      b/w(K)  res'able  curr-res     state
0/Eth   5.0.27.2        10000   5000      0      Kbps  bwCtrl
2/PPP   5.0.45.2        0       1024      0      Kbps  notCnf
4/PPP   5.0.25.101      2048    1024      0      Kbps  up
 5/TKR   5.0.28.2        16000   8000      0      Kbps  bwCtrl
```

opwa This command shows RSVP interfaces and their current OPWA status.

Example:

```
RSVP>list opwa

OPWA running configuration
Network OPWA    CTL-LOAD
0       Y       Y
2       Y       Y
4       Y       Y
5       Y       Y
```

sender/receiver-records-in-sram

This command shows the list of senders and receivers that have been statically configured.

Example:

```
RSVP> list sender

Following senders/receivers are defined in SRAM:
Rec.No   Type     DestAddr       Dest Port   Protocol   Src Addr        Src Port
1    Sender(PATH) 5.0.25.100      25         17         5.0.25.101      25
2    Receiv(RESV) 5.0.25.101      26         17         0.0.0.0         0
3    Receiv(RESV) 5.0.25.101      5006       17         0.0.0.0         0
```

system

This command shows the currently running values of the RSVP system parameters, which will be different from those in SRAM if any of them have been dynamically altered using the Talk 5 commands.

Example:

```
RSVP> list system

RSVP running configuration:
RSVP Status:                    Running
Current Existing Flows:         0
Current Existing Sessions:      0
Maximum RSVP Msg Size:          1500 (bytes)
Refresh Interval:               30 (sec)
Allowed Successive Msg Loss:    3 (frame)
Flow Life-Time:                 158 (sec)
Refresh Slew Max:               30 (percent)
System resv Max:                unlimited
System current resv:            0 (kbps)
```

Reset

Use the **reset** command to reset various aspects of the RSVP configuration. The **reset** command overwrites any parameters that were dynamically configured using Talk 5 and substitutes the values that were most recently configured using Talk 6.

Syntax:

reset	interface
	queue-stat
	rsvp
	system-parameters

interface

Updates the RSVP interface parameters with the configuration data that is stored in SRAM. The command prompts you for the interface number.

The reservations over this interface will be lost and re-established at the next path and resv refresh time, subject to resource availability. There is a risk that some reservations may be lost if the resources to renew them, such as bandwidth, are no longer available.

queue-stat

Clears the flow-control queues at all the interfaces that are configured for RSVP.

rsvp Stops RSVP on the router and restarts RSVP if it is enabled in SRAM.

All path and resv messages on the router will be cleaned up when RSVP is stopped. When RSVP is restarted, the reservations will be restarted at the next path and resv refresh time, subject to resource availability. There is a risk that some reservations may be lost if the resources to renew them, such as bandwidth, are no longer available.

system-parameters

Updates the RSVP system parameters with the configuration data that was created in Talk 6 and is stored in SRAM. The RSVP system parameters are those that are set using the Talk 6 **set** command.

Send

Use the **send** command to dynamically send IP ping and RSVP messages.

Syntax:

send data-packet
ip-ping
path
ptear
resv
rtear

data-packet

This is a command to send test data over a defined IP flow. It can send multiple packets per second, subject to the speed of the router and resource limitations. A message is displayed every time the tenth packet is sent.

Example:

```
RSVP>send data
IP Dest Address: [0.0.0.0]? 5.0.25.100
Destination UDP port: [1]? 100
IP Srce Address: [5.0.25.101]?  1
Source UDP port: [1]? 100
Number of pings per second: [1]?
UDP packet length: [56]?
RSVP send data 1 to 5.0.25.100 protocol 17 source port 100 dest port 100.
..........RSVP send data 11 to 5.0.25.100 protocol 17 source port 100 dest port
100.
..........RSVP send data 21 to 5.0.25.100 protocol 17 source port 100 dest port
100.
RSVP>
```

1 This is the IP address of the router that sends this IP flow.

ip-ping

Sends an IP ping (ICMP echo) message. See the **ping** command in the chapter “Configuring and Monitoring IP” in the *Protocol Configuration and Monitoring Reference Volume 1*.

path Sends an RSVP *path* message, either for itself or as a proxy for another host. The input format for this command is the same as for the Talk 6 **add sender** command. See the Talk 6 **add sender** command for descriptions of the parameters required.

By default, these messages are sent every 30 seconds. The path remains in existence until you remove it with the **send ptear** command or reset RSVP.

This command can dynamically add a sender to the configuration. You can use Talk 2 to view the ELS trace of the path refreshes.

ptear Sends an RSVP *ptear* message, either for itself or as a proxy for another host. Tearing down a path using the **send ptear** command removes both the traffic flow and the reservation. It prompts you for the parameters that identify a path, for example, the IP destination address and the IP session address. See the Talk 6 **add** command for a description of the requested parameters.

The path state specified in the **send ptear** command must exist or an ELS error message is generated. You can use Talk 2 to view the ELS messages associated with this command.

resv Sends an RSVP *resv* message, either for itself or as a proxy for another host. It prompts you for the parameters that identify a path, such as the IP destination address and the IP session address. See the Talk 6 **add**

command for a description of the requested parameters. You can use Talk 2 to view the ELS messages associated with this command. To view these trace messages, you have to enable them using these commands from either the Talk 6 or the Talk 5 prompt:

Example:

```
Config>event
ELS config>disp sub rsvp all
```

If you attempt this command for a receiver that has not set up an RSVP session, this command displays the message `Inputting session does not exist`. Use the **show rsvp flow** command for a display of the existing RSVP flows.

Example:

```
RSVP>send resv
RESV requestor IP Address: [0.0.0.0]? 5.0.25.101
Session > IP Address: [5.0.25.101]?
Session > Port Number: [1]? 201
Session> Protocol Type (UDP/TCP): [UDP]?
Inputting session does not exist.
RSVP>
RSVP>show rsvp flow

Number of flows:        1

Num To (Session)    From            Prot DPrt  SPrt In-If Out-If Rsvd Nhop's
--------------------------------------------------------------------------------
1   5.0.25.101      5.0.25.100      UDP  26    26    4     6      N    0
RSVP>
RSVP>send resv
RESV requestor IP Address: [0.0.0.0]? 5.0.25.101  1
Session > IP Address: [5.0.25.101]?  2
Session > Port Number: [1]? 26
Session> Protocol Type (UDP/TCP): [UDP]?
Style> (WF, FF, SE): [FF]?
Need confirmation?(Yes or [No]):
Service Type: CTL-LOAD
Tspec> Peak Rate (in byte/sec) [250000]? 25000
Tspec> Average Rate (in byte/sec) [200000]? 20000
Tspec> Burst Size (in bytes) [2000]?
Tspec> Max. Pkt Size [1500]?
Tspec> Min Pkt Size [53]?

Existing Filters:
Filter 1 (sender-address : sender-port):  5.0.25.100:26

Make reservation to all senders?(Yes or [No]): Y
A new RESV message will be sent from 5.0.25.101:26 to 5.0.25.100:26
RESV message sent
RSVP>
RSVP>sh r flow

Number of flows:        1

Num To (Session)    From            Prot DPrt  SPrt In-If Out-If Rsvd Nhop's
--------------------------------------------------------------------------------
1   5.0.25.101      5.0.25.100      UDP  26    26    4     6      Y  3   0
RSVP>

*t 2  4
43:56:28 RSVP.074: Send RESV refresh for session 5.0.25.101:26
43:56:28 RSVP.073: --RSVP send IP pkt to 5.0.25.100 on net 4, return code=0
```

1 The requestor's address must be an IP unicast address.

2 The IP session address, which is the destination address for the session, can be either the IP unicast address of the receiver or an IP multicast address of a multicast group of which the receiver is a member.

3 Notice that the *Rsvd* (Reserved) field of the flow entry changes from N (No) to Y (Yes) after the reservation is made. If this value is N, a flow exists, but there is no reservation. The flow is being sent using best effort QoS.

4 The Talk 2 ELS trace shows the reserve refreshes being sent by default every 30 seconds.

rtear Sends an RSVP *rsvtear* message, either for itself or as a proxy for another host. This command disconnects an RSVP traffic flow, but does not tear down the path from the sender, so the IP traffic flow continues with best effort QoS. The command prompts you for the parameters that identify an RSVP traffic flow, for example, the IP receiver's destination address and the IP session address. See the Talk 6 **add** command for a description of the requested parameters.

The IP traffic flow specified in the **send rtear** command must exist or an ELS error message is generated. You can use Talk 2 to view the ELS messages associated with this command.

Show

Use the **show** command to show various aspects of RSVP.

Syntax:

```
show        adspec
            classifier
            ds
            flowspec
            queue
            rsvp
                    flows
                    senders
                    sessions
                    reservations
                    requests
            vc
```

adspec
Shows the advertisement spec (adspec) of all flows. Adspec is the output of OPWA; it lists information about the resources reserved at every link along an active RSVP session path.

classifier
Shows all the current QoS flow entries in the RSVP packet classifier and/or the IP cache.

ds Shows the current reservations over Differentiated Services (DS) links. The streamID field allows the user to correlate the reservations with those displayed by the DS feature **show stream** command.

flowspec
Shows the sender tspec, reservation tspec, and request tspec that are currently in the RSVP state tables.

queue Shows the current statistics about RSVP's software queues. This applies only for non-ATM links.

rsvp Shows aspects of the current RSVP connection status.

flows Shows the active RSVP traffic flows. See the example in the Talk 5 **send resv** command for an example of this command.

senders
Shows the RSVP senders. Senders are configured, but are not necessarily activated.

sessions
Shows the RSVP sessions, both active sessions that have reserved flows and inactive ones that exist but have no reservations at present.

reservations
Shows the RSVP reservations.

requests
Shows the RSVP requests.

vc Shows the currently established ATM SVCs that are reserved by RSVP.

Stop-RSVP

Use the **stop-rsvp** command to stop the RSVP function in the router.

Syntax:

stop rsvp

Chapter 23. Using SNMP

This chapter describes SNMP. It contains the following sections:

- “Network Management”
- “SNMP Management”

Network Management

Refer to the *IBM 2210 Introduction and Planning Guide* for information about Network Management.

SNMP Management

The IBM 2210 provides a Simple Network Management Protocol (SNMP) interface to network management platforms and applications, such as the Nways Campus Manager products.

SNMP is used for monitoring and managing IP hosts in an IP network and uses software called an SNMP agent to enable network hosts to read and modify some of the IBM 2210’s operational parameters. In this way, SNMP establishes network management for the IP community.

You need to consider the following aspects of SNMP when you configure SNMP for your IBM 2210.

Community
The community allows you to define the IP address of the SNMP management station that is allowed to access the information in the SNMP agent’s Management Information Base (MIB). You define a community name for use in accessing the MIB.

Authentication
The community name is used as an authentication scheme to prevent unauthorized users from learning information about an SNMP agent or modifying its characteristics.

This scheme involves defining one or more sets of MIB data (referred to as MIB views) and associating an access privilege (read-only, read-write), an IP mask, and a community name with each MIB view. The IP mask establishes which IP addresses can originate access requests for a given MIB view and the community name serves as a password that must be matched by the SNMP requests. The community name is included in each SNMP message and verified by the IBM 2210 SNMP agent. An SNMP request will be rejected if it does not provide the correct community name, does not match the IP mask, or attempts an access that is inconsistent with the assigned access privilege.

SNMP Password
The SNMP password is used to encrypt and authenticate security sensitive MIB objects such as password or encryption key in the user profile section of Authentication Feature. Setting the SNMP password to a string of zero length indicates that security sensitive data is not accessible. When the SNMP password is set to *clear*, the data is SNMP-accessible without encryption. When the SNMP password is set to other strings, the data is retrievable with encryption and authentication using a key derived from the SNMP password. For further information, refer to the MIB definition.

MIB Support

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.

IBM 2210 provides a comprehensive set of standard MIBs, enterprise-specific MIBs for monitoring and managing resources, and Readme files.

You can find the Readme files documenting IBM 2210 MIB support by accessing the appropriate release directory on the World Wide Web at:

ftp://ftp.nways.raleigh.ibm.com/pub/netmgmt/2210/

To receive a copy of a specific MIB, enter the **get** command with the name of the MIB. For example, the command **get ibm.mib** places a copy of the specified MIB in the directory from which you connected to the FTP server.

You can access the following information from the ftp site:

- Standard MIBs
- Enterprise MIBs
- SNMP generic traps
- Enterprise-specific MIBs
- Settable values

SNMP generic traps, Enterprise MIBs, and settable values are located in the Readme files.

All MIB objects are implemented as READ-ONLY objects even if their access clause is defined as read-write or read-create, except those MIB objects identified in the Readme file that support SETs for objects that have their access clause defined as read-write or read-create.

Trap Messages

Trap messages are unsolicited messages sent from the SNMP agent in the device to an SNMP manager in response to a device or network condition, such as a device reload or network down.

Chapter 24. Configuring and Monitoring SNMP

This chapter describes the SNMP configuring and monitoring commands. It includes the following sections:

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 describes the SNMP configuration commands.

Table 30 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 2210.

Enter the SNMP configuration commands at the `SNMP Config>` prompt.

Table 30. SNMP Configuration Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
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.
Disable	Disables SNMP protocol and traps associated with named communities.
Enable	Enables 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 This command is also used to set a trap UDP port and to set the password used to encrypt and authenticate security-sensitive data.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

SNMP Configuration Commands (Talk 6)

Table 31. 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>		
		warm_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>		
		warm_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#>			
	password				
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: 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.

community name

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: none

Example:

```
SNMP Config>  add community
Community Name  []? comm01
Community added successfully
```

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 also specify hosts that receive the traps. If no address is specified, no trap is generated.

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: none

IP address

Valid Values: Any valid IP address.

Default Value: 0.0.0.0

ip mask

You also may supply a 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: 255.255.255.255

Example:

```
SNMP Config> add address
Community Name []?
IP Address [0.0.0.0]?
IP Mask [255.255.255.255]?
```

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>.

view name

Specifies the name of the view to be created.

Valid Values: Any alphanumeric character string up to 31 characters in length. Characters such as spaces, tabs, or <Esc> key sequences are not accepted.

Default Value: none

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.

If a community is added using the **add community** command, all supported MIB views are assigned to the community unless the **set community view** command is used to assign specific views to the community.

MIB OID name

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 1 element. Since all MIB OIDs begin with *1.3.6.1*, the minimum number of elements that you need to be provide in order for the view to differ from *all* is 5 (1.3.6.1.X).
- You can define a maximum of 31 characters, including the . separators.
- All elements after the first four (*1.3.6.1*) are integers between 0 and 127.

Note: This value must be numeric in dotted notation, *not* a symbolic value.

Default Value: none

Example:

```
SNMP Config> add sub_tree
View Name  []? view01
MIB OID name []? 1.3.6.1.1
Subtree added successfully
```

Delete

Use the **delete** command to delete a community and all of its addresses, a specific address, or a subtree from a view.

Syntax:

delete	community
	address
	sub_tree

community

Removes a community and its IP addresses.

community name

Specifies 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.

Valid Values: A string of 1 to 31 alphanumeric characters. Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: none

Example:

```
SNMP Config> delete community
Community Name []?
```

address

Removes an address from a community. You must supply the name.

community name

Specifies the name of the community from which an address is to be removed. 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

IP address

Specifies the IP address to be removed.

Valid Values: Any valid IP address.

Default Value: none

Example:

```
SNMP Config>  delete address
Community Name []?
IP address []?
```

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.

view name

Specifies 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 device's SNMP agent.

This parameter should be answered if you decide to restrict a community from accessing the entire MIB managed by the device's SNMP agent.

Default Value: none

MIB OID name

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.

Valid Values: An object identifier in the form of <element1>.<element2>.<element3>. . ., where:

- You need a minimum of 1 element. Since all MIB OIDs begin with *1.3.6.1*, the minimum number of elements that you need to be provide in order for the view to differ from *all* is 5 (1.3.6.1.X).
- You can define a maximum of 31 characters, including the . separators.
- All elements after the first four (*1.3.6.1*) are integers between 0 and 127.

Default Value:nne

Example:

```
SNMP Config> delete sub_tree
View name[]?
MIB OID[]?
```

Disable

Use the **disable** command to disable the SNMP protocol or specified traps on the device.

Syntax:

disable snmp

trap

sram-write

snmp Disables SNMP.

Example: `disable snmp`

trap *trap type*

Disables specified traps or all traps.

trap type

Specifies the type of trap to be disabled. Valid trap types are shown in Table 32.

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: none

Example:

```
SNMP Config> disable trap link_up
Community name []?
```

sram-write

Table 32. SNMP Trap Types

Trap Type	Description
all	Specifies all traps in a specified community.
cold_start	A cold start trap means that the transmitting device is reinitializing and that the agent's configuration or the protocol entity implementation may be altered.
warm_start	A warm start trap means that the transmitting device is reinitializing, but the configuration or protocol implementation will remain the same. Specify the community name as part of the command line.
link_down	A link_down trap 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	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	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	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 device.

Syntax:

enable snmp
trap
sram-write

snmp Enables SNMP

Example: `enable snmp`

trap *trap type*

Enables specified traps or all traps.

trap type

Specifies the trap type to be enabled. Valid trap types are shown in Table 32.

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: none

sram-write

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 of the **list community** command for details on the options.

Example: `list all`

```
SNMP Config>list all

   SNMP is enabled
   Trap UDP port: 162
   SRAM write is enabled

            Community Name                 Access
   --------------------------------  -------------------
   oxnard                            Read, Write, Trap
   public                            Read, Trap


           Community Name             IP Address         IP Mask
   --------------------------------  ---------------  ---------------
   oxnard                            1.1.1.2          255.255.255.255
   public                            All              N/A


           Community Name                   Enabled Traps
   --------------------------------  --------------------------------
   oxnard                            Link Down, Cold Restart
   public                            None


            Community Name                        View
   --------------------------------  --------------------------------
   oxnard                            mib2
   public                            All

             View Name                          Sub-Tree
   --------------------------------  --------------------------------
   mib2                              1.3.6.1.2

   Password is set. (security data flow encrypted)
```

list community *option*
Displays the current attributes of an SNMP community. Options are access, address, traps, 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.

Example:

```
SNMP Config  list community access

            Community Name   Access
            public           Read, Write, Trap
            oxnard           Read, Trap
```

Example:

```
SNMP Config>  list community address

            Community Name   IP Address   IP Mask
            public           All          N/A
            oxnard           1.1.1.2      255.255.255.255
```

Example:

```
SNMP Config  list community traps

            Community Name   Enabled Traps
            public           Link Down, Cold Restart
            oxnard           NONE
```

Example:

```
SNMP Config>  list community view

            Community Name   View
            public           All
            oxnard           mib2
```

list views

Displays the current views for a specified SNMP community.

Example:

```
SNMP Config  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 or SNMP password.

Syntax:

set community access

community view

trap_port

password

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.

options

Choose an option from the following list:

read_trap
Allows read access and trap generation to the named community.

write_read_trap
Allows write and read access and trap generation to the community specified.

trap_only
Indicates the community is used only when sending an SNMP trap.

comm_name

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: none

Example: `set community access <options> comm_name`

community view
Use the **set community view** command to assign a MIB view to a community.

comm_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: none

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.

Example: `set community view comm_name <all` *or* `view_text_name>`

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.

Default Value: standard port

Example: `set trap_port udpport#`

UDP Port Number
Specifies a User Datagram Protocol port other than the standard UDP port.

Default Value: 162

password
Use the set password command to specify the password to encrypt and authenticate the security sensitive MIB objects that are defined in the MIB. Setting the password to a string of zero length provides the maximum security by disallowing any access or setting of the security sensitive MIB objects. Setting the password to ″clear″ gives the least amount security by allowing data to flow without authentication. Setting the password to any

other string allows access and setting of the security sensitive MIB objects which are encrypted and authenticated with this password.

Examples:

```
(a) setting the password to a string of zero length:

        SNMP Config>set pa
        Password:
        Remove password? (Yes, No): y
        Password is set to NULL. (security data are not accessible)

    (b) setting the password to "clear":

        SNMP Config>set pa
        Password:
        to verify Enter password again:
        Password is set to "clear". (WARNING: security data flow in clear)

    (c) setting the password to "test":
        SNMP Config>set pa
        Password:
        to verify Enter password again:
        Password is set. (security data flow encrypted)
```

Accessing the SNMP Monitoring Environment

To access the SNMP monitoring environment, type **t 5** at the OPCON prompt (*):

```
* talk 5
```

Then enter the following command at the **+** prompt:

```
+ protocol snmp
SNMP>
```

SNMP Monitoring Commands

This section describes the SNMP monitoring commands.

Table 33 lists the SNMP monitoring commands. The SNMP monitoring 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 monitoring. 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 **reset** command. This command 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 monitoring process.

Enter the SNMP monitoring commands at the `SNMP>` prompt.

Table 33. SNMP Monitoring Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
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.

Table 33. SNMP Monitoring Command Summary (continued)

Command	Function
Disable	Disables traps associated with named communities. Disabling SNMP or SRAM_write must be done using the `SNMP Config>` configuration environment.
Enable	Enables traps associated with named communities. Enabling SNMP or SRAM_write must be done using the `SNMP Config>` configuration environment.
List	Displays the current configuration of SNMP communities, views, access modes, traps, and network addresses.
Reset	Updates the SNMP configuration with the values in the currently stored 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 and password. See 474 for additional information.
Statistics	Displays statistics about the SNMP agent.
Reset	Updates the SNMP configuration with the values in the current stored SNMP configuration.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

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 page 466.

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 469.

Disable

Use the **disable** command to disable specified traps on the device.

For information on using the **disable** command, see "Disable" on page 470.

Enable

Use the **enable** command to enable specified traps on the device.

For information on using the **enable** command, see "Enable" on page 471.

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

For information about using the **list** command, see “List” on page 472.

Reset

Use the SNMP **reset** command to update the SNMP configuration with the values in the current stored SNMP configuration. This action allows changes to the current SNMP configuration when the device is restarted or reloaded.

Save

Use the **save** command to permanently save the specified changes.

Set

For information on using the **set** command, see “Set” on page 473.

Statistics

Use the **statistics** command to display statistics about the SNMP agent.

Syntax:

statistics

Example: `statistics`

```
               Max      Current    Current
              Alloc     Alloc       In Use

SNMP agent:  512000     181144     133120

SNMP MIBs:   1048576     57976      19712
```

The following information is displayed:

Max Alloc
The maximum amount of memory (in bytes) that is reserved for the SNMP component.

Current Alloc
As memory is needed, it is taken from the reserved pool (designated by MAX ALLOC) and moved in to an ″active″ memory pool. The size of this ″active″ memory pool size is indicated by the CURRENT ALLOC value.

Current In Use
This value represents the memory currently allocated from the ″active″ memory pool (designated by CURRENT ALLOC) that is in use by the SNMP component.

SNMP Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

SNMP supports the CONFIG (Talk 6) **delete interface** command with the following consideration:

When the network interface is deleted, all rows in MIB tables indexed by the associated ifIndex will be removed.

GWCON (Talk 5) Activate Interface

SNMP supports the GWCON (Talk 5) **activate interface** command with the following consideration:

When the network interface is activated, all rows in MIB tables indexed by the associated ifIndex will be created.

All SNMP interface-specific commands are supported by the GWCON (Talk 5) **activate interface** command.

GWCON (Talk 5) Reset Interface

Simple Network Management Protocol (SNMP) supports the GWCON (Talk 5) **reset interface** command with the following consideration:

When the network interface is reset, all rows in MIB tables indexed by the associated ifIndex will be remain, but the object values may be different.

All Simple Network Management Protocol (SNMP) interface-specific commands are supported by the GWCON (Talk 5) **reset interface** command.

GWCON (Talk 5) Component Reset Commands

SNMP supports the following SNMP-specific GWCON (Talk 5) **reset** commands:

GWCON, Protocol SNMP, Reset Command

Description:

The SNMP agent activates all changed SNMP configuration values.

Network Effect:

Access to the SNMP agent may be removed or restricted depending on the new configuration.

Limitations:

- It is not possible to use the **reset** command to enable SNMP if it was previously disabled. This requires a reboot.

The following table summarizes the Simple Network Management Protocol (SNMP) configuration changes that are activated when the **GWCON, protocol SNMP, reset** command is invoked:

Commands whose changes are activated by the GWCON, protocol SNMP, reset command
CONFIG, protocol SNMP, add address
CONFIG, protocol SNMP, add community
CONFIG, protocol SNMP, add sub_tree

CONFIG, protocol SNMP, delete address
CONFIG, protocol SNMP, delete community
CONFIG, protocol SNMP, delete sub_tree
CONFIG, protocol SNMP, disable trap
CONFIG, protocol SNMP, disable snmp
CONFIG, protocol SNMP, set community
CONFIG, protocol SNMP, set password
CONFIG, protocol SNMP, set trap_port

GWCON (Talk 5) Temporary Change Commands

SNMP supports the following GWCON commands that temporarily change the operational state of the device. These changes are lost whenever the device is reloaded, restarted, or you execute any dynamically reconfigurable command.

Commands
GWCON, protocol SNMP, add address
GWCON, protocol SNMP, add community
GWCON, protocol SNMP, add sub_tree
GWCON, protocol SNMP, delete address
GWCON, protocol SNMP, delete community
GWCON, protocol SNMP, delete sub_tree
GWCON, protocol SNMP, disable trap
GWCON, protocol SNMP, disable snmp
GWCON, protocol SNMP, set community
GWCON, protocol SNMP, set password

Non-Dynamically Reconfigurable Commands

The following table describes the SNMP configuration commands that cannot be dynamically changed. To activate these commands, you need to reload or restart the device.

Commands
CONFIG, protocol SNMP, enable snmp

Chapter 25. Using DLSw

This chapter describes Data Link Switching (DLSw) and the implementation of the 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 543.

The 2210 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 483
- "Setting up DLSw" on page 498
- "Sample DLSw Configuration" on page 503

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.

DLSw and ATM

A number of ATM to Frame Relay interworking products allow traffic to be shipped on networks that consist of both Frame Relay and ATM devices. Typically, the Frame Relay devices operate at T1 speeds (1.544 Mbps) while ATM speeds are typically one or two orders of magnitude faster. DLS can play an important role in accommodating this speed differential.

DLS is an alternative to bridging SNA traffic. If you have two high speed campus networks that are attached by a slow T1 WAN link, you may decide between bridging and DLS. The unfortunate side effect of bridging is that all broadcast traffic will be forwarded across the T1 link, using up valuable bandwidth. DLSw, on the other hand, terminates the sessions locally, and does not use the WAN for broadcast traffic, resulting in more efficient use of the slower WAN link and ultimately better performance.

How DLSw Works

LLC2, SDLC, and QLLC are connection-oriented protocols. DLSw provides the dynamic characteristics of routable protocols ***and*** preserves both end-to-end reliability and control features for effective communication.

Problems in the Bridging Solution

Figure 43 on page 482 illustrates the traditional approach to bridging LLC2 frames across WAN links. With the traditional 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.

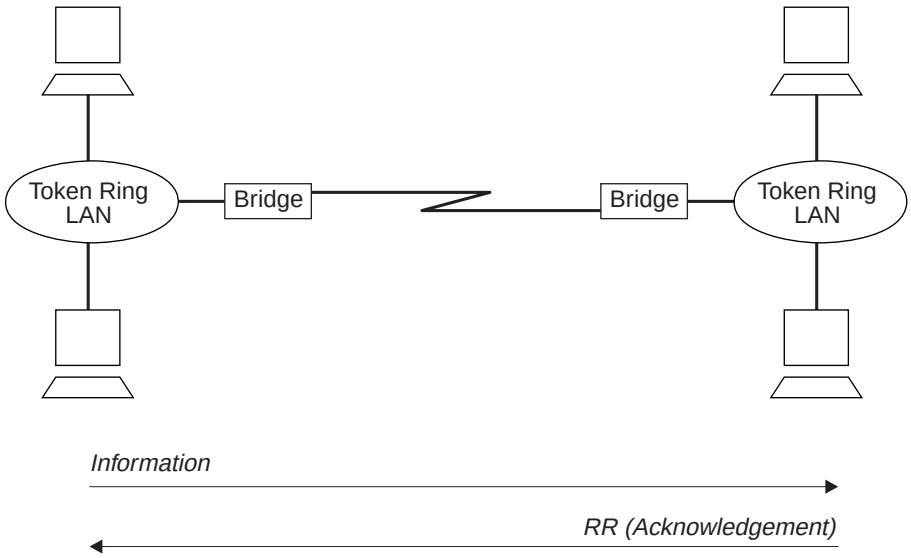


Figure 43. Traditional Approach to Bridging Across WAN Links

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 44 on page 483 illustrates this relationship between two DLSw peer routers, each attached to a Token-Ring Network.

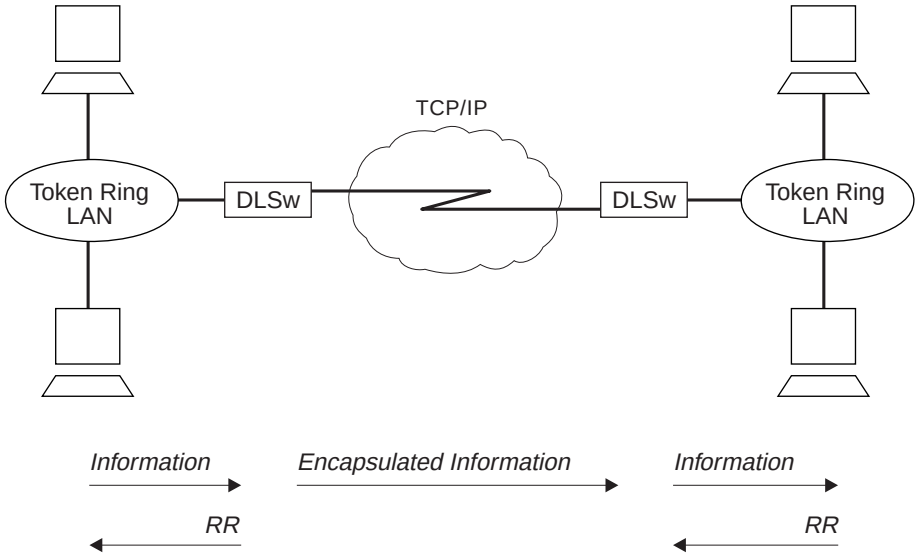


Figure 44. Data Link Switching over the WAN

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 device (see Figure 44), 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, and QLLC control traffic at the local device.
- 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

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 three 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 enables 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 NetBIOS SessionAlive Spoofing parameter controls whether or not the NetBIOS SessionAlive frames are forwarded to the DLSw peer. This parameter is important when NetBIOS sessions have been established across DLSw peers over an ISDN link. If this parameter is enabled and the Keepalive parameter is disabled, then no DLSw traffic will pass between the DLSw partners if idle NetBIOS sessions are established between the DLSw peers. This would allow an underlying ISDN connection to terminate while maintaining an idle NetBIOS session over DLSw.

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 RFC 2166 and allow 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.

Reducing Explorer Traffic: If the amount of explorer traffic being forward between DLSw neighbors is too high, there are some capabilities to reduce this explorer traffic.

DLSw open SAPs
Each DLSw sends a lists of all SAPs open on any interface to its DLSw neighbors via DLSw capabilities exchange. The DLSw neighbors can use this SAP list to limit the explorer traffic sent to this DLSw.

DLSw MAC address lists
Each DLSw can configure a local MAC address list. This list is defined as exclusive (respresents all MAC addresses accessible via this DLSw) or non-exclusive (represents a set of MAC addresses accessible via this DLSw). Each entry in the list contains a MAC address mask and a MAC address value. The entire MAC address list and exclusivity type is sent to all DLSw neighbors via DLSw capabilities exchange. The DLSw neighbors can use this MAC address list to limit the explorer traffic sent to this DLSw.

The MAC address lists operate in a similar manner as the NetBIOS name lists. For information about NetBIOS Name Lists, see “NetBIOS Name Lists” on page 141.

DLSw MAC cache entries
A DLSw can configure individual MAC cache entries that map a particular MAC address with a particular DLSw neighbor. Multiple MAC cache entries can be used to map a particular MAC address with multiple DLSw neighbors. The DLSw uses this list locally to limit where DLSw explorers destined for a configured MAC address are sent.

MAC address filters
MAC address filters configured for the bridge net interface apply to DLSw traffic. These inbound MAC address filters on the bridge net can be used to limit the traffic given to DLSw, thus limiting the explorer traffic sent to DLSw partners. For more information about MAC filters, refer to “Using and Configuring Mac Filtering”and“Monitoring MAC Filtering”in the *Software User’s Guide*.

Limiting Explorers per transport queue
Occasionally, the performance of a TCP session to a DLSw partner can be significantly impacted by a burst of traffic or network problem. In these cases, it is possible for DLSw to queue explorer traffic (SNA and NetBIOS) waiting to be sent to the DLSw partner. If the amount of data in the queue gets too large it can have an adverse affect on the memory. In order to reduce this impact, DLSw has two configuration parameters that control how many SNA explorer frames can be simultaneously queued to a single DLSw partner and how many NetBIOS explorer frames can be simultaneously queued to a single DLSw partner. These parameters are ’Maximum SNA explorers per transport queue’ and ’Maximum NetBIOS explorers per transport queue’.

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

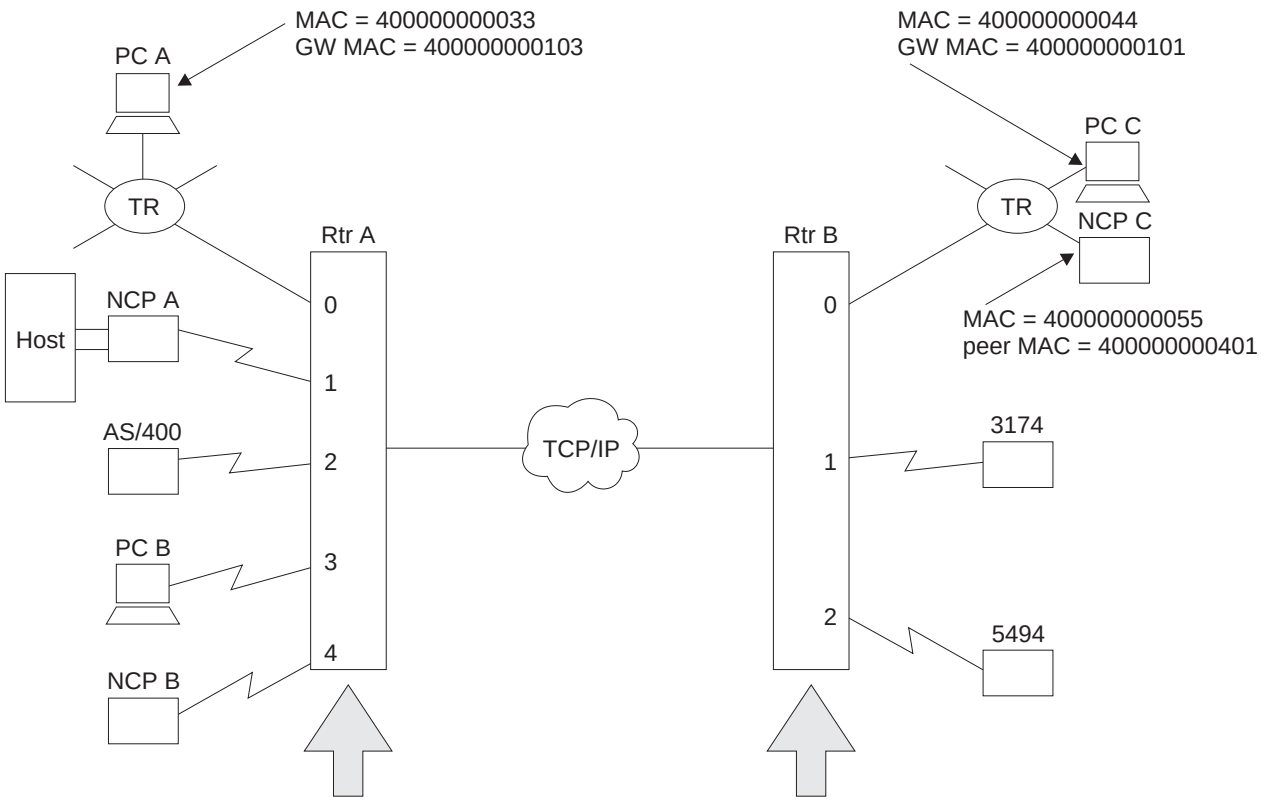
- Ethernet/802.3
- ATM (as a LAN Emulation client)
- Frame Relay (using RFC 1490/2427 bridged frame formats)
- PPP
- Dial circuits that use PPP or FR framing (for example, ISDN)

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 1, 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.

Note: DLSw supports SDLC PU1 devices communicating with SDLC-attached or LAN-attached devices supporting PU1 devices (for example, 3745). DLSw can also support PU1 devices communicating with devices not supporting PU1 devices by emulating the PU1 device as a PU2.0 device to the host.



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 45. Example DLSw SDLC Configurations

Figure 45 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 45 on page 488) 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:

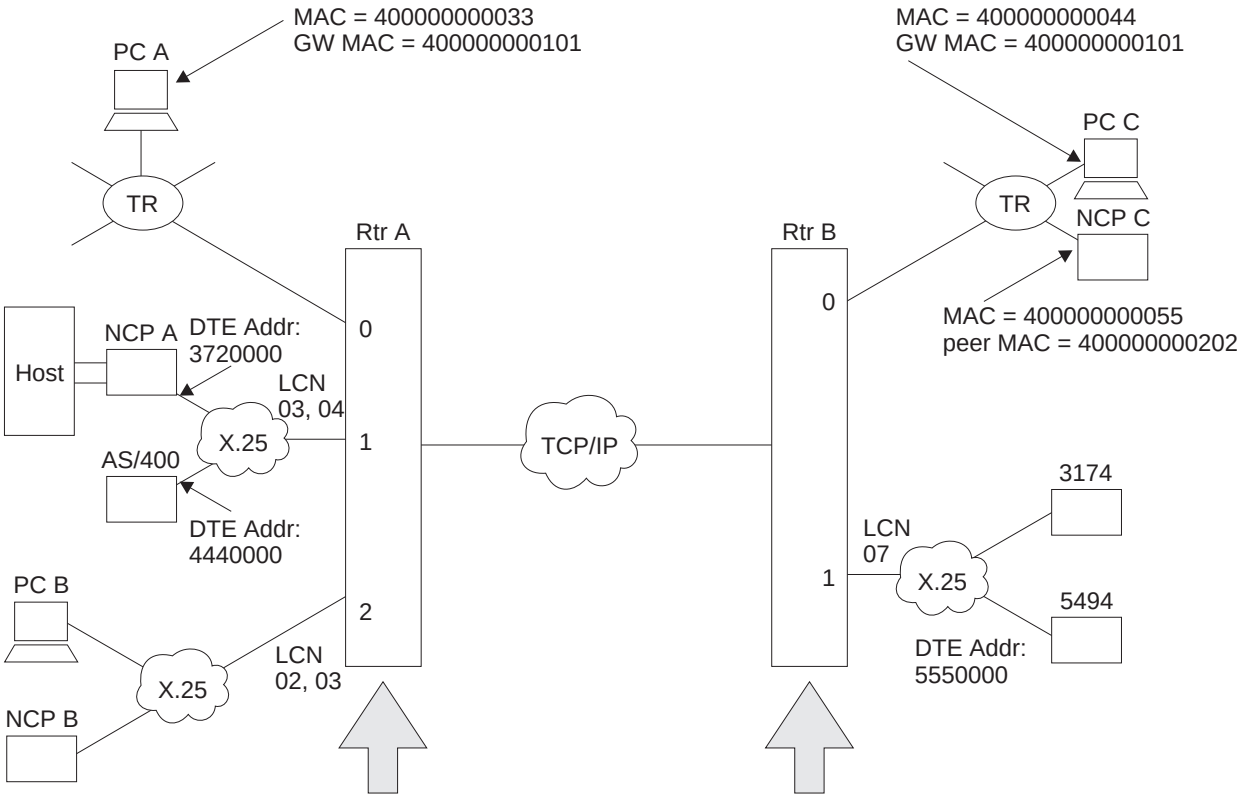
- You need one of the SDLC-SDLC functions or configurations that is not currently supported by DLSw

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 the *Software User's Guide*.

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 46. Example DLSw QLLC Configurations

Figure 46 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, because 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, because 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, because 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 because 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.

To allow any X.25 calling party (regardless of DTE address or CID) to be accepted by DLSw and matched to a specific DMAC and DSAP (one per box), you can

configure a QLLC Destination record with a CID value of ″ANYCALL″ and the desired DMAC and DSAP. DLSw dynamically assigns the SMAC and SSAP. If this feature is used, DLSw accepts all calls. No calls are presented to APPN and all security features associated with address mapping are bypassed.

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 502 for an example of these steps, and refer to the chapter "Using the X.25 Network Interface" in *Software User's Guide* 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, refer to the chapter entitled “Using, Configuring, and Monitoring XTP” in the *Software User’s Guide*.

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 47 , this internal interface is the equivalent of collapsing a DLC connection (for example, LLC over a LAN) into a single software interface.

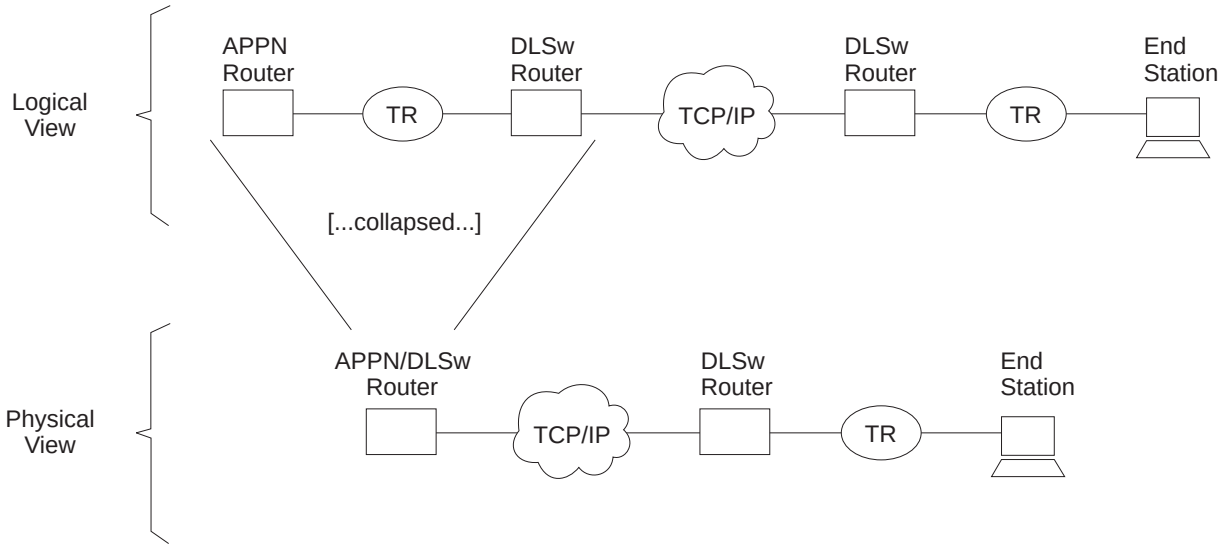


Figure 47. APPN-to-DLSw Software Interface

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 enable

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.*

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 497.

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.

It is possible to disable the neighbor priority feature for all MAC addresses or for certain sets of MAC addresses. To disable it for all MAC addresses, set the *wait neighbor priority timer* to 0. To disable it for a set of MAC addresses, create a MAC cache explorer override and set its *wait neighbor priority timer* to 0.

If the neighbor priority feature is disabled, the DSLw partner information is not cached for the MAC address. SNA and NetBIOS explorers are always sent to all applicable DLSw partners and the first DSLw partner to respond is used to establish the DLSw session (regardless of its priority).

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 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 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 using. The DLSw router that initiates the circuit will choose the circuit priority to use. If the local DLSw router initiates the circuit, the circuit priority it chooses is based on the configured circuit priority defaults and circuit priority overrides. If the remote DLSw router initiates the circuit, the local DLSw router will notify the remote DLSw router of its need to use a circuit priority based on the configured defaults and overrides, but the remote DLSw router may choose a different value. 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.

There are four default circuit priorities that can be configured:
- Default SNA explorer traffic circuit priority
- Default SNA session traffic circuit priority
- Default NetBIOS explorer traffic circuit priority
- Default NetBIOS session traffic circuit priority

These different values allow different ratios of SNA and NetBIOS and explorer and session traffic to be assigned.

There may be cases where you want to assign a certain circuit priority to specific traffic. For example, you might want to make traffic destined for a certain SNA MAC address higher priority than all other traffic. This can be accomplished using circuit priority overrides (**add priority**) command. This enables an explorer and session circuit priority to be assigned to a specific range of source MAC addresses and SAPs and destination MAC addresses and SAPs. These circuit priority overrides are evaluated in the order in which they are configured. The circuit priority is set to the value in the first circuit priority override match found. If no circuit priority override match is found, the default circuit priority is used.

Setting up DLSw

The following sections explain the setup procedures for DLSw:
- “DLSw Configuration Requirements”
- “Setting Global Buffers” on page 499
- “Configuring Adaptive Source Route Bridging (ASRT) for DLSw” on page 499
- “Configuring the Internet Protocol (IP) for DLSw” on page 500
- “Configuring OSPF for DLSw” on page 500
- “Configuring SDLC Interfaces” on page 501
- “Configuring X.25 Interfaces” on page 502
- “Configuring DLSw” on page 502

In addition, a sample DLSw configuration with explanatory notes has been included (see Figure 48 on page 504).

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 34.

Table 34. 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	For dynamic routing or 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 4-MB DRAM 2210, 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 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 the **add protocol-filter dsap 4** command at the `Config ASRT>` prompt.

   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 ensure that DLSw partners have a route to each other's internal address. The easiest way to accomplish this to enable dynamic routing using either RIP or OSPF as your routing protocol. Using OSPF is recommended since it usually requires less network overhead than RIP.
   - See "Configuring OSPF for DLSw" to enable OSPF.
   - Enter **enable RIP** at the `IP Config>` prompt to enable RIP.
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, the internal address can be advertised if host routes are being advertised. To enable advertisement of the host in RIP routes, use the **enable sending host-routes** command from IP configuration.
   - If you are using OSPF, the internal address is advertised automatically.

Configuring OSPF for DLSw

If you want to use OSPF as your routing protocol, you need to configure it 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 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 SDLC Interfaces

The SDLC configuration command enables you to create or modify the SDLC interface configuration as part of the DLSw configuration process.

Note: If SDLC is the encapsulator for V.25 bis, the physical link parameters cannot be set at the SDLC level as they must be configured at the V.25 bis level. In this case, you must not configure the following SDLC parameters:
- Role - This must be primary.
- Group - You cannot set a group poll address.
- Type - This must be point-to-point.
- Duplex
- Idle state
- Clocking
- Speed
- Cable
- Encoding
- Inter-frame delay

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
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-poll** 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 (that is, 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 the *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. At the `DLSw config>` prompt, 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 MRS-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 device 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 48 on page 504.

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 are 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.

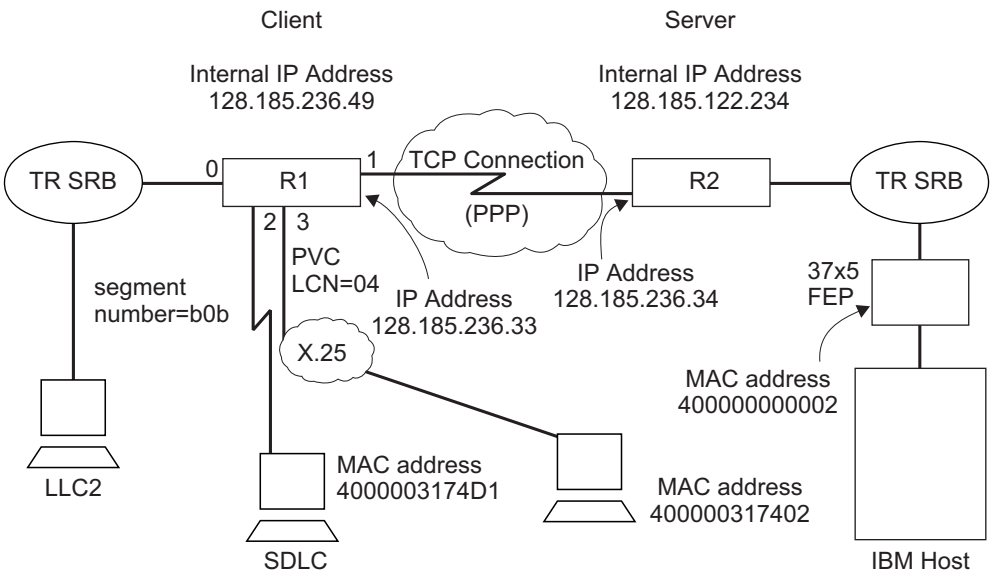


Figure 48. Sample Diagram for DLSw Configuration

Sample Configuration Commands

This section provides examples of the following:

- "Step 1: Adding Devices"
- "Step 2: Configuring Protocols" on page 508
- "Step 3: Implementing Protocol Filtering" on page 511
- "Step 4: Configuring DLSw" on page 512

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.

Enter the **list device** command at the `config>` prompt to display a list of the configured devices and their interface numbers.

```
Config (only)>list device
Ifc 0 Ethernet                    CSR  81600, CSR2  80C00, vector 94
Ifc 1 WAN PPP                     CSR 381620, CSR2 380D00, vector 125
Ifc 2 WAN SDLC                    CSR  81640, CSR2  80E00, vector 92
Ifc 3 WAN X.25                    CSR  81620, CSR2  80D00, vector 93
Ifc 4 WAN Frame Relay             CSR 381640, CSR2 380E00, vector 124
Ifc 5 Token Ring                  CSR 600000, vector 95
```

Notice that this **list** command shows that a token-ring device has been assigned to interface 5.

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 5
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 2210:

```
SDLC 2 Config>list link
Link configuration for: LINK_2   (ENABLED)

Role:         PRIMARY          Type:        POINT-TO-POINT
Duplex:       FULL             Modulo:      8
Idle state:   FLAG             Encoding:    NRZ
Clocking:     EXTERNAL         Frame Size:  2048
Speed:        0                Group Poll:  00
Cable:        RS-232 DTE

Timers:    XID/TEST response:  2.0 sec
           SNRM response:      2.0 sec
           Poll response:      0.5 sec
           Inter-poll delay:   0.2 sec
           RTS hold delay:     DISABLED
           Inter-frame delay:  DISABLED
           Inactivity timeout: 30.0 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 48 on page 504, you must configure interface 3 to be X.25 and have QLLC support for DLSw on the indicated PVC. The following example starts from scratch with a non-X.25 serial interface. 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 14. Using IP” on page 213.

   - 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 48 on page 504).

```
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 48 on page 504) **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.

```
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 is                   ENABLED
SRB Segment number                     000
MAC <-> IP mapping cache size          128
Max DLSw sessions                      1000
DLSw global memory allotment           141312
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 SessionAlive Spoofing DISABLED
Dynamic Neighbor Priority              MEDIUM

QLLC base source MAC address           40514C430000
QLLC maximum dynamic addresses         64
Type of local MAC list                 NON-EXCLUSIVE
Use of local MAC list is               ENABLED
Use of remote MAC list is              ENABLED
SNA explorer limit                     100
NetBIOS explorer limit                 100
```

You enable DLSw, and set the SRB segment number. The segment number refers to the token-ring device, as shown in Figure 48 on page 504.

```
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 Figure 48 on page 504), 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]?
Enable/Disable NetBIOS SessionAlive Spoofing (E/D) [D]?
Neighbor Priority (H/M/L) [M]?

DLSw config>list group

   Group#                  Xmit    Rcv      Max      Keep-    SessAlive
Mcast IP Addr Role    CST Bufsize Bufsize  Segsize  alive    Spoofing  Priority
------------- ------- --- ------- -------- -------- -------- --------- --------
Group  1      CLIENT   p    5120    5120     1024   DISABLED 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 Figure 48 on page 504). 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]?
Enable/Disable  NetBIOS SessionAlive Spoofing (E/D)[D]?
Neighbor Priority (H/M/L) [M]?

DLSw config>list tcp
                     Xmit    Rcv     Max      Keep-    SesAlive
Neighbor        CST  Bufsize Bufsize Segsize  Alive    Spoofing Priority
--------------- ---  ------- ------- -------  -------- -------- ---------
128.185.122.234  p     5120    5120    1024   DISABLED DISABLED MEDIUM
```

Define Each SDLC Link Station: You must define each SDLC link station.

```
DLSw config>add sdlc
Interface # [0]? 2
SDLC Address or 'sw' (switched dial-in) [C1]?
Source MAC address [4000112402C1]? 4000003174d1
Source SAP in hex [4]?
Destination MAC address [000000000000]? 400000000002
Destination SAP in hex [0]? 4
PU type (1/2/4/5) [2]?
XID0 block num in hex (0-0xfff) [0]? 017
XID0 id num in hex (0-0xfffff) [0]? 00001
Poll with TEST (T), SNRM (S), or DELAYED SNRM (D) [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–FE), or one of the following:
   'SNA', 'NB', or LNM [4]? sna
SAP(s) 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 is                   ENABLED
SRB Segment number                     020
MAC <-> IP mapping cache size          128
Max DLSw sessions                      1000
DLSw global memory allotment           141312
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 SessionAlive Spoofing DISABLED
Dynamic Neighbor Priority              MEDIUM

QLLC base source MAC address           40514C430000
QLLC maximum dynamic addresses         64
Type of local MAC list                 NON-EXCLUSIVE
Use of local MAC list is               ENABLED
Use of remote MAC list is              ENABLED
SNA explorer limit                     100
NetBIOS explorer limit                 100
```

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
```

Chapter 26. Configuring and Monitoring DLSw

This chapter describes how to configure and monitor the Data Link Switching protocol. It includes the following sections:

Accessing the DLSw Configuration Environment

Use the CONFIG process to change the configuration of the router. The new configuration takes effect when the device 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 Console

For help using the Command Line Interface, press ESCAPE, then '?'

* 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>
```

Preconfiguration 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.

Important Consideration

For the IBM 2210 with 4 MB DRAM running DLSw:

- The maximum number of global buffers should be set to 50. This will ensure that DLSw has the memory needed to run effectively.
- You can use the **set global-buffers** command from the main `Config>` prompt to set the maximum number of global buffers.

DLSw Configuration Commands

This section summarizes and explains the DLSw configuration commands. The DLSw configuration commands enable you to create or modify a DLSw configuration. Table 35 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 35. DLSw Configuration Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
Add	Adds an SDLC link station, a TCP neighbor IP address, a QLLC station or destination, cache entries, MAC address list entries, circuit priority overrides, or MAC cache explorer-overrides.
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, a TCP connection, a QLLC station or destination, cache entries, MAC address list entries, circuit priority overrides, or MAC cache explorer overrides.
Disable	Disables the DLSw protocol, SDLC link station, LLC disconnect function, dynamic neighbors, a QLLC station or interface, or local and remote MAC address list use.
Enable	Enables the DLSw protocol, SDLC link station, LLC disconnect function, dynamic neighbors, a QLLC station or interface, local and remote MAC address list use, or IPv4 DLSw precedence bit setting.
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, QLLC destinations, stations, and interfaces, cache entries, or MAC address list entries. 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, parameters for QLLC operation, and MAC address list-related parameters.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

Add

Use the **add** command to configure an SDLC link station, a TCP neighbor IP address, a QLLC station or destination, cache entries, MAC address list entries, circuit priority overrides, and MAC cache explorer overrides.

Syntax:

add cache-entry

explorer-override

mac-list

priority

qllc...

sdlc

tcp

cache-entry

Adds a configured MAC cache entry. This cache entry maps a specific MAC address to a specific DLSw peer. One MAC address can be mapped to multiple DLSw peers by adding multiple cache entries.

Example: add cache-entry

```
Enter MAC Address [400000000000]? 10005a123456
Enter the DLSw neighbor IP Address [0.0.0.0]? 128.185.122.234

MAC cache entry has been created.
```

explorer-override

Adds a MAC cache explorer override entry. This override allows a set of MAC addresses to possess different MAC cache and explorer flow characteristics. When a MAC cache entry is created, the list of explorer overrides is searched in the order that they are configured. If a match is found, then the MAC cache and explorer related parameters from the first matching explorer override are used. If no match is found, then the DLSw global MAC cache and explorer related values are used.

Example: add explorer-override

```
Enter MAC address value [000000000000]?400031740000
Enter MAC address mask [FFFFFFFFFFFF]?ffffffff0000
Database age timeout (0-1000 secs. Decimal)[0.0]?0
Max wait timer ICANREACH (1-1000 secs. Decimal) [2.0?]
Neighbor priority wait timer (0,0-5.0 secs. Decimal) [2.0]?0
Time to delay sending test response (0.0-5.0 secs. Decimal) [0.0]?
Forwarding explorers (E/L/D) [E]?

Enter position in explorer override list to insert new entry ....
  Record number (0=add at end of list) [0]?
Explorer override record has been created.
```

MAC address value and MAC address mask

When combined, these two fields represent a set of MAC addresses. To determine whether a configured MAC cache explorer override record with the given value and mask should be used for a specific MAC address, the following algorithm is used:

```
if ((<specific MAC address>AND<override's mask>) == <override's value>)
match on explorer override is found; use override's value
```

Database age timeout

Specifies how long to hold unused DLSw 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.

Neighbor priority wait timer

Specifies the 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.

A value of zero indicates that the neighbor priority feature is not to be used. There will be no cached DLSw peer information for the MAC address. A CANREACH is always sent and the first DLSw peer to send an ICANREACH is used (regardless of its priority).

Delay sending TEST response

The amount of time to wait after completing exploration for a MAC address before sending the TEST response. This is useful if there are two DLSw 2210s on the same bridged LAN capable of reaching the same MAC address via DLSw peers. If one DLSw 2210 is preferable, the TEST response can be delayed at the less preferable DLSw 2210.

Forwarding explorers

Specifies whether explorers should be forwarded to all applicable DLSw peers, forwarded only on the local TCP connection, or not forwarded at all.

Position in explorer override list to insert new entry

Since the first matching MAC cache explorer override match is used, the order in which the explorer override entries are configured is important. This field specifies where in the current override list to insert this new entry. The **list explorer-override** command can be used to see the current explorer override list. A value of zero for this field specifies to add the new entry at the end of the current list.

mac-list

Adds a local MAC address list entry. All added local MAC address list entries make up the local MAC address list. The local MAC address list is sent to each DLSw peer to indicate the set of MAC addresses that are reachable using this DLSw.

Example: `add mac-list`

```
Enter MAC Address Value[400000000000]? 10005a000000
Enter MAC Address Mask [ffffff000000]?

MAC list entry has been created.

For the new entry to take effect, you must restart or commit the change using
     't 5': SET MAC LIST
```

Enter MAC Address Value and Enter MAC Address Mask

These two fields when combined represent a set of MAC addresses reachable using this DLSw. If a frame is received at a peer DLSw, then these two fields are used in the following algorithm:

```
if ( (<frame's destination MAC address> AND <MAC Address Mask>)
     == <MAC Address Value> )
match on MAC address list found; forward frame to this DLSw
```

priority

Adds a circuit priority override entry. When a DLSw session is being established, the list of circuit priority overrides is searched in the order that they are configured. If a match on the source SAP range and source MAC address range and destination SAP range and destination MAC address range is found, then the matching circuit priority override entry's session

and explorer priorities are used. If no match with any circuit priority override entry is found, then the default circuit priority values are used.

Example: `add priority`

```
Enter range of source SAPs .....
  Lower source sap value [0]?
  Upper source sap value [FE]?

Enter range of source MAC addresses .....
  Lower source MAC address [000000000000]?
  Upper source MAC address [FFFFFFFFFFFF]?

Enter range of destination SAPs .....
  Lower destination sap value [0]?
  Upper destination sap value [FE]? c

Enter range of destination MAC addresses .....
  Lower destination MAC address [000000000000]? 10005a000000
  Upper destination MAC address [FFFFFFFFFFFF]? 10005affffff

Enter desired circuit priorities .....
Priority for session traffic (C/H/M/L) [M]? c
Priority for explorer traffic (C/H/M/L) [M]? m

Enter position in circuit priority override list to insert new entry .....
  Record number (0=add at end of list) [0]?
Circuit priority override record has been created.
```

Lower source sap value

Upper source sap value
These two fields when combined represent the range of source saps assigned to this circuit priority override. If the value of the source sap does not matter, then specify the full range of source sap values (lower source value = 0 and upper source value = fe).

Lower source MAC address

Upper source MAC address
These two fields when combined represent the range of source MAC addresses assigned to this circuit priority override. If the value of the source MAC address does not matter, then specify the full range of source MAC address values (lower source MAC address = 000000000000 and upper source MAC address = ffffffffffff).

Lower destination sap value

Upper destination sap value
These two fields when combined represent the range of destination saps assigned to this circuit priority override. If the value of the destination sap does not matter, then specify the full range of destination sap values (lower destination sap value = 0 and upper destination sap value = fe)

Lower destination MAC address

Upper destination MAC address
These two fields combined represent the range of destination MAC addresses assigned to this circuit priority override. If the value of the destination MAC address does not matter, then specify the full range of destination MAC address values (lower destination MAC address = 000000000000 and upper destination MAC address = ffffffffffff).

Priority for session traffic
The circuit priority to assign to all session traffic that matches this circuit priority override entry's range of source SAPs, source MAC addresses, destination SAPs, and destination MAC addresses.

Priority for explorer traffic
The circuit priority to assign to all explorer traffic that matches this circuit priority override entry's range of source SAPs, source MAC addresses, destination SAPs, and destination MAC addresses.

Position in circuit priority override list to insert new entry
Since the first matching circuit priority override match is used, the order in which the circuit priority override entries are configured is important. This field specifies where in the current circuit priority override list to insert this new entry. The **list priority** command can be used to see the current circuit priority override list. A value of zero for this field specifies to add the new entry at the end of the current list.

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:

addqllc	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.

CAUTION:
If a QLLC Destination record is configured with ″ANYCALL″, all calls are accepted by DLSw (regardless of DTE address or connection id). Be aware, there are security issues when accepting all incoming calls.

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 or 'sw' (switched call-in) [C1]?
Source MAC address [4000112402C1]? 4000003174d1
Source SAP in hex [4]?
Destination MAC address [000000000000]? 400000000002
Destination SAP in hex [0]? 4
PU type (1/2/4/5) [2]?
XID0 block num in hex (0-0xfff) [0]? 017
XID0 id num in hex (0-0xfffff) [0]? 00001
Poll with TEST (T), SNRM (S), or DELAYED SNRM (D) [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 or "sw". "Sw" indicates that this is a switched SDLC call-in circuit.

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.

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 monitoring **flip** command to help flip the MAC address, in such cases.

Note: The destination address cannot have a value of 0 if this is a switched SDLC call-in circuit (indicated by “sw” as the SDLC address).

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.

Note: The destination SAP cannot have a value of 0 if this is a switched SDLC call-in circuit (indicated by “sw” as the SDLC address).

PU type
The SNA Physical Unit type of the SDLC station. This may have one of the following values:

1 A PU1 node. If a PU type of 1 is selected, the user is prompted to select whether communicating with the PU4/5 via SDLC, LAN, or as emulated 2.0. This allows the user to choose how the SDLC-attached PU1 device communicates with the PU4/5 device.

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).

PU1 device communicates with the PU4/5 device
If a PU type of 1 is chosen, the method by which the PU1 device communicates with the PU4/5 device is also chosen. The choices are:

SDLC The SDLC-attached PU1 device communicates with an SDLC-attached PU4/5 device that supports SDLC-attached PU1 devices.

LAN The SDLC-attached PU1 device communicates with

a LAN-attached PU4/5 device that supports LAN-attached PU1 devices. This option is rarely used.

as Emulated 2.0
The SDLC-attached PU1 device communicates with a LAN-attached PU4/5 device that does not support LAN-attached PU1 devices. This is accomplished by emulating the PU1 device as though it is a PU2.0 devices to the Host (the Host should have the PU type defined as 2). The option can also be used for an SDLC-attached PU1 device to communicate with a Channel-attached, SDLC-attached, or QLLC-attached device.

Note: You cannot set this parameter for a switched SDLC call-in circuit. A PU type of 2.0 is assumed.

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 1 or 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. It should also be left zero for PU1 devices communicating with SDLC-attached devices supporting PU1 devices. If non-zero, it must match the corresponding PU address field in the IBM NCP switched major node configuration for the link station.

Note: If you set this parameter to a non-zero value for a switched SDLC call-in circuit, the configured information is placed in XID_0. For a switched SDLC call-in circuit, the configured XID_0 block num is used differently. The software assumes that the call-in station will always build its own XID_0. If this parameter is set to a non-zero value, the station's XID_0 is modified with the configured value. If this parameter is set to a zero value, the station's XID_0 is not modified.

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.

Poll type
Defines how and when to poll the SDLC device:

TEST Poll the SDLC device with TEST frame when the interface becomes active

SNRM Poll the SDLC device with a SNRM frame when the interface becomes active.

DELAYED SNRM
Poll the SDLC device with a SNRM frame when the DLSW session has been established and the interface is active.

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]?
Enable/Disable NetBIOS SessionAlive Spoofing (E/D) [D]?
Neighbor Priority (H/M/L) [M]?
```

Enter the DLSw neighbor IP Address
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 484.

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).

Enable/Disable NetBIOS SessionAlive Spoofing (E/D)
Indicates whether you want to discard NetBIOS SessionAlive I-Frames (do not forward to the DLSw partner). The default is D (Disable) meaning do not discard frame.

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 78 for an explanation of each of these commands.

Syntax:

ban

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–FE), or one of the following:
   'SNA', 'NB', or LNM [0]? sna
SAP(s) 0 4 8 C closed 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, a QLLC station or destination, cache entries, MAC address entries, circuit priority overrides and MAC cache explorer overrides from the DLSw configuration.

Syntax:

delete cache-entry
explorer-override
mac-list
priority
qllc...
sdlc
tcp

cache-entry
Removes a configured MAC cache-entry.

Example: `delete cache-entry`

```
Enter mac cache record number [1]? 1
MAC cache entry has been deleted
```

mac cache record number
The record number of the MAC cache entry to be deleted. The record number can be determined by doing a **list cache all** configuration command.

explorer-override
Removes a MAC cache explorer override entry.

Example: `delete explorer-override`

```
Enter explorer override record number [1]?
Explorer override record has been deleted.
```

Explorer override record number
The record number of the MAC cache explorer override entry to be deleted. The record number can be determined by doing a **list explorer-override** command from *talk 6*.

mac-list
Removes a local MAC address list entry.

Example: `delete mac-list`

```
Enter mac list record number [1]? 1
Local MAC list entry 10005A000000 / FFFFFF000000 has been deleted.

For the deletion to take effect, commit the change using
  't 5': SET MAC-LIST.
```

mac list record number
The record number of the MAC list entry to be deleted. The record number can be determined by doing a **list mac-list all** configuration command.

priority
Removes a circuit priority override entry.

Example: `delete priority`

```
Enter circuit priority override record number [1]? 1
Circuit priority override record has been deleted.
```

Circuit priority override record number
The record number of the circuit priority override entry to be deleted. The record number can be determined by doing a **list priority** configuration command.

qllc Removes support for a QLLC station on an X.25 network, or for a DLSw destination for QLLC stations.

Syntax:

delete 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.

Syntax:

delete sdlc

Example: `delete sdlc`

```
Interface #[0]? 1
SDLC Address or 'sw' (switched dial-in) [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 or “sw” for a switched SDLC call-in circuit.

tcp Removes the IP address (*ip_address*) of the DLSw peer to which you can make a TCP connection.

Syntax:

delete tcp *ip_address*

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, a QLLC station or interface, or use of local and remote MAC address lists.

Syntax:

disable dls
dynamic-neighbors
llc
mac-list
qllc...
sdlc

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`

mac-list
Disables the use of local or remote MAC address lists.

Syntax:

mac-list local

remote

Example: `disable mac-list local`

```
Use of local MAC list is  DISABLED

For the change to take effect, restart or commit the change using
  't 5' : 'SET MAC-LIST'.
```

Example: `disable mac-list remote`

```
Use of remote MAC list is  DISABLED

For the change to take effect, restart or commit the change using
  't 5' : 'SET MAC-LIST'.
```

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 or 'sw' (switched dial-in) [C1]?
Record updated
```

Enable

Use the **enable** command to enable the DLSw protocol,SDLC link station, LLC disconnect function, dynamic neighbors, a QLLC station or interface, or use of local and remote MAC address lists.

Syntax:

enable dls

dynamic-neighbors

ipv4 dlsw precedence

llc

mac-list

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.

ipv4 dlsw precedence
Sets the router to set the IP precedence bits for IP version 4. These precedence bits are read by the BRS feature of the router to prioritize DLSw traffic.

Example:

```
enable IPv4 DLSw Precedence
IPv4 Precedence is now enabled.
```

llc Allows the router to terminate an LLC connection upon the loss of the TCP connection.

mac-list
Disables the use of local or remote MAC address lists.

Syntax:

mac-list	local
	remote

Example: enable mac-list local

```
Use of local MAC list is  ENABLED

For the change to take effect, restart or commit the change using
  't 5' : 'SET MAC-LIST'.
```

Example: enable mac-list remote

```
Use of remote MAC list is  ENABLED

For the change to take effect, restart or commit the change using
  't 5' : 'SET MAC-LIST'.
```

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:

qllc	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 or 'sw' (switched dial-in) [C1]?
Record updated
```

Join-Group

Use the **join-group** command to enable 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 484. 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:

The following example is for the default [G]. The descriptions following the example contains information for both (G) and (M).

```
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]?
Enable/Disable NetBIOS SessionAlive Spoofing (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.

Multicast IP address
The multicast IP address is a DLSw Version 2 compliant multicast IP address in the range of 224.0.10.0 to 224.0.10.191 that is used to send and/or receive DLSw explorer traffic.

Read Only , Write Only or Read Write
This parameter indicates whether the configured multicast IP address

should be used to only receive explorer traffic (Read Only), only send explorer traffic (Send Only), or to both send and receive explorer traffic (Read Write).

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 484.

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).

Enable/Disable NetBIOS SessionAlive Spoofing (E/D)
Indicates whether you want to discard NetBIOS SessionAlive I-Frames (do not forward to the DLSw partners associated with this group). The default is D (Disable) meaning do not discard frames.

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, QLLC stations, destinations, interfaces, cache entries, MAC address list entries, circuit priority overrides, and MAC cache explorer overrides.

Syntax:

list cache
dls
explorer-override
groups
llc2
mac-list
open
priority
qllc...
sdlc
tcp
timers

cache Lists configured MAC address cache entries.

Syntax:

cache all
entry-number

cache all

Example: `cache all`

```
Entry  Mac Address   IP Address
-----  ------------  ---------------
    1  10005A123456  128.185.236.49
    2  10005A789ABC  128.185.236.49
```

cache entry-number

Example: `cache entry-number`

```
Enter mac cache record number [1]?

Entry  Mac Address   IP Address
-----  ------------  ---------------
    1  10005A123456  128.185.236.49
```

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. See page 549 for an example of this output.)

explorer-override
Displays the configured MAC cache explorer overrides.

Example: `list explorer-override`

```
    Explorer      Explorer      DB Age    Wait ICR  Nbr Pri   TESTrsp  Forwarding
ID  MAC Value     MAC Mask      Timeout   Timeout   Timeout   Delay    Explorers
--  ------------  ------------  --------  --------  --------  -------  ---------
 1  400031740000  FFFFFFFF0000  DISABLED    20      DISABLED  0.0      AllPartners
 2  10005A000000  FFFFFF000000   1200       20      2.0       0.0      NoPartner
```

groups

Displays group information for this DLSw peer that was configured with the **groups config** command. See “List” on page 548 for more information.)

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 539.) 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.

mac-list

Lists configured MAC address list entries.

Syntax:

mac	all
	entry-number

mac-list all

Example: `list mac-list all`

```
Entry  Mac Value     Mac Mask
-----  ------------  ------------
    1  10005A000000  FFFFFF000000
    2  400031740000  FFFFFFFF0000
```

mac-list entry-number

Example: `list mac-list entry-number`

```
Enter mac list record number [1}?

Entry  Mac Value     Mac Mask
-----  ------------  ------------
    1  10005A000000  FFFFFF000000
```

open Displays all open SAPs and their associated interfaces.

Example: `list open`

```
Interface    SAP(s)
    0         0  4
    1         0  4  8  C
```

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
Default priority for SNA DLSw session traffic is          MEDIUM
Default priority for NetBIOS DLSw session traffic is      MEDIUM
Default priority for SNA DLSw explorer traffic is         MEDIUM
Default priority for NetBIOS DLSw explorer traffic is     MEDIUM


Message allocation by C/H/M/L priority is   4/3/2/1
Maximum frame size for NetBIOS is           2052


    Source/  SAP      MAC Address                  Session   Explorer
ID  Dest     Range    Range                        Priority  Priority
--  -------  -------  --------------------------   --------  --------
 1  Source:  00 - FE  000000000000 - FFFFFFFFFFFF  CRITICAL  MEDIUM
    Dest  :  00 - 0C  10005A000000 - 10005AFFFFFF
 2  Source:  04 - 04  400031740000 - 40003174FFFF  CRITICAL  MEDIUM
    Dest  :  00 - FE  000000000000 - FFFFFFFFFFFF
```

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 session traffic a priority of Medium, with a message allocation of 4/3/2/1, the router processes 4 SNA session 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 520.

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 520. 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.

Note: Switched SDLC call-in circuits are indicated by “FF(sw)” in the address field.

Example: `list sdlc all`

```
Net  Addr    Status    Source SAP/MAC    Dest SAP/MAC     PU Blk/IdNum  PollType
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
3    FF(sw)  Enabled   04 4000103d01d2   04 400000000003  2  017/00002
```

Net The ID number of the interface that connects to the SDLC link station.

Addr The SDLC address, in the range 01 – FE or “FF(sw)” for a switched SDLC call-in circuit, 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.

PollType
The type of SDLC frame that the router uses to make initial contact with the SDLC station, either a TEST frame, a SNRM frame or a delayed SNRM frame (a SNRM frame sent only after the DLSw session is established). 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`

```
                     Xmit    Rcv     Max     Keep-    SesAlive
Neighbor        CST Bufsize Bufsize Segsize Alive    Spoofing Priority
--------------- --- ------- ------- ------- -------- -------- --------
128.185.122.234  p    5120    5120    1024  DISABLED DISABLED MEDIUM
128.185.14.1     p    5120    5120    1024  DISABLED 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.

SesAlive Spoofing
The status of the NetBIOS SesAlive spoofing 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          20    seconds
Join Group Interval               900   seconds
Neighbor priority wait timer      2.0   seconds
Neighbor Inactivity Timer           5   minutes
Time to delay sending test resp.  0.0   seconds
```

For additional information, refer to the **list timers** command.

NetBIOS

Displays the NetBIOS configuration prompt.

For a description of NetBIOS commands, see “NetBIOS Commands” on page 155.

Syntax:

netbios

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-FE), or one of the following:
  'SNA', 'NB', or LNM [4]? sna
SAP(s) 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, parameters for QLLC operation, MAC address list related parameters and circuit priority overrides.

Syntax:

set cache
dynamic-tcp
explorer-limit
llc2
mac-list
maximum
memory
priority
qllc
srb
timers

cache The **set cache** command enables 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]?
Enable/Disable NetBIOS SessionAlive Spoofing (E/D) [D]?
Neighbor Priority (H/M/L) [M]?
```

For a description of the parameters listed here, see the **add tcp** command on page 525.

explorer-limit

Allows you to specify limits on the number of SNA and NetBIOS explorer frames to be simultaneously queued to be sent to a DLSw partner.

Example: `set explorer-limit`

```
Max SNA explorers per transport queue (0-1000)[100]?
Max NB explorers per transport queue (0-1000)[100]?
DLSW explorer limit values have been set.
```

Max SNA explorers per transport queue

The maximum number of SNA explorer frames that can be simultaneously queued to be sent to an individual DLSw partner.

Max NB explorers per transport queue

The maximum number of NetBIOS explorer frames that can be simultaneously queued to be sent to an individual DLSw partner.

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.

mac-list
Modifies the local MAC address list exclusivity.

Example: `set mac-list`

```
Local MAC list exclusivity (E=exclusive, N=non-exclusive) [N]? e

MAC list parameter set.

For the change to take effect, restart or commit the change using
  't 5' : 'SET MAC-LIST'.
```

Local MAC list exclusivity
Indicates whether the local MAC list is exclusive (represents all MAC addresses to be accessible via this DLSw) or non-exclusive (represents a set of MAC addresses to be accessible via this DLSw).

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 monitoring 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 session and explorer priority of Critical, which corresponds by default to a message allocation value of 4. If you then assign NetBIOS session and explorer 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 might also want to assign different priorities to your SNA and NetBIOS session and explorer traffic.

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`

```
Default priority for SNA DLSw session traffic (C/H/M/L) [M]?
Default priority for NetBIOS DLSw session traffic (C/H/M/L) [M]?
Default priority for SNA DLSw explorer traffic (C/H/M/L) [M]?
Default priority for NetBIOS DLSw explorer traffic (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]? 516
```

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).

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 (0, 1.0-5.0 secs. Decimal) [2.0]?
Neighbor Inactivity Termination Timer (0-255 minutes) [5]?
Time to delay sending test response (0.0-5.0 secs. Decimal) [0.0]?
DLSw timer values have been set.
```

Database age timeout
Specifies how long to hold unused DLSw 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
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.

A value of zero indicates that the neighbor priority feature is not to be used. There will be no cached DLSw peer information for each MAC address. A CANUREACH is always sent and the first DLSw peer to send an ICANREACH is used (regardless of its priority).

Inactive neighbor termination timer
The time DLSw waits before taking down an inactive (zero sessions) passive TCP connection.

Delay sending TEST response
The amount of time to wait after completing exploration for a MAC address before sending the TEST response. This is useful if there are two DLSw 2210s on the same bridged LAN capable of reaching the same MAC address via DLSw peers. If one DLSw 2210 is preferable, the TEST response can be delayed at the less preferable 2210.

DLSw Monitoring Commands

This section describes the DLSw monitoring commands. These commands take effect immediately but do not become part of router's SRAM configuration. Thus, while monitoring commands enable 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.

Accessing the DLSw Monitoring Environment

To enter the DLSw monitoring 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 Console

For help using the Command Line Interface, press ESCAPE, then '?'

* talk 5
+ protocol dls
DLS>
```

DLSw Monitoring Commands

This section describes the DLSw Monitoring commands listed in Table 36. Use these commands to gather information from the database.

Table 36. DLSw Monitoring Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
Add	Dynamically adds an SDLC link station, a TCP neighbor IP address, a QLLC station or destination, cache entries, MAC address list entries, circuit priority overrides, or MAC cache explorer overrides 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 51 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, a QLLC station or destination, cache entries, mac address list entries, circuit priority overrides, and MAC cache explorer overrides.
Disable	Dynamically disables the LLC switching function, an SDLC link station, dynamic neighbors, a QLLC station or interface, , or use of local and remote mac address lists.
Enable	Dynamically enables the LLC switching function, an SDLC link station, dynamic neighbors, a QLLC station or interface, or use of local and remote mac address lists.
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, sessions for QLLC destinations, stations, and interfaces, cache entries, and mac address list entries. 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, parameters for QLLC operation, or mac address list related parameters.
Test	Test particular MAC addresses against the current MAC address cache and MAC address lists.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Add

Use the **add** command to dynamically configure an SDLC link station, a TCP neighbor IP address, a QLLC station or destination, cache entries, mac address list entries, circuit priority overrides, and MAC cache explorer overrides without affecting the SRAM configuration.

Syntax:

add	cache-entry
	explorer-override

mac-list

priority

qllc...

sdlc

tcp

For examples and field descriptions, see the **add** command in the configuration chapter at “Add” on page 516.

BAN

Use the **ban** command to access the BAN (Boundary Access Node) monitoring prompt. Enter the **ban** command from the `DLS>` prompt.

Syntax:

ban

Once you access the BAN monitoring prompt, you can begin entering specific monitoring commands. See “Chapter 4. Using the Boundary Access Node (BAN) Feature” on page 51 for an explanation of the BAN monitoring 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-FE), or one of the following:
  'SNA', 'NB', or LNM [0]? 04
SAP(s) 4 closed on interface 1
```

(An explanation of the **close-sap** parameters can be found on page 526.)

Delete

Use the **delete** command to dynamically remove an SDLC link station, DLSw session, a TCP neighbor IP address, a QLLC station or destination, cache entries, mac address list entries, circuit priority overrides, or MAC cache explorer overrides without affecting the DLSw SRAM configuration. Use of this command also terminates any existing session.

Syntax:

delete

cache-entry

dls

explorer-override

mac-list

priority

qllc...

sdlc

tcp

cache-entry

Deletes the specified cache entry

Example: `delete cache-entry`

```
Enter MAC Address [400000000000]? 10005a123456
MAC 10005A123456 / IP address 128.185.122.234 configured cache entry deleted.
```

dls Removes a currently active DLSw session.

Example: `delete dls`

```
Session identifier [1]?
```

explorer-override

Removes the specified MAC cache explorer override entry.

Example: delete explorer-override

```
Enter explorer override record number [1]?
Explorer override record has been deleted.
```

mac-list

Deletes the specified mac address list entry.

Example: `delete mac-list`

```
Enter mac list record number [1]?

Local MAC list entry 10005A000000 / FFFFFF000000 has been deleted.
```

priority

Deletes the specified circuit priority override entry.

Example: `delete priority`

```
Enter circuit priority override record number [1]?
Circuit priority override record has been deleted.
```

qllc Removes support for a QLLC destination or station. If you delete a station that currently is active, DLSw verifies that you wish to take down the connection before doing so. Deleting a destination has no effect on existing connections.

Syntax:

qllc 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 or 'sw' (switched dial-in) [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 or “sw” for a switched SDLC call-in circuit.

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, a QLLC station or interface, or use of local and remote mac address lists without affecting the DLSw SRAM configuration. Disabling the ***entire*** DLSw function from monitoring is not supported.

Syntax:

disable	dynamic-neighbors
	llc
	mac-list
	qllc...
	sdlc

(Examples using parameters of the **disable** command can be found beginning on page 528.)

Enable

Use the **enable** command to dynamically enable the LLC disconnect function, an SDLC link station, dynamic neighbors, a QLLC station or interface, or use of local and remote address lists without affecting the DLSw SRAM configuration.

Syntax:

enable	dynamic-neighbors
	llc
	mac-list
	qllc...
	sdlc

(Examples using parameters of the **enable** command can be found beginning on page 530.)

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 25. Using DLSw” on page 481 .

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. For additional information and an example, see “Chapter 25. Using DLSw” on page 481.

Syntax:

leave-group

Example:

```
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, QLLC stations, destinations and interfaces, configured cache entries, MAC address list entries and MAC cache explorer overrides.

Syntax:

list
- dls...
- explorer-override
- groups...
- llc2...
- mac-list
- 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           141312
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 SessionAlive Spoofing DISABLED
Dynamic Neighbor Priority              MEDIUM
QLLC base source MAC address           40514C430000
QLLC maximum dynamic addresses         64
Type of local MAC list                 NON-EXCLUSIVE
Use of local MAC list is               ENABLED
Use of remote MAC list is              ENABLED
SNA explorer limit                     100
NetBIOS explorer limit                 100
```

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 NetBIOS SessionAlive Spoofing
Whether NetBIOS SessionAlive I-frames are forwarded to DLSw peers established 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 sessions 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 sessions 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 sessions 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 sessions 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 sessions 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 sessions 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
```

list dls cache all
The **list dls cache all** 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 dls cache all`

```
    Mac Address   Entry Type  Secs to live      IP Address(es)   LFSize
1.  10005A123456  PERMANENT   (not being timed) 128.185.236.84   0
2.  10005A789ABC  STATIC      (not being timed) 128.185.236.84   0
3.  10005AF1809B  DYNAMIC        810            128.185.236.84   2052
4.  10005AF181A4  DYNAMIC        1170           128.185.236.84   2052
5.  400000000088  DYNAMIC        1170           128.185.236.84   2052
```

dls cache config

Displays DLSw configured MAC cache entries.

Example: `list dls cache config`

```
Mac Address   IP Address       Source     Last Mod
------------  ---------------  ---------  ---------
10005A123456  128.185.236.84   PERMANENT  UNCHANGED
10005A789ABC  128.185.236.84   STATIC     ADDED
```

list dls cache range

Displays information for a specified range of cache entries.

Example: `list dls cache range`

```
Start [1]?
Stop ]1]? 20
       Mac Address   Entry Type  Secs to live      IP Address(es)   LFSize
   1.  10005A123456  PERMANENT   (not being timed) 128.185.236.84   0
   2.  10005A789ABC  STATIC      (not being timed) 128.185.236.84   0
   3.  10005AF1809B  DYNAMIC        810            128.185.236.84   2052
   4.  10005AF181A4  DYNAMIC        1170           128.185.236.84   2052
   5.  400000000088  DYNAMIC        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 acknowledgment by the attached end station.

explorer-override
Lists the configured MAC cache explorer overrides.

Example: list explorer-override

```
    Explorer      Explorer      DB Age    Wait ICR  Nbr Pri   TESTrsp  Forwarding
ID  MAC Value     MAC Mask      Timeout   Timeout   Timeout   Delay    Explorers
--  ------------  ------------  --------  --------  --------  -------  ---------
 1  400031740000  FFFFFFFF0000  DISABLED    20      DISABLED  0.0      AllPartners
 2  10005A000000  FFFFFF000000   1200       20      2.0       0.0      NoPartner
```

mac-list all
Displays all local and remote MAC address list entries.

Example: `list mac-list all`

```
MAC Value     MAC Mask      IP Address
------------  ------------  ---------------
10005AF17F23  FFFFFFFFFFFF  Local
10005AF1809B  FFFFFFFFFFFF  128.185.236.84
4000189E2000  FFFFFFFFF000  128.185.236.84
4000189E3000  FFFFFFFFF000  Local
```

mac-list config
Displays all locally configured MAC address list entries.

Example: `list mac-list config`

```
Entry  Mac Value     MAC Mask      Source     Last Mod
-----  ------------  ------------  ---------  ---------
    1  10005AF17F23  FFFFFFFFFFFF  STATIC     UNCHANGED
    2  4000189E3000  FFFFFFFFF000  STATIC     UNCHANGED
```

mac-list local
Displays all active local MAC address list entries.

Example: `list mac-list local`

```
LOCAL MAC List
Type of MAC List (active) ....... EXCLUSIVE
Type of MAC List (pending) ...... EXCLUSIVE


MAC Value     MAC Mask
------------  ------------
10005AF17F23  FFFFFFFFFFFF
4000189E3000  FFFFFFFFF000
```

mac-list remote
Displays ALL active remote MAC address list entries for a specific DLSw peer.

Example: `list mac-list remote`

```
Enter the DLSw neighbor IP Address [0.0.0.0]? 128.185.236.84

Partner IP Address ............. 128.185.236.84
Type of MAC List ............... EXCLUSIVE
Use of remote MAC lists ......... ENABLED


MAC Value     MAC Mask
------------  ------------
10005AF1809B  FFFFFFFFFFFF
4000189E2000  FFFFFFFFF000
```

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     Keep-    SesAlive
Mcast IP Addr   Role      CST Bufsize Bufsize Segsize Alive    Spoofing Priority
--------------- --------- --- ------- ------- ------- -------- -------- --------
224.0.10.0      READWRITE  p    5120    5120    1024  DISABLED DISABLED MEDIUM
Group  2        PEER       p    5120    5120    1024  DISABLED 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.

SesAlive Spoofing
Displays the status of the NetBIOS SessionAlive Spoofing 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 groups 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(s)
    0        0  4
    1        0  4  8  C
```

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.

llc2 sessions ban

Displays current information for LLC2 sessions involving the BAN function.

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
```

priority

Displays DLSw circuit priority information.

Example: `list priority`

```
Default priority for SNA DLSw session traffic is           HIGH
Default priority for NetBIOS DLSw session traffic is       MEDIUM
Default priority for SNA DLSw explorer traffic is          MEDIUM
Default priority for NetBIOS DLSw explorer traffic is      LOW

Message allocation by C/H/M/L priority is   4/3/2/1
Maximum frame size for NetBIOS is           516

    Source/  SAP      MAC Address                   Session   Explorer
ID  Dest     Range    Range                         Priority  Priority
--  -------  -------  ---------------------------   --------  --------
 1  Source:  00 - FE  000000000000 - FFFFFFFFFFFF   CRITICAL  MEDIUM
    Dest  :  00 - 0C  10005A000000 - 10005AFFFFFF
 2  Source:  04 - 04  400031740000 - 40003174FFFF   CRITICAL  MEDIUM
    Dest  :  00 - FE  000000000000 - FFFFFFFFFFFF
```

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 25. Using DLSw” on page 481. 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 25. Using DLSw” on page 481.

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 monitoring 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 acknowledgment.

RESET_PEND
DLSw has requested a PVC reset or SVC clear call to the QLLC station and is awaiting acknowledgment.

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 25. Using DLSw” on page 481. 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    Source SAP/MAC    Dest SAP/MAC      PU  Blk/Idnum  PollType
1    C1     Enabled   04 4000103D01C1   00000000000000    2   000/00000  TEST
1    C2     Enabled   04 4000103D01C2   00000000000000    2   000/00000  SNRM
3    FF(sw) Enabled   04 4000103D01D2   04 400000000003   2   000/00000  TEST
```

sdlc sessions

Displays information about all SDLC dls sessions within the router.

Example: `list sdlc sessions`

```
    Net  Address   Source SAP/MAC    Dest SAP/MAC     PU    OutQ   State
1.   1    C1       04 4000103D01C1   00 000000000000  2      0    NET_DOWN
2.   1    C2       04 4000103D01C2   00 000000000000  2      0    NET_DOWN
```

Because 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 monitoring 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]? 128.185.236.84
Vendor ID:                  10005A
Vendor product version:     IBM 2210 Nways MRS 5765-B86 Feature 5045 V3 R2
Initial pacing window:      12
Preferred TCP connections:  1
Supported SAPs:             00 04 08 0C F0
MAC List Exclusivity:       Complete List
MAC List:                   08005ACEEA1C [FFFFFFFFFFFF]
                            4000189E2000 [FFFFFFFFF000]
NetBIOS Exclusivity:        (not supplied)
NetBIOS Name List:          (none supplied)
Multicast Version:          01
IBM CST:                    Passive Transport
IBM Multicast:              Available
IBM Capex Correlator:       19660
```

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 2210 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.

MAC List Exclusivity
Indicates whether the MAC address list sent by this neighbor is to be considered as a complete or partial list of the MAC addresses that are local to that neighbor. A response of “(not supplied)” indicates that this neighbor did not send a MAC address list as part of its capabilities.

MAC List
Displays all MAC list values and masks that this neighbor sent in its MAC address list. A response of “(none supplied)” indicates that this neighbor did not send a MAC address list as part of its capabilities.

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 “(not 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.

Multicast Version
Indicates what version of Multicast this neighbor supports as defined by the AIW standard. A response of *not supplied* indicates that this neighbor did not send a Multicast Version as part of its capabilities.

IBM CST
Indicates which IBM Connectivity Setup Type (CST) this neighbor has configured. A response of *not supplied* indicates that this neighbor did not send an IBM CST as part of its capabilities.

IBM Multicast
Indicates whether or not specific IBM Multicast functions are available at this neighbor. A response of indicates *not supplied* that this neighbor did not send an IBM Multicast as part of its capabilities.

IBM Capex Correlator
Indicates the value of the last IBM Capex Correlator received from

this neighbor. A response of *not supplied* indicates that this neighbor did not send an IBM Capex Correlator as part of its capabilities.

tcp config

Displays the configuration parameters for all configured TCP connections to peer DLSw routers.

Example: `list tcp config`

```
                     Xmit    Rcv     Max     Keep-    SesAlive
Neighbor        CST Bufsize Bufsize Segsize Alive    Spoofing Priority
--------------- --- ------- ------- ------- -------- -------- --------
128.185.236.84   p    5120    5120    1024  DISABLED 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 may consists of 1 or 2 TCP connections) to this neighbor. The valid states are:

DOWN

TCP session not established; no capabilities exchanged (passive partners only).

CAPEX FAILED

Attempt to exchange capabilities failed; TCP session is brought down.

Unicasting

TCP session not established; capabilities successfully exchanged (passive partners only) (ready for DLSw explorer traffic)

PENDING R/W

This 2210 has attempted to establish a TCP session with neighbor.

RD EST/WR PEND

TCP session between neighbor and this 2210 is active, but TCP session between this 2210 and neighbor is not active.

RD EST/WR PEND

TCP session between this 2210 and neighbor is active, but TCP session between neighbor and this 2210 is not active.

CAPEX PENDING

TCP session established; in process of exchanging capabilities.

ESTABLISHED

TCP session established; capabilities exchanged (ready for use for DLSw sessions).

CLOSING

Bringing down TCP session.

RECONNECT WAIT
TCP session not established; waiting for timer to expire in order to attempt to re-establish TCP session.

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          20    seconds
Join Group Interval               900   seconds
Neighbor priority wait timer      2.0   seconds
Neighbor Inactivity Timer           5   minutes
Time to delay sending test resp.  0.0   seconds
```

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.

Delay sending TEST response
The amount of time to wait after completing exploration for a MAC address before sending the TEST response

NetBIOS

Displays the NetBIOS monitoring prompt.

Syntax:

netbios

Example: `netbios`

```
NetBIOS Support User Configuration
NetBIOS config>
```

For a description of NetBIOS commands, see “Chapter 8. Configuring and Monitoring NetBIOS” on page 153.

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 537 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, parameters for QLLC operation, mac address list related parameters, and circuit priority parameters without affecting the DLSw SRAM configuration.

Syntax:

set dynamic-tcp
explorer-limit
llc2
mac-list
memory
priority
qllc
timers

dynamic-tcp

Enables 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]?
Enable/Disable NetBIOS SessionAlive Spoofing (E/D) [D]?
Neighbor Priority (H/M/L) [M]?
```

For a description of these parameters, see the **add tcp** command in “Chapter 25. Using DLSw” on page 481.

explorer-limit

Allows you to specify limits on the number of SNA and NetBIOS explorer frames to be simultaneously queued to be sent to a DLSw partner.

Example: `set explorer-limit`

```
Max SNA explorers per transport queue (0-1000)[100]?
Max NB explorers per transport queue (0-1000)[100]?
DLSw explorer limit values have been set.
```

Max SNA explorers per transport queue
The maximum number of SNA explorer frames that can be simultaneously queued to be sent to an individual DLSw partner.

Max NB explorers per transport queue
The maximum number of NetBIOS explorer frames that can be simultaneously queued to be sent to an individual DLSw partner.

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 539).

mac-list

Allows you to set the local MAC address exclusivity. This command also enables you to commit all changes that have been previously made through the following monitoring commands:

- enable mac-list local
- enable mac-list remote
- disable mac-list local
- disable mac-list remote
- add mac-list

- delete mac-list
- set mac-list

As a result of this command, a new run-time capabilities will be sent to all DLSw peers to communicate the new information.

Syntax: mac-list

Example: **`set mac-list`**

```
Local MAC list exclusivity (E=exclusive, N=non-exclusive) [N]? e

MAC 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 MAC-LIST LOCAL
  - changes made using ENABLE MAC-LIST REMOTE
  - changes made using DISABLE MAC-LIST LOCAL
  - changes made using DISABLE MAC-LIST REMOTE
  - changes made using ADD MAC-LIST
  - changes made using DELETE MAC-LIST
  - changes made using SET MAC-LIST

Would you like to commit the MAC list changes? [No]: y

Use of local MAC list remains    ENABLED.
Use of remote MAC list remains    ENABLED.
Type of local MAC list has changed from NON-EXCLUSIVE to EXCLUSIVE    .
Entry added temporarily:   08005ACEE5D9 / FFFFFFFFFFFF.
Entry added temporarily:   4000189E3000 / FFFFFFFFF000.
Would you still like to commit the MAC list changes? [No]: y


MAC address list changes have been committed.
```

memory

This command enables 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 540.

priority

Allows you to specify the circuit priorities to use for SNA circuits and NetBIOS circuits. You can configure circuit priorities of Critical, High, Medium, or Low (in descending order from Critical to Low).

This command also enables 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 541.

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 542.

Test

Use the **test** command to perform tests against the currently active MAC address cache and MAC address list.

Syntax:

test cache
mac-list

cache Allows you to determine how a frame destined for a particular MAC address would be forwarded based upon current cache and DLSw peer information.

Syntax: cache

Example: `test cache`

```
MAC address to be tested [000000000000]? 10005af1809b

Enter largest frame size to perform test against [2052]?

Destination MAC address being tested .... 10005AF1809B

MAC cache entry found:
  Entry type = DYNAMIC

Handling of SNA explorer SSP messages ....
  Explorer SSP message not sent (information found locally).

Handling of SNA circuit setup SSP messages ....
  Circuit Setup SSP message would be forwarded to 128.185.236.84

Handling of NetBIOS explorer SSP messages ....
  Explorer SSP message would be broadcast.
  How explorer destined for this MAC address is forwarded to DLSw partners
.....
    Send to all partners with non-exclusive mac address lists.
  There are currently no DLSw partners to forward the explorer to.

Handling of NetBIOS circuit setup SSP messages ....
  No currently known transport that can support circuit setup for given lfsize.
```

mac-list
Allows you to match a given MAC address against all currently active MAC address list entries (local and remote). This is useful in resolving MAC address list conflict problems.

Syntax: mac-list

Example: `test mac-list`

```
MAC address to be tested [000000000000]? 10005af1809b

Destination MAC address being tested .... 10005AF1809B


MAC address value  MAC address mask   IP Address
-----------------  -----------------  ---------------
10005AF1809B       FFFFFFFFFFFF       128.185.236.84
```

DLSw Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

Data Link Switch (DLSw) supports the CONFIG (Talk 6) **delete interface** command with the following consideration:

DLSw SAPs, SDLC stations, and QLLC stations are deleted for a deleted interface.

GWCON (Talk 5) Activate Interface

Data Link Switch (DLSw) supports the GWCON (Talk 5) **activate interface** command with the following consideration:

DLSw must be globally enabled for DLSw SAPs, SDLC stations, and QLLC stations to be activated for a new interface.

All Data Link Switch (DLSw) interface-specific commands are supported by the GWCON (Talk 5) **activate interface** command.

GWCON (Talk 5) Reset Interface

Data Link Switch (DLSw) supports the GWCON (Talk 5) **reset interface** command with the following considerations:

- DLSw must be globally enabled for DLSw SAPs, SDLC stations, and QLLC stations to be modified for a reset interface.
- The DLSw SAPs for the reset interface are closed and re-opened, but current DLSw sessions are not affected.
- The SDLC stations and QLLC stations associated the interface being reset are deleted and then restarted causing the associated DLSw sessions to be taken down and re-established.

All Data Link Switch (DLSw) interface-specific commands are supported by the GWCON (Talk 5) **reset interface** command.

GWCON (Talk 5) Temporary Change Commands

Data Link Switch (DLSw) supports the following GWCON commands that temporarily change the operational state of the device. These changes are lost whenever the device is reloaded, restarted, or you execute any dynamically reconfigurable command.

The following commands do not affect any DLSw functions already active at the time of the command unless otherwise noted. The changes do apply to all subsequent DLSw operations, TCP sessions, DLSw sessions, and DLSw traffic.

Commands
GWCON, protocol dls, add cache-entry
GWCON, protocol dls, add explorer-override
GWCON, protocol dls, add mac-list
GWCON, protocol dls, add priority-override
GWCON, protocol dls, add qllc destination

GWCON, protocol dls, add qllc station
GWCON, protocol dls, add sdlc
GWCON, protocol dls, add tcp
GWCON, protocol dls, close-sap
GWCON, protocol dls, delete cache-entry
GWCON, protocol dls, delete explorer-override
GWCON, protocol dls, delete mac-list
GWCON, protocol dls, delete priority-override
GWCON, protocol dls, delete qllc destination
GWCON, protocol dls, delete qllc station **Note:** This command takes down all associated DLSw sessions.
GWCON, protocol dls, delete sdlc **Note:** This command takes down all associated DLSw sessions.
GWCON, protocol dls, delete tcp **Note:** This command takes down all associated TCP and DLSw sessions.
GWCON, protocol dls, disable dynamic-neighbors
GWCON, protocol dls, disable forwarding-explorers
GWCON, protocol dls, disable IPv4 DLSw Precedence
GWCON, protocol dls, disable LLC Disconnect
GWCON, protocol dls, disable mac-list
GWCON, protocol dls, disable qllc callin
GWCON, protocol dls, disable qllc station
GWCON, protocol dls, disable sdlc
GWCON, protocol dls, enable dynamic-neighbors
GWCON, protocol dls, enable forwarding-explorers
GWCON, protocol dls, enable IPv4 DLSw Precedence
GWCON, protocol dls, enable LLC Disconnect
GWCON, protocol dls, enable mac-list
GWCON, protocol dls, enable qllc callin
GWCON, protocol dls, enable qllc station
GWCON, protocol dls, enable sdlc
GWCON, protocol dls, join-group
GWCON, protocol dls, leave-group
GWCON, protocol dls, open-sap
GWCON, protocol dls, set dynamic-tcp
GWCON, protocol dls, set explorer-limit
GWCON, protocol dls, set llc2
GWCON, protocol dls, set mac-list
GWCON, protocol dls, set memory
GWCON, protocol dls, set priority
GWCON, protocol dls, set qllc
GWCON, protocol dls, set timers

Non-Dynamically Reconfigurable Commands

The following table describes the Data Link Switch (DLSw) configuration commands that cannot be dynamically changed. To activate these commands, you need to reload or restart the device.

Commands
CONFIG, protocol dls, disable dlsw
CONFIG, protocol dls, enable dlsw
CONFIG, protocol dls, set cache
CONFIG, protocol dls, set maximum
CONFIG, protocol dls, set srb

Chapter 27. Using 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:

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 ATM addresses or physical medium access control (MAC) addresses. Given only the network layer address of the destination system, ARP locates the ATM address or 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 49 on page 572 illustrates how ARP works.

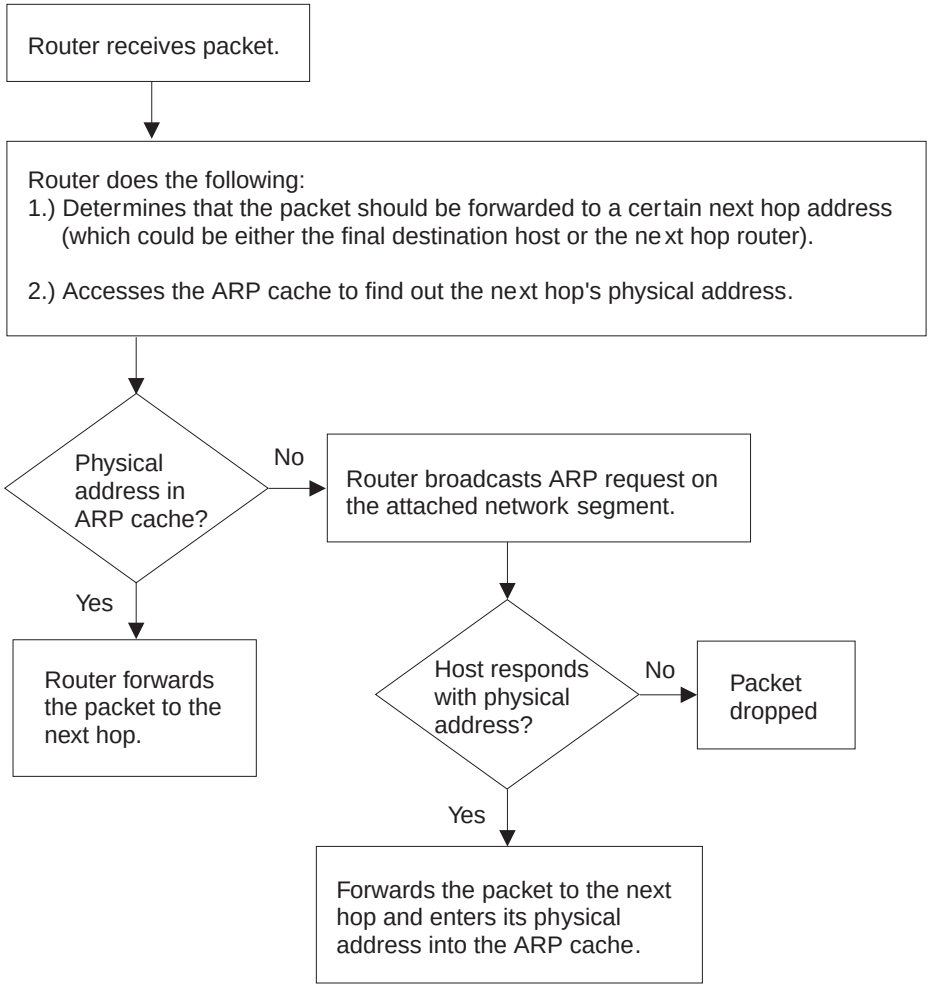


Figure 49. ARP Address Resolution Broadcast

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 Components” on page 574.

Inverse ARP Overview

Inverse ARP, described in RFC 1293/2390, 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; 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 49 on page 572).

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 Configuring and Monitoring Frame Relay Interfaces in *Software User's Guide*.

Classical IP and ARP Over ATM (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" on page 574.

RFC 2225 is an extension to RFC 1577 that changes the client registration mechanism and allows for multiple ATM ARP servers. The 2210 supports both RFC 1577 and 2225 behaviors.

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*

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 2210 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 2210 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 service is defined per LIS. The ARP service may be one ARP Server or several distributed ARP Servers per LIS. The service 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 service also responds to ATMARP requests for ATM addresses corresponding to IP addresses requested by the client. Finally, the ARP service 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 and register with an 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.

RFC 2225 clients register with the ARP Server by sending an ARP request with the source and target protocol addresses set to the same value. The registration process is completed successfully when the clients receive an ARP reply for this request.

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.

Refer to the *Software User's Guide* for additional information about ATM Virtual Interfaces.

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).

Notes:

1. ARP entries are always refreshed if an ARP or InARP is received from that client.
2. The **Auto-refresh** defaults to *No* for a client and to *Yes* for a server.

RFC 2225 clients are required to re-register with the ARP Server every 15 minutes by sending an ARP request for its own IP address. The refresh time is configurable, but RFC 2225 specifies that 15 minutes is the re-registration interval.

RFC 2225 servers are not required to refresh client entries using InARPs. It is the responsibility of the client to re-register. The default value for server **auto-refresh** remains *Yes* so that the server is compatible with RFC 1577 clients. If the LIS has only RFC 2225 clients, **auto-refresh** can be set to *No* on the servers.

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, automatically creating a client. 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.

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 (ESI/Selector combination) 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 signaling 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 signaling 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 ARP 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, three pieces of information are required:

1. The IP address and Subnet mask. (**add address**)
2. The ATM address(es) of the ARP Server (or distributed ARP Servers). (**add arp-server**)
3. Configure the ARP client, and reply *No* when asked whether the client is also a 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.

Multiple ARP Servers can be defined for a given LIS for backup purposes. If the primary ARP Server goes down, the client can switch to a backup ARP Server to avoid a single point of failure. The client will be able to switch back to its primary ARP Server as soon as the primary ARP Server resumes service. The first configured ARP Server ATM address will be chosen as the default primary ARP Server for a given LIS. You can change the primary ARP Server using the **reorder** command from the `ARP Config>` command prompt.

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-arp-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-arp-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 (.).

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 (:).

   This applies to addresses entered for ATM, LAN emulation, and Classical IP & ARP over ATM.

Example of IP Address:

```
01.255.01.00
```

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
```

Classical IP Redundancy Overview

The ARP Server redundancy has two devices. One serves as the primary ARP server and the other serves as the backup to the primary. Classical IP Redundancy allows you to specify in your configuration which device will act as the primary server, and which device will act as the secondary server (backup server). In this type of redundancy, the primary server is configured to service and route for a given LIS. When the primary fails, the backup registers using the primary's ATM address and takes over as the ARP Server. It can also act as the redundant default IP gateway, thereby taking over as the server and the router for that LIS. So, when everything is operational, the primary has two IP addresses on the LIS (a client IP address and a gateway IP address), and the backup has a single client IP address on the LIS. When the primary fails, the primary will obviously cease to have any appearance on the LIS, and the backup will have two IP addresses on the LIS (its original client IP address, and its newly obtained redundancy default IP gateway address). The backup will also assume the role of the ARP Server for that LIS (by taking over the ATM address of the primary).

ARP Server redundancy configuration will give you the capability to control which device acts as primary, and which one acts as the secondary. This allows you to effectively balance the load on your ARP Servers while providing backup. For example, you may want a device to be the primary ARP Server for six LISs and to be the secondary for six other LISs. And you may want a second device to be the secondary for the first six LISs and the primary for the other six LISs. The resulting configuration will have 12 LISs, six being served by one device, and six being served by the other. If either device goes down, the other device will take over the server role for all 12 LISs.

It should be noted that there will be two ATM addresses associated with the ATM endpoint. One ATM address will be the real ATM address, and the other will be a special redundancy ATM address, called the redundancy address. The redundancy address is always registered. The redundancy channel is established between the primary's and secondary's redundancy addresses. The redundancy addresses are used for redundancy activity only. The real addresses are used for the exchange of IP information.

In ARP Server redundancy, when configured as a primary, the primary entity will *always* try to register its real ATM address until it is successful. The primary will also attempt to place a call for the Redundancy channel to the secondary.

Notes:

1. The ARP Server redundancy requires that clients on the LIS be able to associate more than one IP address with a single VCC.
2. The primary and backup must be attached to the same ATM switch.

The following steps describe the ARP Server redundancy configuration process for a non-distributed ARP Server LIS:

1. Configure an ARP Client/Server pair on one device. This will be the primary ARP Server.
2. Configure an ARP Client only on the other device. This will provide the backup ARP Server function.
3. Use different ATM addresses and different IP addresses for the primary ARP Client/Server pair and the ARP Client providing the backup ARP Server function (both IP addresses must be on the same LIS)

Note: Please see the sample configuration provided in “Sample ARP Configurations” on page 609 for more detail.

ARP Server Redundancy provides the capability of a backup server for 1577 clients. 2225 clients do not need ARP Server Redundancy because they are capable of switching to a backup ARP Server.

Distributed ARP Server Overview

The Distributed ARP Server allows you to maintain connectivity with a LIS in the event of an ARP Server failure. You can define as many distributed servers as you need per LIS (normally three to four are sufficient). The distributed servers can be located anywhere in your ATM network. They do not need to be meshed, but there must be some communication path from one to another.

An additional benefit of Distributed ARP Server is that the ATM ARP Service load can be distributed over many devices, allowing large LISs to be handled more efficiently.

Distributed ARP Servers on the same LIS must be configured with:

- The same Server Group ID (SGID)
- An ESI/Selector pair to be used to form the ATM address that other servers can use to contact this server to exchange ARP database information.
- The ATM addresses of the Directly Connected Servers (DCSs) with which the distributed ARP Server attempts to synchronize.

The Distributed ARP Server complies with the IETF draft ″Server Cache Synchronization Protocol (SCSP) – NBMA.″ SCSP is the general-purpose protocol for distributing server databases over ATM networks.

ATM ARP clients must be able to recognize when their connection to the ARP Server is not operational and they must be able to switch to an alternate server. RFC 2225 compliant clients satisfy this requirement.

Examples of Distributed ARP Servers

In Figure 50 on page 581, two ARP Servers are defined on one LIS. These ARP Servers are configured so that they duplicate each other’s ARP database. The

SCSP in each device is configured with the SCSP ATM address of the other ARP Server. The ARP Servers establish a private session in order to exchange database information. The SCSPs in the device interact with the ARP Server in the same device to get and report cache changes.

The ATM ARP client is configured to have two ARP Servers, one as the primary and one as the backup in case of failure. If the client loses contact with the primary, it will register with the backup. The backup will have the full ARP resolution database and will provide ARP resolution service to the client.

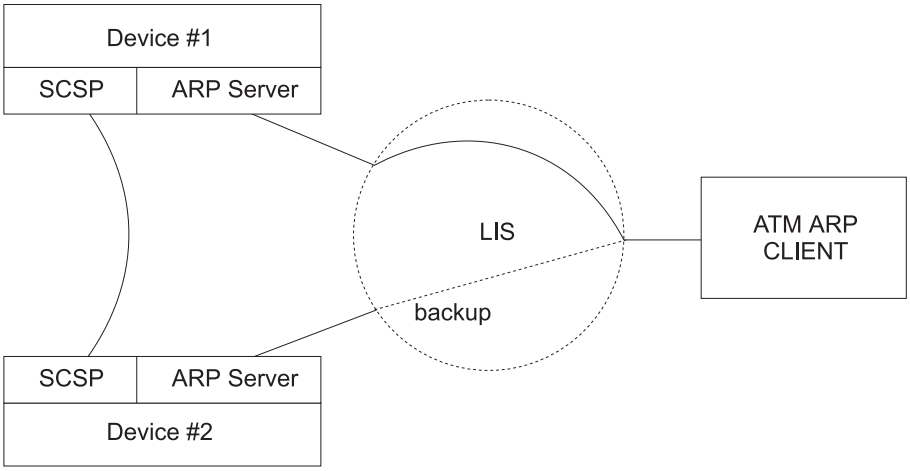


Figure 50. Simple Distributed ARP Server Configuration

In Figure 51 on page 582, three ARP Servers are configured on one LIS. Device 1 is configured with one Directly Connected Server (DCS), Device 2 is configured with two DCSs, and Device 3 is configured with one DCS.

Client 1 is configured with Device 1 as its ARP Server. Client 2 is configured with Device 3 as its primary server and Device 2 as its backup. With this configuration, Client 1 can get the address of Client 2 from Device 1 even though Client 2 is registered with Device 3. Likewise, Client 2 can get the address of Client 1 from Device 3.

If Device 3 fails, Client 2 can switch to Device 2 for ARP service with no loss of connectivity. If Device 1 fails, Client 1 will eventually lose connectivity with the LIS since it does not have a backup ARP Server configured. If Device 2 fails, redundancy is lost. In order for this configuration to retain full redundancy, the devices would need to be fully meshed.

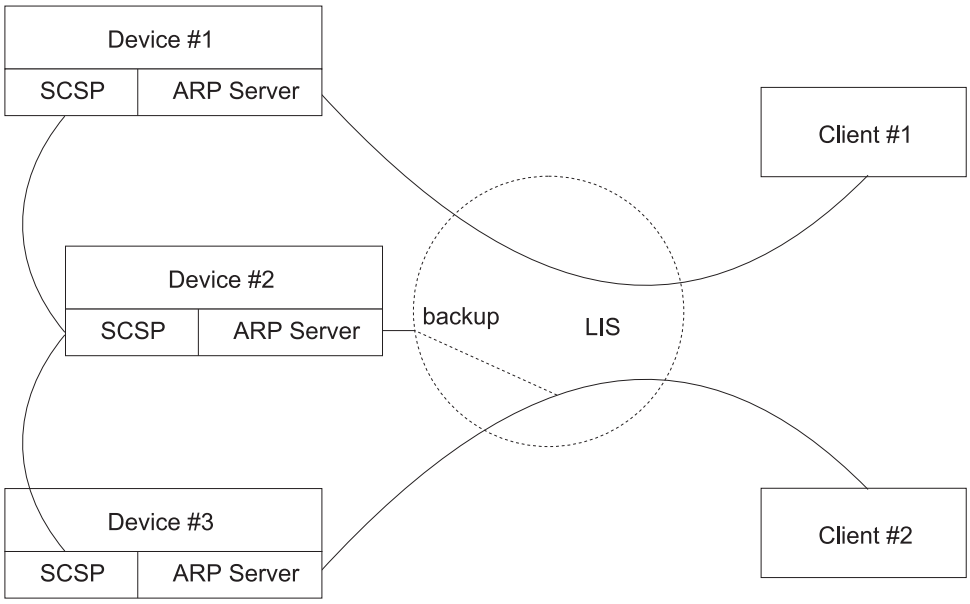


Figure 51. Distributed Configuration with Three ARP Servers

Peer Redundancy

The Distributed ARP Server allows you to provide alternate ARP server support for RFC 2225 clients. ARP Server Redundancy allows you to define a backup ARP server for RFC 1577 clients. The redundancy function and the Distributed ARP Server function can be defined on the same device. In this configuration, both primary and backup are defined as servers with SCSP enabled. When both are operating, they act as ARP servers with the full ARP database available. When the primary fails, the backup takes over the ATM address of the primary (and also keeps its own). Furthermore, if the backup fails, the primary can take over the ATM address of the backup and hence support its 1577 clients.

Note: Both primary and backup must be attached to the same ATM switch.
.

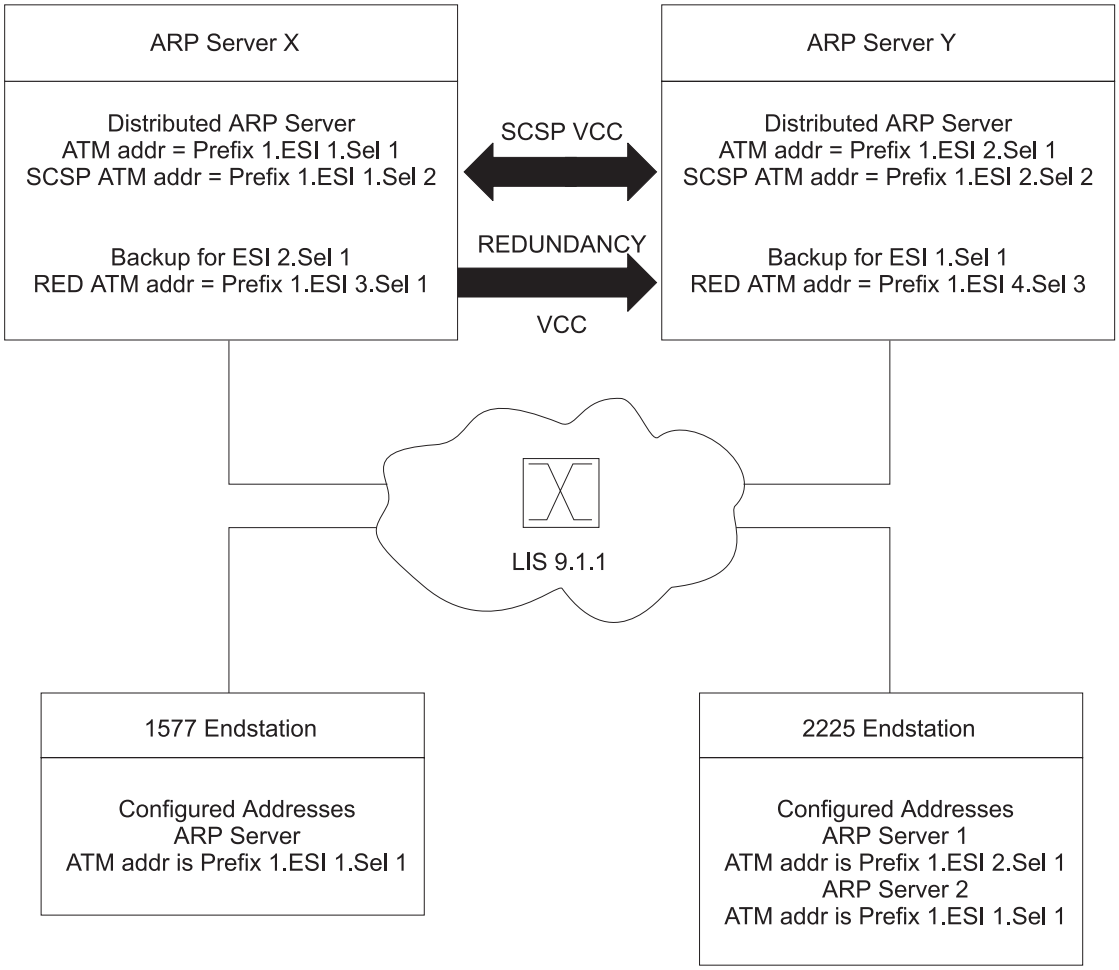


Figure 52. ARP Server Configuration with RFC 1577 and 2225 Clients

A server can be configured as both a redundant server and a distributed server. In Figure 52, distributed ATMARP servers are used for LIS 9.1.1. Both Server X and Server Y are actively serving different sets of ATMARP clients on LIS 9.1.1. Server X is serving the RFC 1577 compliant client and Server Y is the first choice of the 2225 client. The databases of the two ARP Servers are synchronized by the SCSP protocol. If Server Y were to fail, the 2225 client would use the next entry on its list of ATMARP Server ATM addresses and connect to Server X.

In order to provide ARP Server redundancy for the 1577-compliant client, Server X is designated as the primary ARP Server for ATM address Prefix 1.ESI 1.Sel 1 and Server Y is designated as the backup ARP Server for ATM address Prefix 1.ESI 1.Sel 1. The primary and backup ARP Servers provide redundancy support. If Server X fails and Server Y takes over, Server Y will register the ATM address Prefix 1.ESI 1.Sel 1 in addition to its other ATM addresses. Thus Server Y will simultaneously represent ATM address Prefix 1.ESI 1.Sel 1 and ATM address Prefix 1.ESI 2.Sel 1. If Server X subsequently recovers and reestablishes the Redundancy VCC to Server Y, Server Y will deregister ATM address Prefix 1.ESI 1.Sel 1 so that Server X can resume its role as one of the active ARP Servers for the LIS.

Peer redundancy depends on the existence of the redundancy channel between the pair of Distributed ARP Servers configured for redundancy. The ATM client or server

with the *bigger* redundancy ESI will initiate the redundancy channel with the partner. In this example, ESI 3 is greater than ESI 4, so Server X will initiate the call for the redundancy channel to Server Y.

Configuring Peer Redundancy

When creating a new configuration for ARP Server Redundancy, peer redundancy is automatically enabled if the ATM Classical IP client is configured as a distributed ARP Server.

When an existing configuration is used from a previous release, peer redundancy is not automatically enabled. In this case, in order to enable peer redundancy, use the **change redundancy** command to provide the partner server ESI and selector to each of the distributed ARP Servers and write the configuration to the device. Peer redundancy will then be enabled as long as the redundancy ARP Servers are distributed ARP Servers.

When distributed ARP Service for a client is disabled by changing the client configuration for a given IP address and there is a redundancy configuration, peer redundancy will be disabled. You will be prompted to verify the redundancy configuration for correctness when the distributed ARP service is disabled.

IPX and ARP Over ATM Overview (RFC 1483)

The 2210 uses LLC/SNAP encapsulation as specified by RFC 1483 to carry IPX traffic over ATM. 2210s (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, SVCs to IPX routers must be dedicated to IPX; they cannot be shared with other protocols, such as IP.

As with Classical IP, Quality of Server characteristics can be specified by configuring VCC traffic parameters such as Peak and Sustained Rates, and multiple circuits may be configured on a single ATM interface.

The 2210 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 2210 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 2210) 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 2210'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.

Using the ATM Virtual Interface facility, IPX is no longer limited to one address per physical ATM interface. Several ATM Virtual Interfaces can be defined on a physical ATM interface and one IPX address can be configured on each ATM Virtual Interface.

Refer to the *Software User's Guide* for additional information about ATM Virtual Interfaces.

Bridging over ATM Overview (RFC 1483)

Although bridging does not use ARP support, the implementation of bridging over native ATM shares some internal structures with ARP. In this relationship, ATM client and channel records for bridge ports may be displayed and modified (client record only). Note that addition and deletion of these records is done automatically when a bridge port is added or deleted on an ATM interface.

For more details on RFC 1483 support for bridging over ATM, please see "RFC 1483 Support for Bridging" on page 47.

Chapter 28. Configuring and Monitoring ARP

This chapter describes how to configure and monitor ARP protocol activity and how to use the ARP monitoring commands. It includes the following sections:

Accessing the ARP Configuration Environment

For information on how to access the ARP configuration environment, see "Getting Started" in *Software User's Guide*.

Use the following procedure to access the ARP *configuration* process.

1. At the OPCON prompt, enter **talk 6**. (For more details on this command, refer to "The OPCON Process and Commands"in *Software User's Guide*.) For example:

   ```
   * talk 6
   Config>
   ```

   After you enter the **talk 6** command, the CONFIG prompt (`Config>`) displays on the terminal. 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 describes the ARP configuration commands for non-ATM networks. Table 37 lists the ARP configuration commands. You can access ARP configuration commands at the `ARP config>` prompt.

Note: These commands are not used to configure ARP for Classical IP, IPX and Bridging on ATM interfaces. However, they can be used to configure ARP for ATM LAN Emulation clients.

Table 37. ARP Configuration Commands for non-ATM Networks Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
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	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

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 IP-encapsulation-type*

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

IP encapsulation type

This option appears only when you are configuring an Ethernet interface. You should select the IP encapsulation type that matches the kind of encapsulation that is performed by the host. Refer to "Configuring and Monitoring the Ethernet Network Interface" in the *Software User's Guide* for more information.

Valid Values: **Ether** or **ieee-802.3**

Default Value: **Ether**. The configured value is effective only if the encapsulation of the network interface is **both**. Otherwise, this value defaults to the encapsulation type that is configured on the network interface regardless of the value configured for the host.

Example: `add entry`

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
Mac Address []?
IP Encapsulation type (ETHER/IEE-802.3) [ETHER]?
```

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.

Syntax:

change entry *ifc# prot-type prot-addr MAC-addr IP-encapsulation-type*

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

IP encapsulation type

This option appears only when you are configuring an Ethernet interface. You should select the IP encapsulation type that matches the kind of encapsulation that is performed by the host.

Valid Values: **Ether** or **ieee-802.3**

Default Value: **Ether**. The configured value is effective only if the encapsulation of the network is **both**. Otherwise, this value defaults to the encapsulation type that is configured on the network interface regardless of the value configured for the host.

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.

Syntax:

delete entry *ifc# prot-type prot-addr*

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

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: permanent
Auto Refresh: disabled

Mac address translation configuration
IF #           Prot #         Protocol --> Mac Address  IP-Encap 1
 0               0           2.2.2.1 --> 0000C90932EF  Ether
```

1 IP encapsulation is displayed only for Ethernet interfaces that have been configured statically.

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** monitoring command, or until the router is restarted.

Valid Values: An integer number of minutes in the range of 0 to 65535

Default Value: 5 minutes

Example: `set refresh-timer 3`

ARP Over ATM Configuration Commands

This section describes the ARP Over ATM configuration commands. These commands apply to:

- Classical IP & ARP over ATM
- IPX over ATM
- 1483 bridging

Enter the commands at the `ARP Config>` prompt.

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.

Table 38. ARP Over ATM Configuration Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
List	Lists all (the current ARP over ATM configuration), lists ARP Servers (for IP only), or lists pvc-atm-arp-entries, svc-atm-arp-entries, atm-arp-client-configuration and redundancy.
Add	Adds an arp-server, atm-arp-client-configuration, pvc-atm-arp-entry, svc-atm-arp-entry or redundancy.
Change	Changes the atm-arp-client-configuration or redundancy.
Delete	Deletes an arp-server, atm-arp-client-configuration, pvc-atm-arp-entry, svc-atm-arp-entry or redundancy.

Table 38. ARP Over ATM Configuration Command Summary (continued)

Command	Function
Disable	Disables **auto-refresh** so that ARP entries will not be refreshed automatically.
Enable	Enables **auto-refresh** so that ARP entries will be refreshed automatically.
Set	Sets the **refresh-timer** for aging ARP entries.
Reorder	Selects the primary ARP Server from a given ARP Server list.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Add

Use the **add** command to add an arp-server, atm-arp-client-configuration, svc-atm-arp-entry, or redundancy.

Syntax:

add arp-server
atm-arp-client-configuration
pvc-atm-arp-entry
svc-atm-arp-entry
redundancy

arp-server private-nsapa *local-client-IP-address private-NSAP-address*

Adds an arp-server to the client specified or adds a Directly Connected Server (DCS) to an ARP Server.

If the IP address is a client only, the NSAP address is the address of a remote server. Multiple remote servers may be added per client. During initialization, the specified CIP client will place a call to an ARP Server and will use it as a mechanism for resolving IP addresses to ATM addresses.

If the IP address is a server, the NSAP address is the address of a DCS for the Distributed ARP Server. This address must match the SCSP ATM address for the DCS (not the client ATM address). Use the **List Server-Groups** monitoring command at the `SCSP>` command prompt under **t 5** to determine the SCSP ATM address of a server. See “Chapter 29. Monitoring the Server Cache Synchronization Protocol (SCSP)” on page 627 for additional information.

local-client-IP-address

This value will specify the IP address of the client or server.

Valid Values: Any valid IP address

Default Value: None

private-nsap-address

This field is the Private Network Specified Access Point Address that is the addressing format specified in the UNI Versions 3.0 and 3.1. When configuring DCS, this value is the ATM address of the DCS.

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 []? 39840f000000000000000000000410005a3345f3a0
```

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 or server, the refresh timeout and auto-refresh settings, how the ATM address for this client is determined, Distributed ARP Service parameters,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]? 1.1.1.2
This client is also a server? [Yes] yes
Refresh timeout (in minutes) [20]?
Enable auto-refresh? [Yes]:
Refresh by InAtmArp? [Yes]:
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) 222222222222
  (  4) 121212121212
  (  5) AAAAAAAAAAAA
Select ESI [1]?2
Use internally assigned selector? [Yes]: no
Selector Only, Page 00..FF [00] ? 11
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Server 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 Server 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]?
Participate in Server Synchronization [No]? yes
Server Group ID [1]?
Do you want to accept sessions from non-configured DCSs [Yes]?
Hello Interval [3]
Dead Factor [3]?
(  1) Use burned in ESI
(  2) 111111111111
(  3) 222222222222
(  4) 121212121212
(  5) AAAAAAAAAAAA
Server Synchronization ESI [2]?
Server Synchronization selector, Range 00..FF [00]? 12
Server Synchronization Max SDU size (bytes) [9188]?
Re-registration time with Arp Server (in minutes) [15]?


To enable or change multicast support,
please issue the ADD or CHANGE MULTICAST-SUPPORT command.
```

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 Server 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]?
```

Example for Bridging:

```
ARP config> add atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]? ASRT
Clients for this protocol can only be changed here.
Additions must be done under ASRT Config by adding a port.
```

Interface Number

Interface number assigned.

Valid values: Any interface on the device

Default value: 0

Protocol

Valid values: *IP*, *IPX*, **or** *ASRT*

Default value: IP

Client IP Address

Client IP Address (IP only). This should match the address configured using the **p IP** command.

Note: This value is also used for the LSID in the SCSP protocol.

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. ARP entries will age out after this number of minutes if they are not refreshed.

Valid Values: An integer number of minutes in the range of 0 to 65535

Default Value: 5 minutes

Enable auto-refresh

YES or NO.

If No, ARP entries will not be refreshed automatically.

Default Value: No for client, Yes for Server

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, and if auto-refresh is enabled, then AtmArp requests will be transmitted to the ARP Server to reconfirm the ARP entry.

Default Value: No for client, Yes for Server

Select ESI

Specifies whether a universally administered MAC address or a MAC address configured under the ATM interface configuration should be used as the End System Identifier component of the ATM address. This question is preceded by a list of valid ESIs from which the selection is to be made.

Valid Values: Any of the values listed in the menu preceding this question. Any 12 hexadecimal digits that have been defined as an ESI address using the **Add ESI** ATM Network configuration command.

Default Values: 1 (burned-in)

Use internally assigned selector

Use internally assigned selector.

Valid Values:Yes or No

Default Value: Yes

Selector

This is the last byte of the client's ATM address.

Valid Values: Any single octet value that has not been previously used and is within the range defined for the device.

Default Value: 0

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 accepted 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: Integer kbps in the range of 0 to line speed. If you enter 0, the parameter is set to the line speed.

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 to line speed of the ATM device. If you enter 0, the parameter is set to the line speed.

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 to control VCC PCR. If you enter 0, the parameter is set to the line speed.

Default Value: none

Use Best Effort Server 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 VCCs. 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 to control VCC PCR. If you enter 0, the parameter is set to the line speed.

Default Value: 0

Sustained Cell Rate of outbound Data VCCs (Kbps)
Specifies the Sustained Cell Rate (SCR) traffic parameter for the Data VCCs. (Sustained Cell Rate can be considered to reserved bandwidth.) This parameter is applicable only when Best Effort Server is not selected for Data VCCs.

Valid Values: An integer kbps in the range of 0 to PCR value for Data VCC. If you enter 0, the parameter is set to the line speed.

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 to Maximum interface SDU

Default Value: 9188

Participate in Server Synchronization
Specifies whether the ARP database for the LIS upon which this server resides will be distributed.

Valid Values: Yes or No

Default Value: No

Server Group ID

Specifies the value to identify this server group. This value must be unique for all Server Groups (of the ATMARP protocol type) within the ATM network. This value must be used for all servers in this server group (in this LIS).

Valid Values: 0 to 65535

Default Value:1

Accept sessions from non-configured DCS

Specifies whether this Local Server should accept connections from DCSs that are not explicitly configured.

Valid Values: Yes or No

Default Value: Yes

Hello Interval

Specifies the time in seconds between sending of Hello messages for this Local Server.

Valid Values: 0 to 65535

Default Value: 3

Dead Factor

Specifies the multiple of Hello Intervals after which this server should be considered down by directly connected servers (DCSs).

Valid Values: 0 to 65535

Default Value: 3

SCSP ESI

Specifies whether a universally administered MAC address or a MAC address configured under the ATM interface configuration should be used as the End System Identifier component of the SCSP ATM address. This question is preceded by a list of valid ESIs from which the selection is to be made.

Valid Values: Any of the values listed in the menu preceding this question. Any 12 hexadecimal digits that have been defined as an ESI address using the **Add ESI** ATM Network configuration command.

Default Value: ESI of the Client/Server

SCSP Selector

Specifies the selector to be associated with this SCSP Local Server. If the ***SCSP ESI*** defaults to the ESI of the ARP Server, this selector value must be different from the ARP Server selector.

This is the last byte of the ATM address of this local server.

Note: The SCSP ATM address can be shared among the CIP clients on the same interface. The ATM address can also be shared with other protocols using the 1483 ATM LLC. Refer to the chapter entitled “Using and Configuring ATM” in the *Software User’s Guide*. The SCSP ATM address cannot be shared with IP, IPX, or ASRT clients.

Valid Values: Any valid selector that has not been previously used and is within the range defined for the device.

Default Value: 0

Re-registration time with Arp Server (in minutes)
Specifies the time interval between registration requests from the client to the ARP Server.

Valid Values: An integer in the range of 0 to 65535

Default Value: 15

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.

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
```

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
```

Example for Bridging:

```
ARP config> add pvc-atm-arp-entry
Interface Number [0]?
Protocol [IP]? ASRT
Channels for this protocol must be added under ASRT Config by adding
a port.
```

interface number
Valid values: The number of the interface assigned

Default value: 0

protocol
Valid values: *IP*, *IPX*, *ASRT*

Default value: IP

local client IP address
Required for IP. This address associates this PVC with a client.

Valid Values: Any valid IP address

Default Value: 0.0.0.0

destination protocol address
Valid Values: Any valid IP address. (For IPX, any valid 6-byte IPX host number.)

Default Value: 0.0.0.0

permanent virtual circuit VPI
Valid Values: any valid value in the range of 0 to 255

Default Value: 0

permanent virtual circuit VCI

Valid Values: any value in the range of 0 to 65535

Default Value: 0

svc-atm-arp-entry

Adds an SVC and optionally creates a permanent ARP Entry.

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 []? 39840f000000000000000000000210005a00dead03
```

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 []? 39840f000000000000000000000210005a00dead03
```

Example for Bridging:

```
ARP config> add svc-atm-arp-entry
Interface Number [0]?
Protocol [IP]? ASRT
Channels for this protocol must be added under ASRT Config by adding
a port.
```

interface number

Valid values: The number of the interface assigned.

Default value: 0

protocol

Valid values: *IP*, *IPX*, or *ASRT*

Default value: IP

local client IP address

Required for IP. This address associates this SVC with a client.

Valid Values: Any valid IP address

Default Value: 0.0.0.0

destination protocol address

Valid Values: Any valid IP address. For IPX, any valid 6-byte IPX host number.

Default Value: 0.0.0.0

destination ATM address

Valid Values: Any valid ATM address

Default Value: None

Redundancy

Use this command to add a redundancy configuration.

ARP Server Redundancy Example

```
ARP config>add red
Interface Number [0]?
Protocol [IP]?
Local client IP Address [0.0.0.0]? 1.1.1.1
Note: Local Redundancy ESI must be different from the Primary Server ESI
```

```
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) AAAAAAAAAAAA
  (  4) BBBBBBBBBBBB
Select Local Redundancy ESI [1]? 3
Local Redundancy selector, Range 00..FF [00]? aa
Partner's (Redundancy) ATM Address, 40 Hex digits
 [0000000000000000000000000000000000000000]?
39840f0000000000000000001bbbbbbbbbbbbbb  1
Redundancy default IP Gateway also ? [No]: yes
If Yes, Redundancy default IP Gateway protocol address [0.0.0.0]? 1.1.1.3
ARP config>

---------- Backup Server (client) -----------------

ARP config>add red
Interface Number [0]?
Protocol [IP]?
Local client IP Address [0.0.0.0]? 1.1.1.2
Note: Local Redundancy ESI must be different from the Primary Server ESI
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) AAAAAAAAAAAA
  (  4) BBBBBBBBBBBB
Select Local Redundancy ESI [1]? 4
Local Redundancy selector, Range 00..FF [00]? bb
Partner's (Redundancy) ATM Address, 40 Hex digits
 [0000000000000000000000000000000000000000]?
 39840f0000000000000000001aaaaaaaaaaaaaa  1
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) AAAAAAAAAAAA
  (  4) BBBBBBBBBBBB
Partner Server ESI [1]? 2
Partner Server selector, Range 00..FF [00]? 11
Redundancy default IP Gateway also ? [No]: yes
If Yes, Redundancy default IP Gateway protocol address [0.0.0.0]? 1.1.1.3
ARP config>exit
Config>
```

Note: **1** This input should occur on the previous line. It is shown here on a separate line because of formatting.

Interface Number
Interface Number assigned

Protocol
IP

Local Client IP Address
Associates this redundancy configuration with a client.

Valid Values: Any IP address

Default Values: 0.0.0.0

Select Redundancy ESI
Specifies whether the universally administered MAC address or a MAC address configured under the ATM interface configuration should be used as the End System Identifier component of the CIPC or CIPS ATM address. This question is preceded by a list of valid ESIs from which the selection is to be made.

Valid Values: The redundancy ESI must be different from all of the primary ARP client/server functional ESIs.

Default Value: 1

Choose Redundancy Selector
Identifies the selector byte for the Redundancy ATM address.

Valid Value: any single octet value that has not been previously used and is within the range defined for the device.

Default Value: 00

Partner's (Redundancy) ATM Address
Specifies the ATM address of the Redundancy ARP Server.

Valid Values: Only private NSAP addresses are valid. The first byte (Authority and Format Identifier) must contain a value of:
39 — Data Country Code ATM Format
47 —- International Code Designator ATM Format
45 — E.164 ATM Format

Default Value: none

Partner Server ESI
Specifies the ESI component of the partner real ATM address.

Valid Values: Valid server ESI listed in the menu preceding this question.

Default Value: 1

Partner Server Selector
Specifies the selector component of the partner real ATM address.

Valid Values: Value defined for server selector

Default Value: 00

Redundancy's default IP gateway also?
Specifies whether this ARP entity will participate in the provision of default gateway redundancy support for the LIS.

Default Value: No

Redundancy's default IP gateway address
Specifies the IP address of the Redundancy default gateway for this LIS. This is the IP address configured at hosts using the router as their default router.

Default Value: 0.0.0.0

Change

Use the **change** command to change the ATM-ARP configuration.

Syntax:

change entry
atm-arp-client-configuration
redundancy
multicast-support

atm-arp-client-configuration
Changes the atm-arp-client-configuration.

See page 593 for a description of the **Change** parameters.

Example for IP:

```
ARP config> change atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]?
Client IP Address [0.0.0.0]? 1.1.1.2
```

```
This client is also a server? [Yes] yes
Refresh timeout (in minutes) [20]?
Enable auto-refresh? [Yes]:
Refresh by InAtmArp? [Yes]:
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) 222222222222
  (  4) 121212121212
  (  5) AAAAAAAAAAAA
Select ESI [1]?2
Use internally assigned selector? [Yes]: no
Selector Only, Page 00..FF [00] ? 11
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Server 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 Server 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]?
Participate in Server Synchronization [No]? yes
Server Group ID [1]?
Do you want to accept sessions from non-configured DCSs [yes]?
Hello Interval [3]
Dead Factor [3]?
(  1) Use burned in ESI
(  2) 111111111111
(  3) 222222222222
(  4) 121212121212
(  5) AAAAAAAAAAAA
Server Synchronization ESI [2]?
Server Synchronization selector, Range 00..FF [00]? 12
Server Synchronization Max SDU size (bytes) [9188]?
Re-registration time with Arp Server (in minutes) [15]?


To enable or change multicast support,

please issue the ADD or CHANGE MULTICAST-SUPPORT command.
```

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 Server 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.

Example for Bridging:

```
ARP config> change atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]? asrt
Client Address (Port Number) [0]? 2    1
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [No]:
Selector Only, Range 00..FF [0A]?
Validate PCR for best effort VCCs? [No]:
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]?
```

Note: **1** In the case of Bridging, you are prompted for a port number instead of a protocol address.

Also, if using SVC support for a particular port, an internally assigned selector should not be used for the corresponding client. The selector must be specified by the user so that the ATM address for this client is definitively known for configuration at the other end.

Use of this command is required only if you wish to use values other than the defaults for the traffic parameters.

redundancy
Changes the redundancy configuration for a client.

See page 599 for a description of the parameters you can change.

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 arp-server
atm-arp-client-configuration
pvc-atm-arp-entry
svc-atm-arp-entry
redundancy
mars-server

arp-server
Deletes an arp-server or DCS.

Specify the address of the arp-server. This question is preceded by a list of valid arp-servers or DCSs from which the selection is to be made.

Valid Values: any of the values listed in the menu preceding this question.

Default Value: 0

Example for IP:

```
ARP config> del arp-server

ATM Arp Remote Server List:
   IP Address      Number       Address / Sub Address
        1.1.1.1     [ 1]       39.84.0F.00.00.00.00.00.00.00.00.00.02.
                                          11.11.11.11.11.11.11
        1.1.1.1     [ 2]       39.84.0F.00.00.00.00.00.00.00.00.00.03.
                                          AA.AA.AA.AA.AA.AA.AA

Number of the IP Address/Arp Server pair to be deleted
Default of 0 will delete nothing [0]? 1
```

atm-arp-client-configuration
Deletes an atm-arp-client-configuration for a client.

Specify the interface number, protocol, and Client IP address.

interface number
Valid values: any defined interface

Default value: 0

protocol
Valid values: *IP*, *IPX*, **or** *ASRT*

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.

Example for Bridging:

```
ARP config> del atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]? ASRT
Clients for this protocol can only be changed here.
Deletions must be done under ASRT Config by deleting a port.
```

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.

Example for Bridging:

```
ARP config> del pvc
ATM Arp Permanent Virtual Circuit Definitions
No. IF# Prot# P/S     Protocol     -> VPI / VCI    (Client Address)
1    0   23    P                   ->   0 / 87     (Port: 1)
Which Arp entry do you want to delete [0]? 1
Channels for this protocol must be deleted under ASRT Config by
deleting a port.
```

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.

Example for Bridging:

```
   ARP config>del svc
   ATM Arp Switched Virtual Circuit Definitions
   No. IF  Prot  P/S     Protocol     -> Destination ATM Address (Client)
   2    0   23    S                     -> 39.11.22.33.44.55.66.77.88.99.00.11.22
                                       33.44.55.66.77.88.99 (Port: 2)
Which Arp entry do you want to delete [0]? 2
Channels for this protocol must be deleted under ASRT Config by deleting a port.
```

redundancy
Deletes the redundancy configuration for a client.

Disable

Use the **disable** command to disable auto-refresh of ARP entries.

The *auto-refresh* configuration value is overridden by the setting of *auto-refresh* at the atm-arp-client configuration. See page 593 for information about atm-arp-client configuration parameters.

Syntax:

disable auto-refresh

auto-refresh
Disables auto-refresh of an ARP entry.

Valid Values: Yes or No

Default Value: Yes for client, No for Server

Enable

Use the **enable** command to enable auto-refresh of ARP entries.

The *auto-refresh* configuration value is overridden by the setting of *auto-refresh* at the atm-arp-client configuration. See page 593 for information about atm-arp-client configuration parameters.

Syntax:

enable auto-refresh

auto-refresh
Enables auto-refresh of an ARP entry.

Valid Values: Yes or No

Default Value: No for client, Yes for Server

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.

ARP Over ATM Configuration Commands (Talk 6)

Syntax:

list entry
all
arp-servers
atm-arp-client-configuration
pvc-atm-arp-entry
svc-atm-arp-entry
redundancy
mars-servers

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      Number       Address / Sub Address
 1.1.1.1     [ 1]       39.84.0F.00.00.00.00.00.00.00.00.00.02.
                                     11.11.11.11.11.11.11
 1.1.1.1     [ 2]       39.84.0F.00.00.00.00.00.00.00.00.00.03.
                                     AA.AA.AA.AA.AA.AA.AA
```

arp-servers

Lists arp-servers. If this is a client-only configuration, this output lists the configured ARP Servers. If this is a Distributed ARP Server configuration, then configured Directly Connected Servers are listed. See “Distributed ARP Server Overview” on page 580 for an explanation of communication between Distributed ARP Servers.

```
ARP config> list arp-servers

ATM Arp Remote Server List:
   IP Address      Number       Address / Sub Address
        1.1.1.1     [ 1]       39.84.0F.00.00.00.00.00.00.00.00.00.02.
                                          11.11.11.11.11.11.11
        1.1.1.1     [ 2]       39.84.0F.00.00.00.00.00.00.00.00.00.03.
                                          AA.AA.AA.AA.AA.AA.AA
```

atm-arp-client-configuration

Lists the atm-arp-client-configuration.

```
ARP config> list atm

ATM Arp Clients:

--------------------------------------------------

If: 0  Prot: 0  Addr: 1.1.1.2          ESI: 11.11.11.11.11.11  Sel: 11
Server: yes  Refresh T/O: 20   AutoRefr: yes   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
Server Synchronization: yes  SGID: 1       Secure DCSs: no
Hello Interval: 3    Dead Factor: 3
Server Synchronization ESI: 11.11.11.11.11.11   Selector: 12
Server Synchronization Max SDU(bytes):   9188
Arp Server Re-registration time in (minutes): 15
Multicast Support: no    Broadcast Support: no
```

See 593 for field descriptions.

redundancy

Lists the redundancy configurations and the gateway status.

```
ARP config>list red


ATMARP Clients with Redundancy Configured
-----------------------------------------

If:  0    Prot: IP    Addr: 1.1.1.2
Red. ESI: bb.bb.bb.bb.bb.bb   Red. SEL: bb   Pri/Secy: Secondary
Partner's (Red.) ATM Address:  39.84.0F.00.00.00.00.00.00.00.00.01.aa.aa.aa.aa.aa.aa.aa
Partner Server ESI: BB.BB.BB.BB.BB.BB.BB Partner Server SEL: AA
Redundancy Default IP Gateway Address: 1.1.1.3
-----------------------------------------

ARP config>list red


ATMARP Clients with Redundancy Configured
-----------------------------------------

If:  0    Prot: IP    Addr: 2.2.2.2
Red. ESI: aa.aa.aa.aa.aa.aa  Red. SEL: aa   Pri/Secy: Primary
Partner's (Red.) ATM Address:  39.84.0F.00.00.00.00.00.00.00.00.01.bb.bb.bb.bb.bb.bb.bb
Partner Server ESI: BB.BB.BB.BB.BB.BB.BB Partner Server SEL: AA
Redundancy Default IP Gateway Address: 1.1.1.3
-----------------------------------------
```

If: Interface Number

Prot: IP

Addr: IP Address

Red. ESI:

Identifies the ESI portion of the redundancy ATM address:

Burned In - Specifies that the universally administered MAC address of the ATM adapter should be used as the End System Identifier (ESI) portion of the redundancy ATM address.

Locally administrated - Identifies a Locally Administered End System Identifier that is to be used as the ESI component of the redundancy ATM address.

Red. SEL:

Identifies the Selector portion of the redundancy ATM address.

Peer Redundancy/Primary/Secondary

Identifies the role of the client/server.

If primary, the client/server will place a call from its redundancy ATM address to the secondary's redundancy ATM address.

If secondary, this client/server will be idle as long as the redundancy VCC is established.

If peer, the server with the higher redundancy ATM address places the call.

Partner's (Red.) ATM Address:

Specifies the redundancy ATM address for the partner client/server.

Partner Server ESI:

Identifies a locally administered ESI that is to be used as the ESI component of the partner ARP Server ATM address when the ARP Server fails.

Partner Server SEL:

Identifies the Selector portion of the partner server ESI and selector configured on the partner router.

Redundancy Default IP Gateway Address:

Specifies the default IP gateway address. This is the default gateway address configured in the clients served by this ARP

Server. Defining this address enables the ARP Server to provide routing function from one subnet to another subnet.

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
2    0   23    P                   ->  00 / 0068
```

No. VCC number

IF# Interface Number

Prot# Protocol number (Prot# 0 = IP, 7 = IPX, 23=ASRT)

P/S: P for PVC, S for SVC

Protocol

IP Address (IPX host number if the protocol is IPX)

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
```

No VCC number

IF# Interface Number

Prot# Protocol number (Prot# 0 = IP, 7 = IPX, 23=ASRT)

P/S: P for PVC, S for SVC

Protocol

IP Address (IPX host number if protocol is IPX)

Destination ATM Address

Destination ATM Address

Reorder

Use the **reorder** command to select the primary ARP Server from a list of ARP Servers. This command is useful for clients only.

Syntax:

reorder arp

Example: `reorder arp`

```
ATM Arp Remote Server List:
   IP Address      Number       Address / Sub Address
        1.1.1.1     [ 1]       39.84.0F.00.00.00.00.00.00.00.00.02.
                                          11.11.11.11.11.11.11
        1.1.1.1     [ 2]       39.84.0F.00.00.00.00.00.00.00.00.03.
```

```
                                                AA.AA.AA.AA.AA.AA.AA
Number of the IP Address/Arp Server ATM Address pair to be made Primary
Default of 1 will change nothing [1]?
```

Set

Use the **Set** command to set the refresh timer value (in minutes). ARP entires will age out after this number of minutes if they are not refreshed.

Syntax:

set refresh-timer

refresh-timer

Specifies the value for the aging timer for ARP entries.

Valid Values: An integer number of minutes in the range of 0 to 65535

Default Value: 5

Sample ARP Configurations

Sample configurations show the complete sequence of events for that configuration.

ARP Server Redundancy Configuration in a Non-Distributed ARP Server LIS

Configuring a partner

The following sample shows configuring the a ARP Server for ARP Server Redundancy.

Notes:

1. Add the physical ATM device
2. Add locally administered ESIs
3. Add IP address to be used for server
4. Begin ARP Configuration
5. Define an ATM ARP client
6. Define the ARP Server
7. Use locally administered ESI
8. Begin redundancy configuration for IP address 1.1.1.1

```
Config (only)>add device atm       1
Device Slot #(0-3) [0]?
Adding CHARM ATM Adapter device in slot 0  port 1 as interface #0
Use "net 0" to configure CHARM ATM Adapter parameters
Config (only)>net
Network number [0]?
ATM user configuration
ATM Config>int
ATM interface configuration
ATM Interface Config>add esi   2
ESI in 00.00.00.00.00.00 form []? 11.11.11.11.11.11
ATM Interface Config>add esi
ESI in 00.00.00.00.00.00 form []? aa.aa.aa.aa.aa.aa
ATM Interface Config>exit
ATM Config>exit
Config (only)>p ip
Internet protocol user configuration
IP config>add addr    3
Which net is this address for [0]?
New address [0.0.0.0]? 1.1.1.1
Address mask [255.0.0.0]?
IP config>exit
```

```
Config (only)>p arp   4 .
ARP user configuration
ARP config>add atm   5
Interface Number [0]?
Protocol [IP]?
Client IP Address [0.0.0.0]? 1.1.1.1
This client is also a server? [No]: yes   6
Refresh timeout (in minutes) [20]?
Enable auto-refresh? [Yes]:
Refresh by InAtmArp? [Yes]:
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) AAAAAAAAAAAA
Select ESI [1]? 2      7
Use internally assigned selector? [Yes]: n
Selector Only, Range 00..FF [00]? 11
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Server 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 Server 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]?
Participate in Server Synchronization [N]?
Re-registration time with Arp Server (in minutes) [15]?
ARP config>add red      8
Interface Number [0]?
Protocol [IP]?
Local client IP Address [0.0.0.0]? 1.1.1.1
Note: Local Redundancy ESI must be different from the Primary Server ESI
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) AAAAAAAAAAAA
Select Local Redundancy ESI [1]? 3
Local Redundancy selector, Range 00..FF [00]? aa
Partner's (Redundancy) ATM Address, 40 Hex digits
[0000000000000000000000000000000000000000]?
39.84.0F.00.00.00.00.00.00.00.00.00.01.bb.bb.bb.bb.bb.bb.bb

Redundancy default IP Gateway also ? [Yes]:
If Yes, Redundancy default IP Gateway protocol address [0.0.0.0]? 1.1.1.3
ARP config>exit
Config (only)>reload
Are you sure you want to reload the gateway? (Yes or [No]): yes
  EXIT, reload requested
```

Configuring the Partner

The following sample shows configuring the partner for ARP Server Redundancy.

Notes:

1. Add the physical ATM device
2. Add locally administered ESIs
3. Begin ARP Configuration
4. This is the client definition
5. When secondary, define as client only (if not is a Distributed ARP Server LIS). If Distributed ARP Server, the partner can be defined as a Distributed ARP Server LIS.
6. Use locally administered ESI
7. Define the ARP Server
8. Begin redundancy configuration for IP address 1.1.1.2
9. Defining the secondary

```
Config (only)>add device atm   1
Device Slot #(0-3) [0]?
Adding CHARM ATM Adapter device in slot 0  port 1 as interface #0
Use "net 0" to configure CHARM ATM Adapter parameters
Config (only)>net
Network number [0]?
ATM user configuration
ATM Config>int
```

```
ATM interface configuration
ATM Interface Config>add esi
ESI in 00.00.00.00.00.00 form []? 11.11.11.11.11.11
ATM Interface Config>add esi   2
ESI in 00.00.00.00.00.00 form []? bb.bb.bb.bb.bb.bb
ATM Interface Config>exit
ATM Config>exit
Config (only)>p ip
Internet protocol user configuration
IP config>add addr
Which net is this address for [0]?
New address [0.0.0.0]? 1.1.1.2
Address mask [255.0.0.0]?
IP config>exit

Config (only)>p arp   3
ARP user configuration
ARP config>add atm    4
Interface Number [0]?
Protocol [IP]?
Client IP Address [0.0.0.0]? 1.1.1.2
This client is also a server? [No]: no    5
Refresh timeout (in minutes) [5]?
Enable auto-refresh? [no]:
Refresh by InAtmArp? [no]:
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) BBBBBBBBBBBB
Select ESI [1]? 1   6
Use internally assigned selector? [Yes]: y
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Server 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 Server for Data VCCs? [Yes]:
Peak Cell Rate of outbound Data VCCs (Kbps) [0]?
Sustained Cell Rate of outbound Data VCCs (Kbps) [0]?
Participate in Server Synchronization  (Y or N) [N]?    N
Max SDU size (bytes) [9188]?
Re-registration time with Arp Server (in minutes) [15]?
ARP config> add arp-server private-nsapa 7
Local Client IP Address [0.0.0.0]? 1.1.1.2
Private NSAP Address: Specify 40 digits
ATM Address []? 39840F0000000000000000000111111111111111
ARP config>add red   8
Interface Number [0]?
Protocol [IP]?
Local client IP Address [0.0.0.0]? 1.1.1.2
Note: Local Redundancy ESI must be different from the Primary Server ESI
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) BBBBBBBBBBBB
Select Local Redundancy ESI [1]? 3
Local Redundancy selector, Range 00..FF [00]? bb
Partner's (Redundancy) ATM Address, 40 Hex digits
[0000000000000000000000000000000000000000]?
39.84.0F.00.00.00.00.00.00.00.00.00.01.aa.aa.aa.aa.aa.aa.aa    9
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) BBBBBBBBBBBB
Select Partner Server ESI [1]? 2
Partner selector, Range 00..FF [00]? 11
Redundancy default IP Gateway also ? Yes
If Yes, Redundancy default IP Gateway protocol address [0.0.0.0]? 1.1.1.3
ARP config>exit
Config (only)>reload
Are you sure you want to reload the gateway? (Yes or [No]): yes
  EXIT, reload requested
```

Notes:

1. In a Distributed ARP Server LIS, both partners are Distributed ARP Servers.
2. Configuring ARP Server Redundancy is the same process for the Distributed ARP Server case as for the non-Distributed ARP Server case.

Accessing the ARP Monitoring Environment

Use the following procedure to access the ARP monitoring 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" in *Software User's Guide*.) For example:

   ```
   * talk 5
   +
   ```

   After you enter the **talk 5** command, the GWCON prompt (+) displays on the terminal. 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 Monitoring Commands for Non-ATM Networks

This section describes the ARP monitoring commands for non-ATM nets. You can access ARP monitoring commands at the `ARP>` prompt.

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 monitoring prompts.

Table 39 shows the commands.

Table 39. ARP monitoring Commandsfor non-ATM Networks Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
Clear	Clear the cache for a specified interface.
Dump	Display the cache for a specified interface.
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 ARP information.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

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 monitoring 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**

```
 ARP>du 2 ip
Hardware Address    IP Address  IP-Encap    Refresh
 02-07-01-00-00-01  1.1.1.1                 19
 a1-b2-c3-4d-5e-6f  1.1.1.2     ieee 802.3  Permanent
 100                1.1.1.3     ieee 802.3  Permanent
 16                 1.1.1.5                 Not Aging
```

Notes:

1. 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** monitoring command.

2. IP encapsulation applies only to static ARP entries (those with permanent refresh intervals). See the Talk 6 **add entry** command for more information.

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 313.

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  0

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.

ARP Over ATM Monitoring Commands

This section describes the ARP over ATM (CIP) monitoring commands. It describes the monitoring commands for:

- Classical IP & ARP over ATM
- IPX over ATM
- 1483 bridging

The monitoring commands for IPX 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 monitoring. You can access ARP monitoring commands at the `ARP>` prompt. Table 40 shows the commands.

For further information, refer to “Classical IP and ARP Over ATM (RFC 1577)” on page 573 and “IPX and ARP Over ATM Overview (RFC 1483)” on page 584. For additional information on ARP over ATM, and for illustrations showing logical and physical network configurations, refer to *Software User’s Guide*

Since bridging does not use ARP, the monitoring may only be used to check the status of a channel associated with a bridge port. Also, because bridging does not use protocol addresses, only the port number for the associated (local) port is displayed with a channel.

Table 40. ARP Over ATM Monitoring Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.

Table 40. ARP Over ATM Monitoring Command Summary (continued)

Command	Function
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.
Redundancy-State	Display IP clients configured with Redundancy
Statistics	Display statistics of the ARP code over all of the network interfaces.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

Activate

Use the **activate** command to activate configured Classical IP and IPX over RFC 1483 PVCs or SVCs without having to reboot the device.

Syntax:

activate channel

pvc

svc

channel

Activates PVCs and SVCs.

all Use *all* to indicate that the command applies to all interfaces.

specific
Activates channels on the network interface number you provide.

pvc Activates PVCs only.

all Use *all* to indicate that the command applies to all interfaces.

specific
Activates PVCs on the network interface number you provide.

svc Activates SVCs only.

all Use *all* to indicate that the command applies to all interfaces.

specific
Activates SVCs on the network interface number you provide.

Example:

```
ARP>  activate chan spec 0
ATM Arp PVC/SVC(s) Activated:

No. IF# Prot# P/S Protocol -> VPI/VCI or Dest ATM Addr (Client Addr)
 2   0    0    P  4.2.11.2 -> 00/924
 3   0    0    S           ->39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.AB.CD.C8
```

No Specifies the VCC number

IF# Specifies the interface number.

PROT#
Specifies the protocol number:

- IP = 0
- IPX = 7

P/S P indicates PVC. S indicates SVC.

Protocol
Specifies the destination IP address or the IPX host number.

VPI/VCI or Dest ATM Addr
Specifies the VPI/VCI of the defined PVC or the destination ATM address of the SVC.

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.00.00.00.00.00.00.00.00.02.10.00.5A.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 bps.

FwdScr
The Sustained Cell Rate in bps.

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.

Client Address
This is the protocol address of the local client attached to this VCC.

Target Protocol Address
This is the protocol address of the remote client attached to this VCC.

Dump

Use the **dump** command to show which ATM channels are being used for sending datagrams, and their corresponding destination 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>du 0
Hardware Address  IP Address          Refresh      Origin Srvrid   Seq number
  0/0               1.1.1.1            19               1.1.1.1      868997267
```

```
0/0              1.1.1.2           permanent        1.1.1.2      868997028
0/0              1.1.1.3           permanent        1.1.1.2      868997028
0/35             1.1.1.5           20               1.1.1.2      868997042
```

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.

The following are valid Refresh states:

Hardware Address
Specifies the hardware addresses registered with the ARP. If “0/0” appears under the Hardware Address, there is no channel open for this ARP entry.

Refresh States

-
- If “resolve only” appears under Refresh, this ARP entry exists only for the purposes of resolving the ATM address of the given IP address. These entries are used for registering with the ARP server (2225 registration).
- If “not aging” appears under Refresh, the entry will remain indefinitely.

Origin Server Id
The DCS ID (IP address for IBM products) of the server that originated this cache entry. It is possible that if a client registers with multiple servers, there will be more than one cache entry for that client’s IP address. There will be one entry for each server with which the client registers, but each will have a different Origin ID.

Seq. Number
The current sequence number of the corresponding SCSP cache entry (in decimal). This number should match on all servers in the server group for this particular cache entry.

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
```

Redundancy-State

Use the **redundancy-state** command to display (by network) the IP clients that are configured with redundancy.

Syntax:

redundancy-state

The following example shows the Redundancy monitoring section on the primary ARP Server when the Redundancy channel is inactive.

```
CGW Operator monitoring
+p arp
ARP>red
Network number  [0]?
Protocol [IP]
If: 0 Prot: IP  Clients configured with Redundancy
-------------------------------------------------------
Addr: 1.1.1.1   Place Redundancy Call: Yes  Real Esi: Up  Red. Esi: Up  Red. Chnl: Down
                    FLAGS: Real Client: C8  Red. Client: C8 RedFlags: C0
```

```
                    Red. Channel: 0/0
                     Red. Channel: Source ATM address
 39.84.0F.00.00.00.00.00.00.00.00.00.01.AA.AA.AA.AA.AA.AA.AA
                     Red. Channel: Target ATM address
39.84.0F.00.00.00.00.00.00.00.00.00.01.BB.BB.BB.BB.BB.BB.BB
                     Redundancy Status: Active
 Redundancy default IP Gateway protocol address: 1.1.1.3
```

The following example shows the Redundancy monitoring section on the primary Server when the Redundancy channel is active.

```
CGW Operator monitoring
+p arp
 ARP>red
 Network number  [0]?
 Protocol [IP]?
 If: 0 Prot: IP  Clients configured with Redundancy
 ---------------------------------------------------------
 Addr: 1.1.1.1   Place Redundancy Call: Yes  Real Esi: Up  Red. Esi: Up  Red. Chnl: Up
                     FLAGS: Real Client: C8  Red. Client: C8 RedFlags: D0
                     Red. Channel: (VPI/VCI)  0/32
                     Red. Channel: Source ATM address
 39.84.0F.00.00.00.00.00.00.00.00.00.01.AA.AA.AA.AA.AA.AA.AA
                     Red. Channel: Target ATM address
 39.84.0F.00.00.00.00.00.00.00.00.00.01.BB.BB.BB.BB.BB.BB.BB
                     Redundancy Status: Active
 Redundancy default IP Gateway protocol address: 1.1.1.3
 ---------------------------------------------------------
```

The following example shows the Redundancy monitoring section on the partner when the Redundancy channel is inactive and the partner is acting as the backup Arp Server.

```
ARP>red
 Network number  [0]?
 Protocol [IP]?
 If: 0 Prot: IP  Clients configured with Redundancy
 ---------------------------------------------------------
 Addr: 1.1.1.2   Place Redundancy Call: No  Real Esi: Up  Red. Esi: Up  Red. Chnl: Down
                     FLAGS: Real Client: C8  Red. Client: C8 RedFlags: 80
                     Red. Channel: 0/0
                    Red. Channel: Source ATM address
 39.84.0F.00.00.00.00.00.00.00.00.00.01.BB.BB.BB.BB.BB.BB.BB
                     Red. Channel: Target ATM address
 39.84.0F.00.00.00.00.00.00.00.00.00.01.AA.AA.AA.AA.AA.AA.AA
                     Redundancy Status: Active
 Partner Server ESI: 11.11.11.11.11.11, Partner Server SEL: 11, In backup Server mode
 Redundancy default IP Gateway Protocol address: 1.1.1.3
 --------------------------------------------
```

The following example shows the Redundancy monitoring section on the secondary when the Redundancy channel is active and the secondary is acting as client and not in backup ARP Server mode (Since Primary ARP server is active).

```
ARP>red
Network number  [0]?
 Protocol [IP]?
 If: 0 Prot: IP  Clients configured with Redundancy
 ---------------------------------------------------------
Addr: 1.1.1.2   Place Redundancy Call: No  Real Esi: Down  Red. Esi: Up  Red. Chnl: Up
                     FLAGS: Real Client: C0  Red. Client: C8 RedFlags: 90
                     Red. Channel: (VPI/VCI)  0/32
                     Red. Channel: Source ATM address
 39.84.0F.00.00.00.00.00.00.00.00.00.01.BB.BB.BB.BB.BB.BB.BB
                    Red. Channel: Target ATM address
 39.84.0F.00.00.00.00.00.00.00.00.00.01.AA.AA.AA.AA.AA.AA.AA
                     Redundancy Status: Inactive, not trying ...
 Partner Server ESI: 11.11.11.11.11.11, Partner Server SEL: 11, Acting as a Client
 Redundancy default IP Gateway Protocol address: 1.1.1.3
 --------------------------------------------
 ARP>exit
```

The following fields are displayed from the Redundancy monitoring:

If: Specifies the interface number.

Prot: Specifies the protocol.

Addr: Specifies the IP Address of the client.

Place Redundancy Call
Indicates if the client will initiate the redundancy call. See page 583.

Real ESI:
Indicates the state of the real ESI. If the state is Up, the client's ATM address registered successfully with the switch. If the state is Down, the client's attempt to register its ATM address with the switch was unsuccessful.

Red. ESI:
Indicates the state of the Redundancy ESI. If the state is Up, the Redundancy channel ATM address registered successfully with the switch. If the state is Down, the Redundancy channel ATM address did not register successfully with the switch.

Red. Chnl:
Indicates the state of the of the Redundancy channel. If the state is Up, the Redundancy channel is set up between the primary and secondary. If the state is Down, the Redundancy channel is not set up between the primary and secondary.

Red Channel
Provides the VPI/VCI of the Redundancy channel.

Red. Channel Source ATM address:
Identifies the ATM address of the source Redundancy channel.

Red. Channel Target ATM address:
Identifies the ATM address of the target Redundancy channel.

Redundancy Status:
Identifies the status of redundancy.

Active Providing ARP Services.

Inactive
Not providing ARP Services.

Partner Server ESI:
Identifies a locally administered ESI that is to be used as the ESI component of the partner ARP Server ATM address when the partner ARP Server fails.

Partner Server SEL:
Identifies the Selector portion of the ESI and selector configured on the partner device. This field is displayed for backup or peer ARP Server only.

Redundancy Default IP Gateway Address:
Specifies the default IP gateway address. This is the default gateway address configured in the clients served by this ARP Server. This address enables the ARP Server to provide routing function from one subnet to another subnet.

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 614.

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
```

ARP over ATM Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

ARP over ATM supports the CONFIG (Talk 6) **delete interface** command with the following consideration:

When an ATM or Virtual ATM interface is deleted, all ATM ARP client, PVC, SVC and redundancy information related to that interface is deleted.

GWCON (Talk 5) Activate Interface

ARP over ATM supports the GWCON (Talk 5) **activate interface** command with the following consideration:

If MARS is enabled on the ATM ARP client, then **activate** is not supported.

The following table summarizes the ARP over ATM configuration changes that are activated when the GWCON (Talk 5) **activate interface** command is invoked:

Commands whose changes are activated by the GWCON (Talk 5) activate interface command
CONFIG, protocol, arp, add arp-server private-nsapa
CONFIG, protocol, arp, add atm-arp-client-configuration
CONFIG, protocol, arp, add pvc-atm-arp-entry
CONFIG, protocol, arp, add redundancy
CONFIG, protocol, arp, add svc-atm-arp-entry
CONFIG, protocol, arp, change atm-arp-client-configuration
CONFIG, protocol, arp, change redundancy
CONFIG, protocol, arp, reorder arp-server
Commands performed as part of the GWCON (Talk 5) activate interface command
GWCON, protocol, arp, activate channel specific
GWCON, protocol, arp, activate svc specific
GWCON, protocol, arp, activate pvc specific

GWCON (Talk 5) Reset Interface

ARP over ATM supports the GWCON (Talk 5) **reset interface** command with the following consideration:

If MARS is enabled on the ATM ARP client, then reset is not supported.

The following table summarizes the ARP over ATM configuration changes that are activated when the GWCON (Talk 5) **reset interface** command is invoked:

Commands whose changes are activated by the GWCON (Talk 5) reset interface command
CONFIG, protocol, arp, add arp-server private-nsapa
CONFIG, protocol, arp, add atm-arp-client-configuration
CONFIG, protocol, arp, add pvc-atm-arp-entry
CONFIG, protocol, arp, add redundancy
CONFIG, protocol, arp, add svc-atm-arp-entry
CONFIG, protocol, arp, change atm-arp-client-configuration
CONFIG, protocol, arp, change redundancy
CONFIG, protocol, arp, delete arp-server
CONFIG, protocol, arp, delete atm-arp-client-configuration
CONFIG, protocol, arp, delete pvc-atm-arp-entry
CONFIG, protocol, arp, delete redundancy
CONFIG, protocol, arp, delete svc-atm-arp-entry
CONFIG, protocol, arp, reorder arp-server
Commands performed as part of the GWCON (Talk 5) reset interface command
GWCON, protocol, arp, activate channel specific
GWCON, protocol, arp, activate svc specific
GWCON, protocol, arp, activate pvc specific

GWCON (Talk 5) Component Reset Commands

ARP over ATM supports the following ARP over ATM-specific GWCON (Talk 5) **reset** commands:

GWCON, Protocol, ARP, Activate Channel Specific Command

Description:

Activates the new ATM PVCs or SVCs configured for the specified interface.

Network Effect:

none

Limitations:

None

The following table summarizes the ARP over ATM configuration changes that are activated when the **GWCON, protocol, arp, activate channel specific** command is invoked:

Commands whose changes are activated by the GWCON, protocol, arp, activate channel specific command

CONFIG, protocol, arp, add pvc-atm-arp-entry
CONFIG, protocol, arp, add svc-atm-arp-entry

GWCON, Protocol, ARP, Activate Channel All Command

Description:
Activates the new ATM PVCs or SVCs configured for all interfaces.

Network Effect:
None.

Limitations:
None.

The following table summarizes the ARP over ATM configuration changes that are activated when the **GWCON, protocol, arp, activate channel all** command is invoked:

Commands whose changes are activated by the GWCON, protocol, arp, activate channel all command
CONFIG, protocol, arp, add pvc-atm-arp-entry
CONFIG, protocol, arp, add svc-atm-arp-entry

GWCON, Protocol, ARP, Activate SVC Specific Command

Description:
Activates the new ATM SVCs configured for the specified interface.

Network Effect:
None.

Limitations:
None.

The following table summarizes the ARP over ATM configuration changes that are activated when the **GWCON, protocol, arp, activate svc specific** command is invoked:

Commands whose changes are activated by the GWCON, protocol, arp, activate svc specific command
CONFIG, protocol, arp, add svc-atm-arp-entry

GWCON, Protocol, ARP, Activate SVC All Command

Description:
Activates the new ATM SVCs configured for all interfaces.

Network Effect:
None.

Limitations:
None.

The following table summarizes the ARP over ATM configuration changes that are activated when the **GWCON, protocol, arp, activate svc all** command is invoked:

Commands whose changes are activated by the GWCON, protocol, arp, activate svc all command
CONFIG, protocol, arp, add svc-atm-arp-entry

GWCON, Protocol, ARP, Activate PVC Specific Command

Description:
Activates the new ATM PVCs configured for the specified interface.

Network Effect:
None.

Limitations:
None.

The following table summarizes the ARP over ATM configuration changes that are activated when the **GWCON, protocol, arp, activate pvc specific** command is invoked:

Commands whose changes are activated by the GWCON, protocol, arp, activate pvc specific command
CONFIG, protocol, arp, add pvc-atm-arp-entry

GWCON, Protocol, ARP, Activate PVC All Command

Description:
Activates the new ATM PVCs configured for all interfaces.

Network Effect:
None.

Limitations:
None.

The following table summarizes the ARP over ATM configuration changes that are activated when the **GWCON, protocol, arp, activate pvc all** command is invoked:

Commands whose changes are activated by the GWCON, protocol, arp, activate pvc all command
CONFIG, protocol, arp, add pvc-atm-arp-entry

GWCON, Protocol, IP, Reset IP Command

Description:
When a new IP address is activated, the corresponding ATM ARP parameters are activated for that new IP address.

Network Effect:
None.

Limitations:
None.

The following table summarizes the ARP over ATM configuration changes that are activated when the **GWCON, protocol, ip, reset ip** command is invoked:

Commands whose changes are activated by the GWCON, protocol, ip, reset ip command
CONFIG, protocol, arp, add arp-server private-nsapa
CONFIG, protocol, arp, add atm-arp-client-configuration
CONFIG, protocol, arp, add pvc-atm-arp-entry
CONFIG, protocol, arp, add redundancy
CONFIG, protocol, arp, add svc-atm-arp-entry

Chapter 29. Monitoring the Server Cache Synchronization Protocol (SCSP)

Use the SCSP monitoring function to monitoring the state of server synchronization.

Accessing the SCSP monitoring Environment

Use the following procedure to access the SCSP monitoring commands. This process gives you access to the SCSP *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 (+) is displayed on the terminal. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the + prompt, enter the **protocol scsp** command to get you to the `SCSP` prompt.

   Example:

   ```
   + prot scsp
   SCSP>
   ```

SCSP Monitoring Commands

This section describes the SCSP monitoring commands. You can access SCSP monitoring commands at the `SCSP>` prompt. Table 41 shows the commands.

Table 41. SCSP Monitoring Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
List	List SCSP monitoring information.
Statistics	Displays SCSP statistics for a Directly Connected Server.
Dump	Dumps SCSP cache for a server group.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

List

Use the **List** command to display information about the state of server synchronization. You can also enter a ? after a specific command name to list its options.

Syntax:

list dcs . . .
server-groups . . .

dcs *interface sgid*

Lists the all DCSs within a server group.

interface

Specifies the interface number that the DCS is defined upon.

sgid Specifies the server-group identifier of this DCS.

Example:

```
SCSP>li dcs
Network number [0]?
Server Group ID [0]?
      DCS Id      HFSM State  CAFSM State  M/S  CRL Len  ReTran Len
 03.04.02.00   Bidirectional      Aligned   S         0           0
```

DCS Id
The hex identifier of the DCS (on the other end of the VCC).

HFSM State
The state of the Hello Finite State Machine for this DCS session. The states are:

Down_Inop
The channel to the DCS is down.

Down The channel to the DCS is down, but an open is in progress.

Waiting
The Local Server (LS) has sent an Hello message to the DCS and it is waiting for a response.

Unidirectional
The LS has received a Hello message from the DCS but it has not yet recognized this Local Server.

Bidirectional
The LS and the Directly Connected Server recognize each other. This is the normal state.

CAFSM State
The state of the Cache Alignment Finite State Machine for this DCS session. The states are:

Down Cache alignment has not yet begun.

MS_Neg
The LS and the DCS are negotiating for Master/Slave status.

Summarize
The LS and DCS are exchanging cache summarization information.

Update
The LS and DCS are exchanging full cache information.

Aligned
The LS and DCS are synchronized. This is the normal state.

M/S Indicates Master (M) or Secondary (S) status of this LS. This is significant only if the CAFSM is in Summarize, Update, or Aligned state.

CRL Len
Cache Length Request List length. This is the number of cache update records that remain to be sent during Update state.

ReTran Len
ReTransmit list length. This is the number of cache update records that remain to be acknowledged by the DCS.

server-groups *interface*

Lists information about all server groups.

interface

network interface number

Example:

```
SCSP config> list ser
Network number [0]?
       SGID  Protocol      LSID  DCSs  ATM Addr
          1    ATMARP  04040100     0  39.84.0F.00.00.00.00.00.00.00.00.00.01.
                                      11.11.11.11.11.11.01
          0    ATMARP  03040100     1  39.84.0F.00.00.00.00.00.00.00.00.00.01.
                                      11.11.11.11.11.11.01
```

sgid Server Group Identifier. This is the configured identifier for the servers in this server group (LIS).

Protocol

The type of database being exchanged.

lsid Hex value of the Local Server Identifier. This identifies this Server within the server group. For ATMARP, this is the client's IP address.

DCSs The number of Directly Connected Servers associated with this Local Server.

ATM address

The ATM address of this Local Server. DCSs must use this to set up connections with this Local Server.

Statistics

Use this command to show statistics for a Directly Connected Server.

Use the **Statistics** command to display information about the state of server synchronization. You can also enter a ? after a specific command name to list its options.

Syntax:

statistics *interface server-group dcs-id*

interface

Specifies the interface number that the DCS is defined upon.

Valid values: Any defined interface

Default value: 0

server-group

Specifies the server-group of this DCS.

Valid values: 0 - 65535

Default value: 0

dcs-id Specifies the hex identifier of the DCS (on the other end of the VCC).

Valid Values:

Default Value: 0

Example:

```
SCSP>stat 0 0
DCS ID (hex) [0]?
DCS with that ID not found, listing all DCS's.
```

```
---------------------------------------------
DCS ID: 03.04.02.00
HFSM State: Bidirectional   DCS Hello Interval(sec): 3   DCS Dead Factor: 3
CAFSM State: Aligned  Master/Slave: S   CA seq: 33D35204   CSUS seq: 33D351F1
Cache Summary List sent?: yes   Cache Summary List ACKed?: yes
Cache Request List Size: 0   Cache ReTransmit List Size: 0   Age(sec): 302
ATM Addr: 39.84.0F.00.00.00.00.00.00.00.00.00.04.12.12.12.12.12.12.01
VPI: 0  VCI: 32  Missed Hello Msgs: 0   RID doesn't match LSID: 0
Short Messages: 0   Sequence Mismatches: 0
```

DCS Id
See page 628.

HFSM State
See page 628.

CAFSM State
See page 628.

Hello Interval
The interval in seconds that the DCS sends out Hello messages.

DCS Dead Factor
The number of Hello Intervals that must pass without receiving a Hello message after which this DCS is considered down.

Master/Slave
See page 628.

CA Seq
The current Cache Alignment message sequence number.

CSUS Seq
The current Cache State Update Solicit message sequence number.

Cache Summary List Sent?
Yes if the Cache Summary List has been completely transmitted to the DCS (during cache alignment state).

Cache Summary List Acked?
Yes if the entire Cache Summary List has been acknowledged (during cache alignment state).

Cache Request List size
See page 628.

Cache ReTransmit List size
See page 628.

Age The number of seconds that this DCS has been initialized.

VPI, VCI
The Virtual Path Identifier and Virtual Channel Identifier of the VCC to the DCS

Missed Hello Msgs
The number of Hello messages missed while in bidirectional state

RID doesn’t match LSID
Number of messages received where the Receiver ID in the message does not match the LSID of the LS.

Short Messages
Number of badly formed (too short) SCSP messages.

Sequence Mismatches
Number of sequence number protocol violations.

Dump

Use the **Dump** command to dump the SCSP cache for a server group.

Syntax:
dump *interface server-group*

interface
Specifies the interface number that the DCS is defined upon.

Valid values: Any defined interface

Default value: 0

server-group
Specifies the server-group of this DCS.

Valid values: 0 - 65535

Default value: 0

Example:

```
SCSP> dump 0 0
Next key to assign = 33D351F1
Key           Origin ID              Seq. No.   Age   Paddr
03.04.02.00   03.04.02.00           869487085     0   03.04.02.00
03.04.01.00   03.04.01.00           869487090     0   03.04.01.00
SCSP>
```

Key The hex cache key value for this cache entry. For ATM ARP this corresponds to the server's IP address (in hex).

Origin ID
The hex DCS ID of the server originating this cache entry. For ATM ARP, this corresponds to the server's IP address.

Age The number of minutes that this entry has existed without a matching server cache entry. Once 20 minutes is reached, this entry is aged.

Paddr The corresponding protocol address of this cache entry. If this is blank, there is no corresponding server cache entry (for example, ATM ARP cache entry).

Chapter 30. Using IPX

This chapter describes how to use the IPX protocol on your 2210. It includes the following sections:

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 stand-alone NetWare bridge.
- Support for the Novell NetBIOS emulator.

IPX Addressing

The following sections describe IPX addressing.

Network Numbers

An IPX network number specifies the location of a particular network in an internetwork. You can use multi-part 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). These addresses allow communication between two entities on different networks.

Host Numbers

Each IPX circuit needs a 6-byte host (node) number.

Token-Ring and Ethernet circuits 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. IPXWAN uses the configured node id followed by x’0000’.

ATM circuits use their End System Identifier (ESI) as their host number. Their burned-in ESI will be used if one has not been configured.

IPX Circuits

The IPX routing software models network interfaces as either a single IPX broadcast circuit, as one or more IPXWAN point-to-point circuits, or a combination of both types of circuits. The type of encapsulation, IPX addressing and routing protocols used on the circuit depend on the underlying DLC and whether the IPX circuit is configured as broadcast or IPXWAN point-to-point.

IPX broadcast circuits have the following characteristics:

- Used on LAN interfaces
- Used on WAN interfaces when IPXWAN is not configured
- Restricted to a single IPX broadcast circuit per interface
- Must be assigned a non-zero IPX network number

- For LANs it uses the network interface's MAC address as the circuit's IPX node number
- For WANs it uses the configured IPX host number as the circuit's IPX node number
- Allows concurrent use of RIP/SAP and Static routes and services.

IPXWAN point-to-point circuits have the following characteristics:

- Can be used on WAN interfaces only
- Might not be restricted to a single IPXWAN point-to-point circuit per interface
- Uses IPXWAN to negotiate parameters
- Might not require an IPX network number
- Uses an IPXWAN NodeID followed by 0000 as the circuit's IPX node number
- Restricted to a single negotiated routing type.

The following sections describe the modeling of each type of supported network interfaces.

LANs (Token-Ring, Ethernet, ATM LAN Emulation)

The IPX routing software models a LAN interface as a single IPX broadcast circuit.

The circuit must be assigned a unique non-zero IPX network number.

The network interface's MAC address serves as the circuit's IPX node number.

The LAN all-stations address (x'FFFFFFFFFFFF') is used to receive and transmit broadcast packets, such as RIP and SAP updates.

The normal encapsulation types are supported for the appropriate type of LAN interface.

The IPX maximum packet size is derived from the MTU configured for the interface.

For token-ring interfaces, source-routing can be enabled on the interface to allow the IPX forwarder to reach end-stations (and other routers) across source-route bridges.

Any or all of the following routing types may be used on the circuit:

- Static Routes/Services
- RIP/SAP (Numbered)

Native ATM

The IPX routing software models an ATM interface as a single IPX broadcast circuit. As such, the underlying ATM PVCs and SVCs defined by the user to interconnect routers on the ATM network are transparent to the IPX routing software.

The circuit must be assigned a unique IPX non-zero network number.

The ESI component of the ATM address is used as the circuit's IPX node number. The burned-in ESI will be used if an ESI has not been configured in the IPX ATM ARP-client associated with the ATM interface.

The LAN all-stations address (x'FFFFFFFFFFFF') serves as the IPX broadcast address. Packets addressed to the broadcast address are transmitted on all VCs on the interface by the underlying ATM DLC.

The IPX maximum packet size is derived from the MTU configured for the interface.

The underlying ATM DLC uses ATM InARP to map destination IPX node addresses to the appropriate ATM VC. Optionally, destination IPX node addresses can be statically configured for VCs connected to routers which do not support ATM InArp.

In order to support non-fully meshed ATM topologies, split-horizon can be disabled on the circuit. This allows RIP and SAP to propagate information to all VCs on the interface so that intermediate routing between VCs on the same interface can occur.

Fully-meshed ATM topologies need not disable split-horizon.

Any or all of the following routing types may be used on the circuit:

- Static Routes/Services
- RIP/SAP (Numbered)

Point-to-Point Protocol (PPP)

The IPX routing software models a PPP interface as either a single IPX broadcast circuit or a single IPXWAN point-to-point circuit.

The IPX maximum packet size is derived from the MTU negotiated by the underlying PPP DLC.

IPX Broadcast Circuit: When configured as a broadcast circuit, the circuit must be assigned a unique non-zero network number.

Since there is no MAC address associated with a PPP interface, a configured host number is used as the circuit's IPX node number.

Any or all of the following routing types may be used on the circuit:

- Static Routes/Services
- RIP/SAP (Numbered)

IPXWAN Point-to-Point Circuit: When configured as an IPXWAN point-to-point circuit, uses IPXWAN to negotiate routing parameters.

The IPXWAN numbered RIP routing type requires a unique non-zero network number to be assigned to the circuit. The other IPXWAN routing types (unnumbered RIP, static routing) do not require a network number (value 0).

Because there is no MAC address associated with the PPP interface, the IPXWAN node id followed by 0000 is used as the circuit's IPX node number.

The routing type to be negotiated on the circuit is configurable. If static routing is enabled, no other routing type will be negotiated. Any or all of the following remaining types can be enabled and will be negotiated down to a single routing type in descending order of preference:

- Unnumbered RIP/SAP
- Numbered RIP/SAP

Frame Relay

The IPX routing software models a Frame Relay interface as:

- a single IPX broadcast circuit, or
- a set of one or more IPXWAN point-to-point circuits, or

- a combination of both.

The IPX maximum packet size is derived from the MTU configured for the interface.

The underlying Frame Relay DLC uses InARP to map destination IPX node addresses to the appropriate Frame Relay virtual circuit. Optionally, destination IPX node addresses can be statically configured for VCs connected to routers which do not support InArp.

IPX Broadcast Circuit: All virtual circuits on the Frame Relay interface which are not configured as IPXWAN point-to-point circuits are grouped together and modeled as a single IPX broadcast circuit which must be assigned a unique non-zero network number. As such, the underlying virtual circuits defined by the user to interconnect routers on the Frame Relay network are transparent to the IPX routing software.

Because there is no MAC address associated with a Frame Relay interface, a configured host number is used as the circuit's IPX node number.

The LAN all-stations address (x'FFFFFFFFFFFF') serves as the IPX broadcast address on the circuit. Packets addressed to the broadcast address are transmitted on all VCs in the IPX broadcast circuit by the underlying Frame Relay DLC. This Frame Relay protocol-broadcast function is activated by enabling the following Frame Relay configuration options:

- Protocol Broadcast
- Multicast Emulation

In order to support non-fully meshed Frame Relay topologies, split-horizon can be disabled on the IPX broadcast circuit. This allows RIP and SAP to propagate information to all virtual circuits in the IPX broadcast circuit so that intermediate routing between virtual circuits in the same IPX broadcast circuit can occur.

Fully-meshed Frame Relay topologies need not disable split-horizon.

Any or all of the following routing types may be used on the circuit:

- Static Routes/Services
- RIP/SAP (Numbered)

IPXWAN Point-to-Point Circuit: IPX can be configured to operate as IPXWAN point-to-point circuits over individual Frame Relay PVCs and SVCs. IPXWAN is used to negotiate routing parameters.

The IPXWAN numbered RIP routing type requires a unique non-zero network number to be assigned to the circuit. The other IPXWAN routing types (unnumbered RIP, static routing) do not require a network number (value of 0).

Because there is no MAC address associated with the Frame Relay interface, the IPXWAN node id followed by 0000 is used as the circuit's IPX node number.

The routing type to be negotiated on the circuit is configurable. If static routing is enabled, no other routing type will be negotiated. Any or all of the following remaining types can be enabled and will be negotiated down to a single routing type in descending order of preference:

- Unnumbered RIP/SAP
- Numbered RIP/SAP

X.25

The IPX routing software models an X.25 interface as a single IPX broadcast circuit. As such, the underlying VCs defined by the user to interconnect routers on the X.25 network are transparent to the IPX routing software.

The circuit must be assigned a unique IPX non-zero network number.

Since there is no MAC address associated with an X.25 interface, a configured host number is used as the circuit's IPX node number.

The LAN all-stations address (x'FFFFFFFFFFFF') serves as the IPX broadcast address on the circuit. Packets addressed to the broadcast address are transmitted to all destination X.25 addresses in the IPX broadcast circuit by the underlying X.25 DLC.

The IPX maximum packet size is derived from the MTU configured for the interface.

In order to support non-fully meshed X.25 topologies, split-horizon can be disabled on the IPX broadcast circuit. This allows and SAP to propagate information to all destination X.25 addresses in the IPX broadcast circuit so that intermediate routing between VCs in the same IPX broadcast circuit can occur.

Fully-meshed X.25 topologies need not disable split-horizon.

Any or all of the following routing types may be used on the circuit:

- Static Routes/Services
- RIP/SAP (Numbered)

Destination IPX node addresses must be statically configured for all destination X.25 addresses, since the X.25 DLC does not support InArp.

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 IPX globally.

```
IPX config> enable ipx
```

3. Add a broadcast-circuit on WAN or LAN, or an IPXWAN circuit on WAN.

```
IPX Config>add broadcast-circuit
Which interface [0]? 1
IPX circuit number[3]? 5
IPX network number in hex
('0' is only allowed on IPXWAN unnumbered circuits) [1]? 01

IPX Config>add ipxwan-circuit
Which interface [0]? 2
IPX circuit number[4]? 6
IPX network number in hex
('0' is only allowed on IPXWAN unnumbered circuits) [1]? 40
Use Frame Relay PVC ? no
Frame Relay SVC circuit name ? Indianapolis
```

Note: IPX network number 0 is valid only on IPXWAN unnumbered RIP or static routing circuits. IPX network number FFFFFFFF is not a valid IPX

network number. IPX network number FFFFFFFE is reserved for the IPX Default Route and may not be used as an IPX network number.

4. If you have enabled IPX to run over a serial circuit, 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, Token Ring or ATM LAN Emulation Client. You do not have to set the frame type for circuits other than Ethernet, Token Ring or ATM LAN Emulation Client. See “Frame” on page 666 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
IPX circuit number [1]? 2
```

6. Optionally change any IPXWAN parameters for which you do not want to use the default values.

```
IPX config> set ipxwan
IPX circuit number [1]? 3
Routing type ('u'=Unnumbered, 'r'=RIP, 'b'=Both, 's'=Static) [u] r
Connection Timeout (in sec) [60]? 90
Retry timer (in sec) [60]? 45
```

Optional Configuration Tasks

Optional settings that you can adjust are described in the following sections.

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 circuits. 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 circuits.

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
IPX circuit number [1]? 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-circuit basis. All router circuits 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 circuits.

Valid entries remain in the SAP services table for a period of three multiples of the SAP update interval, and the router broadcasts its SAP services table information once every update interval.

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
IPX circuit number [1]? 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 53, 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.

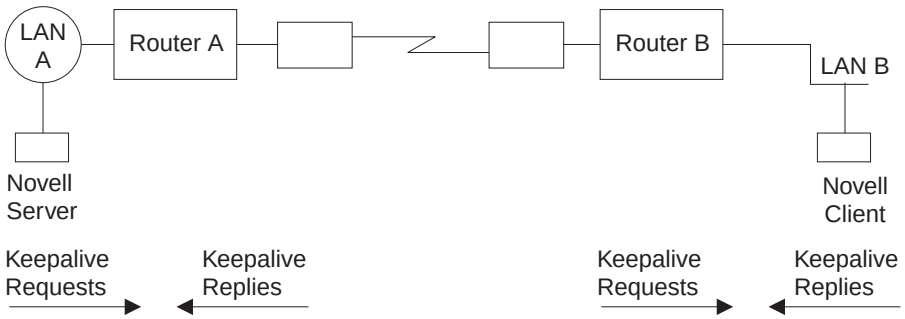


Figure 53. Keepalive Filtering

To set up Keepalive filtering, enable it on the dial circuits.

```
IPX Config> enable keepalive-filtering
IPX circuit number [1]? 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
```

On an IPX circuit basis, you may configure the RIP circuit cost (in ticks). The circuit cost is considered when calculating the total route cost in route advertisements. If multiple routes to the same destination exist, you can influence route selection by assigning a higher route cost to one IPX circuit than to another. Use the following command to set the circuit cost for a specific directly connected route.

```
IPX config> set rip-ticks
IPX circuit number [1]? 2
RIP ticks value (in 55msec ticks [0]? 3
```

Configuring Static Routes

Static routes can be configured per destination network number. Each static route is associated with a circuit and is installed in the routing table when IPX is activated on the circuit. The static route is removed from the routing table when IPX is deactivated on the circuit, the circuit itself is taken down, or any dynamically-learned route to the destination network is learned. Dynamically-learned routes (via RIP) always override static routes. The static route will be reinstalled in the routing table when IPX is reactivated on the circuit, the circuit itself comes back up, or when all RIP routes to the destination network are lost.

Static routes are particularly useful over dial-on-demand circuits where RIP is disabled and routes to destination networks are statically configured on the dial-on-demand circuit.

Static routing may be used on a circuit by itself or in combination with RIP. The only exception to this is when static routing is enabled on an IPXWAN circuit. In this case, static routing is the only routing type negotiated by IPXWAN.

Static routes will be advertised by RIP, subject to split-horizon and applicable filters.

When multiple static routes per destination network are configured, the same rules used to choose RIP routes are used to determine which static routes are installed in the routing table. Multiple static routes to the same destination network will be installed in the routing table if they are of equal cost. Up to the configured routes per destination can be concurrently stored in the routing table.

The following example shows how to configure an IPX static route.

```
IPX Config> disable rip
IPX circuit number [1]? 2

IPX Config> enable route-static

IPX Config> add route-static
IPX net address: (1-ffffffe) [1]? 30
IPX circuit number [1]? 2
Next-hop address, in hex [] ? 400000003000
Ticks: (0-30000) [0]? 4
Hops: (0-14) [0]? 4
```

Configuring Static Services

Static services can be configured per service type or name pair. Each static service is associated with a circuit and is installed in the SAP services table when IPX is activated on the circuit, and a route to the service's network is known (either by

static route or RIP advertisement). The static service is removed from the SAP table when IPX is deactivated on the circuit, the circuit itself is taken down, the route to the server's network is lost, or the same service is learned dynamically. As long as a route to the server's network is known, the static service will be reinstalled in the service table when IPX is reactivated on the circuit, the circuit itself comes back up, or when the SAP-learned service is lost. Dynamically-learned services (using SAP) always override static services.

Static services are particularly useful over dial-on-demand circuits where SAP is disabled and services are statically configured on the dial-on-demand circuit.

Static services may be used on a circuit by itself or in combination with RIP/SAP. The only exception to this is when static routing is enabled on a IPXWAN circuit. In this case, static routing is the only routing type negotiated by IPXWAN.

Static services will be advertised by SAP, subject to split-horizon and applicable filters.

When multiple static services per name or type are configured, the same rules used to choose SAP services are used to determine which static service is installed in the routing table. Note that if there are equal-cost static services configured, the one defined on the same circuit as the current route to the server's network will be installed in the service table.

The following example shows how to configure an IPX static service.

```
IPX Config> disable sap
IPX circuit number [1]? 2

IPX Config> enable sap-static

IPX Config> add sap-static
Sap type: (0-ffff) [4]?
Sap name: []? FILE_SERVER01
IPX circuit number [1]? 2
IPX net address: (1-fffffffe) [1]? 30
IPX node address, in hex: []? 400000202000
IPX socket: (0-ffff) [451]?
Hops: (0-14) [0]? 4
```

Configuring the RIP Default Route

The default route is a special case of a static route. It is used as a last resort as a next hop for unknown destination networks.

The default route is especially useful on dial-on-demand circuits when RIP is disabled. Configuring the default route on the dial-on-demand circuit allows clients to request routes and send packets to destination networks on the other side of the circuit without having to configure a static route for each destination.

RIP Handling

For routers using RIP, the default route is designated by network number FFFFFFFE.

When advertising RIP routes, the default route (like any other static route) will be advertised, after being subjected to the RIP filters and split-horizon.

When responding to a RIP request for an unknown destination network, the router responds to the request only if it has a default route in the routing table.

When forwarding packets, if the route to the destination network is unknown, the forwarder will forward the packet to the next-hop router that is advertising the default route (or the next-hop router indicated by the local static default route definition in the case of static routing).

The following example shows how to configure a RIP default route.

```
IPX Config> enable route-static

IPX Config> add route-static
IPX net address: (1-fffffffe) [1]? ffffffe
IPX circuit number [1]? 2
Next-hop address, in hex: []? 400000003030
Ticks: (0-30000) [0§? 4
Hops: (0-14) [0]? 4
```

Interaction with SAP

Generally, SAP advertisements are accepted only if a route to the server's network is known. If the route to the server's network is not known, but a default route is known, the advertisement is also accepted (after being subjected to the SAP filters).

SAP advertisements that are accepted by virtue of the existence of the default route will be advertised on all IPX circuits other than the one from which the SAP advertisement was accepted (split-horizon). Of course, the advertisement will be subjected to the SAP filters before being advertised. The same rules apply to responses to SAP requests.

Configuring Global IPX Filters (IPX Access Controls)

Global IPX filters are applied to all IPX circuits. 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 (X'0453' and X'0452'). RIP and SAP are required to correctly forward IPX packets.
- Remember that the global filter list applies to all circuits. 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 000000000001. 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 circuit filters are mutually-exclusive. If global SAP filtering is enabled, circuit SAP filters cannot be enabled (and vice versa). If global IPX filtering is enabled (*access-controls*), circuit 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 circuits. 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 Circuit Filters - Overview

The IPX routing feature supports four types of circuit-based filters: ROUTER, RIP, SAP, and IPX. One *input* and one *output filter* can be defined per circuit. 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 circuits.

Note: The global filters and circuit filters are mutually-exclusive. If global SAP filtering is enabled, circuit SAP filters cannot be enabled (and vice versa). If global IPX filtering is enabled (*access-controls*), circuit IPX filters cannot be enabled (and vice versa).

Configuring IPX Circuit Filters

To configure IPX circuit 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 circuit using the **create filter** command, specifying whether it is an input or output filter.
4. Enable IPX circuit filtering using the **enable all** 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 an IPX circuit, disable a filter on an IPX circuit (or all IPX circuits), 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 circuit. 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 circuit. 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 circuit. 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 circuit. 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 circuit. 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*** circuits.

IPX Filters (input) are kept in ordered lists of items by circuit. 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 circuit, and therefore might allow a received packet to be forwarded out on one circuit but not out on some other circuit.

IPX filters (output) are kept in ordered lists of items by circuit. 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 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 Circuit-Filter Configuration Commands” on page 678 describes in more detail the commands used to configure IPX circuit 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 54 on page 650, 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 54 on page 650, 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
```

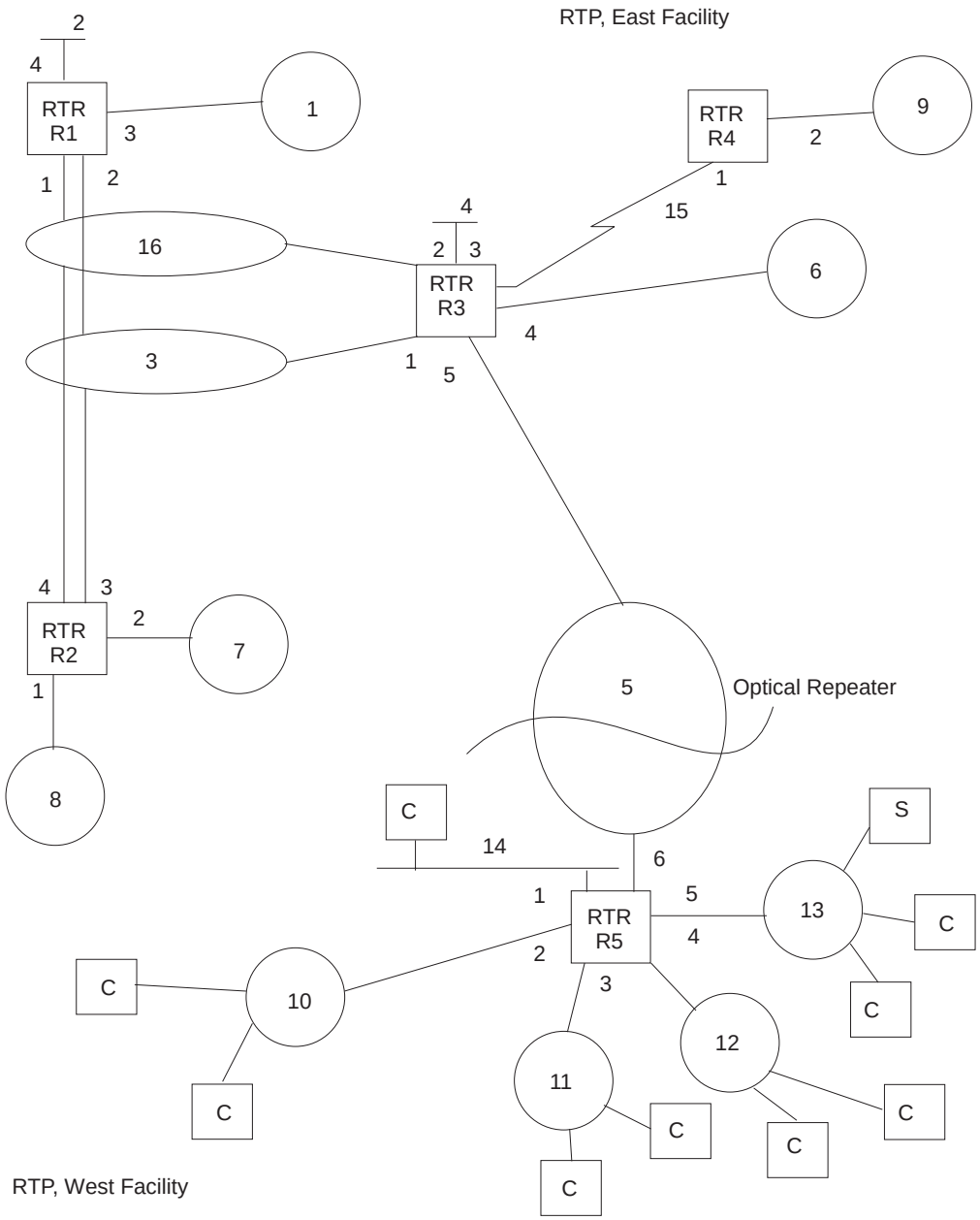


Figure 54. Sample IPX Network

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 circuit 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 55, 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 circuit at headquarters, and enabled at each of the branches to keep them from generating unnecessary traffic.

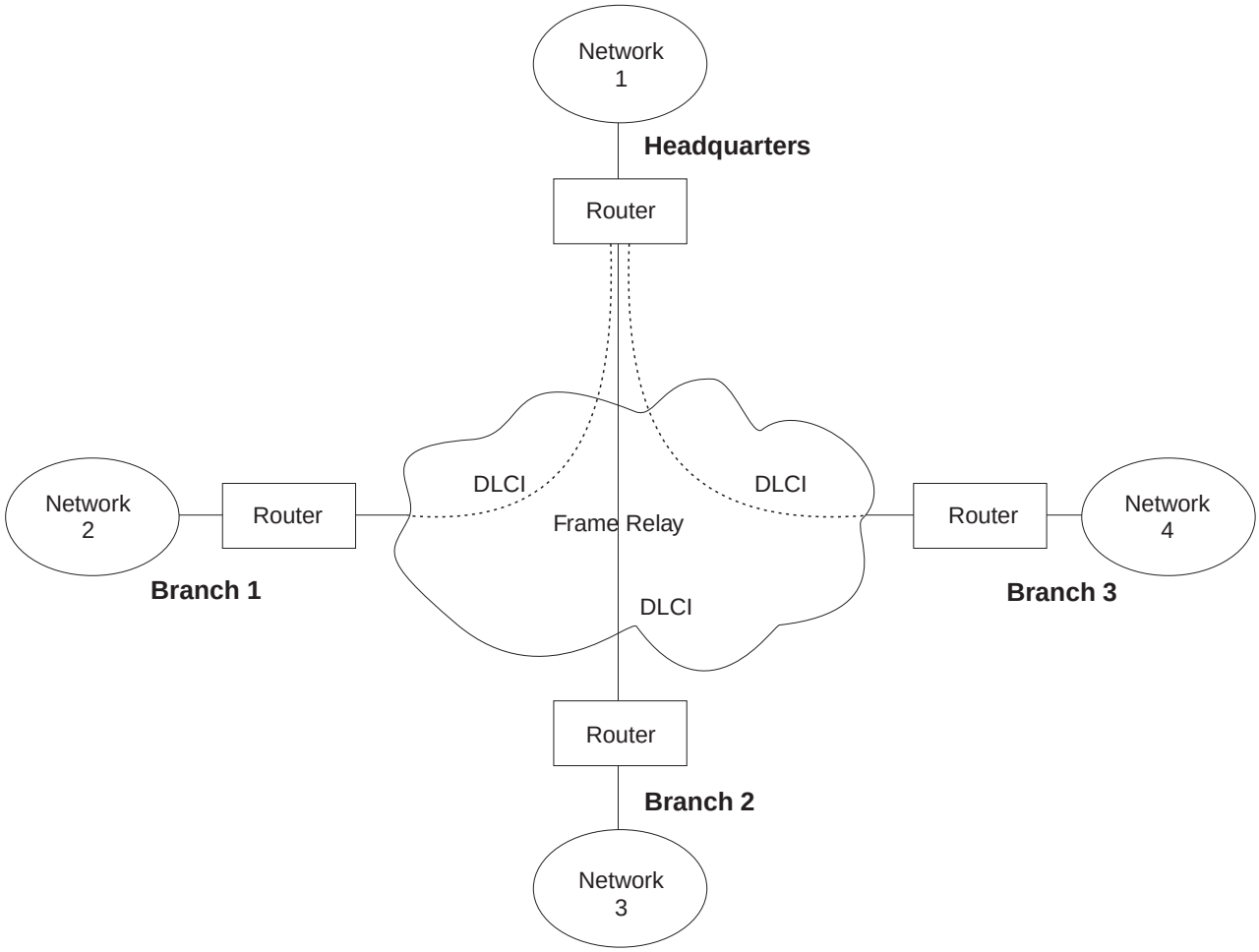


Figure 55. Partially Meshed Frame-Relay Network

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 circuit [1]? 2

IPX Config>set split-horizon disabled
Which circuit [1]? 2

IPX Config>set split-horizon heuristic
Which circuit [1]? 2
```

Chapter 31. Configuring and Monitoring IPX

This chapter describes how to configure the IPX protocol and use the IPX monitoring commands. It includes the following sections:

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 42 lists the IPX configuration commands. These commands specify the network parameters for routers transmitting IPX packets. These commands are entered at the `IPX config>` prompt. To activate the configuration changes, restart the router.

Table 42. IPX Configuration Commands Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page xxxii.
Add	Adds an IPX broadcast or IPXWAN point-to-point circuit, adds global IPX filters (access controls), global SAP filters, static routes or services.
Delete	Deletes an IPX broadcast or IPXWAN point-to-point circuit, deletes global IPX filters (access controls), global SAP filters, static routes, or services.
Disable/Enable	Disables or enables IPX globally or on specific IPX circuits, globally disables or enables the use of IPX static routes or services. Disables or enables Keepalive filtering, RIP-SAP broadcast pacing, SAP reply to get nearest server, NetBIOS broadcasts, and disables or enables RIP or SAP on specific circuits.
Filter-lists	Accesses the IPX circuit-filter configuration. This environment is where the IPX circuit-based ROUTER, RIP, SAP, and IPX filters are configured.
Frame	Specifies the data link format for Ethernet, Token-Ring circuits . This also applies to Token-Ring and Ethernet LAN Emulation Clients.
List	Displays the current IPX configuration.
Move	Reorders the global IPX filter items (access control), or moves an IPX circuit from one interface to another.
Set	Sets the host number, IPXWAN router name and node ID, IPXWAN routing type, 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, RIP circuit cost (RIP ticks), Keepalive filtering table size, and split-horizon usage.
Exit	Returns you to the previous command level. See "Exiting a Lower Level Environment" on page xxxii.

Add

Use the **add** command to add a global IPX filter (access controls), an IPX broadcast circuit, a global SAP filter, an IPX point-to-point circuit, or a static route or service to your IPX configuration.

Syntax:

add
- access-control . . .
- broadcast-circuit . . .
- filter . . .
- ipxwan-circuit . . .
- route-static . . .
- sap-static . . .

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'00000000' to X'FFFFFFFF'

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'000000000000' to X'FFFFFFFFFFFF'

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'00000000' to X'FFFFFFFF'

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'000000000000' to X'FFFFFFFFFFFF'

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.

broadcast-circuit *interface# ipx-circuit# network#*
Adds an IPX broadcast circuit.

interface#
Specifies the network interface on which the IPX circuit number is configured.

Valid Values: valid network interface number

Default: 0

ipx-circuit#
Specifies the IPX circuit number. This number must be unique among all configured IPX circuits in the router and is used to reference IPX circuits in many of the configuration commands.

Valid Values: 1- 65535

Default: next available IPX circuit number

network#
Specifies the IPX network number to be used on the IPX circuit. IPX network number 0 is valid only on IPXWAN unnumbered RIP or static routing circuits. IPX network number FFFFFFFF is not a valid IPX network number. IPX network number FFFFFFFE is reserved for the IPX Default Route and may not be used as an IPX network number.

Valid Values: 1 - FFFFFFFD

Default: 1

Example:

```
add broadcast-circuit
Which interface [0]?
IPX circuit number [1]?
IPX network number in hex
(0 is allowed only on IPXWAN unnumbered circuits) [1]? 400
```

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 to 16.

Default Value: 1

Service-type
Numeric service class.

Valid Values: A hexadecimal value in the range of X'0000' to X'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').

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.

ipxwan-circuit *interface# ipx-circuit# network# [use-PVC] [FR-circ#]*
Adds an IPXWAN point-to-point circuit.

interface#
Specifies an existing PPP or Frame Relay interface on which the IPX circuit should be configured.

Valid Values: valid network interface number

Default: 0

ipx-circuit#
Specifies the IPX circuit number. This number must be unique among all configured IPX circuits in the router and is used to reference IPX circuits in many of the configuration commands.

Valid Values: 1- 65535

Default: next available IPX circuit number

network#
Specifies the IPX network number to be used on the IPX circuit. IPX network number 0 is valid only on IPXWAN unnumbered RIP or static routing circuits. IPX network number FFFFFFFF is not a valid IPX

network number. IPX network number FFFFFFFE is reserved for the IPX Default Route and may not be used as an IPX network number.

Valid Values: 0 - FFFFFFFD

Default: 1

use-PVC
This parameter is only required if the IPXWAN circuit is being configured on a Frame Relay interface. Specifies whether the IPXWAN circuit is being configured on a Frame Relay PVC or SVC. 'Yes' means the IPXWAN circuit is being configured on a PVC. 'No' means the IPX circuit is being configured on an SVC.

Valid Values: Yes or No

Default: Yes

FR-circ#
This parameter is only required if the circuit is being configured on Frame Relay. If the IPXWAN circuit being configured is a Frame Relay PVC, then the parameter specifies the Frame Relay PVC circuit number. If the IPXWAN circuit being configured is a Frame Relay SVC, then the parameter specifies the Frame Relay SVC circuit name.

Valid Values: a valid Frame Relay PVC circuit number or Frame Relay SVC circuit name

Default: 16 (PVC) or None (SVC)

Example:

```
add ipxwan-circuit
Which interface [0]? 2
IPX circuit number [1]? 3
IPX network number in hex
(0 is allowed only on IPXWAN unnumbered circuits) [0]? 412
Use Frame Relay PVC ? yes
Frame Relay PVC circuit number [16]?

add ipxwan circuit
Which interface [0]? 3
IPX circuit number [2]? 4
IPX network number in hex
(0 is allowed only on IPXWAN unnumbered circuits) [0]? 413
Use Frame Relay PVC ? No
Frame Relay SVC circuit name ? Indianapolis
```

route-static *dest-net ipx-circuit# nextHop ticks hops*
Adds a static route.

dest-net
Specifies the destination IPX network number.

Valid Values: X'1' to X'FFFFFFFE'

Default Value : 1

ipx-circuit#
Specifies an existing IPX circuit on which the static route should be configured.

Valid Values: existing IPX circuit number

Default Value: 1

nextHop
Specifies the IPX host number of the next-hop router through which the destination network can be reached.

Valid Values: X'1' to X'FFFFFFFFFFFE'

Default Value: none

ticks
Indicates the number of ticks between the destination network and this router. The number of ticks represents the amount of time it takes to transmit a 576-byte IPX packet from this router to the destination network. Each tick is 55 milliseconds.

Valid Values: 0 to 30000

Default Value: 0

hops
Indicates the number of hops between the destination network and this router.

Valid Values: 0 to 14

Default Value: 0

Example:

```
add route-static
IPX net address: (1-ffffffe) [1]? 30
IPX circuit number [1]? 2
IPX node address (in hex) []? 020000002030
Ticks: (0-3000) [0]? 4
Hops: (0-14) [0]? 4
```

sap-static *serviceType serviceName ipx-circuit# serverNet serverNode serverSocket hops*
Adds a static SAP service.

serviceType
Specifies the hexadecimal service class of the service.

Valid Values: X'0' to X'FFFF'

Default Value: 4

serviceName
Specifies the ASCII name of the service.

Valid Values: up to 47 of the following ASCII characters: ’A’–’Z’, ’a’–’z’, ’0’–’9’, ’_’, ’-’, ’@’.

Default Value: None

ipx-circuit#
Specifies an existing IPX circuit on which the SAP static service should be configured.

Valid Values: existing IPX circuit number

Default Value: 1

serverNet
Specifies the internal IPX network number or home IPX network number of the server.

Valid Values: X'1' to X'FFFFFFFE'

Default Value: 1

serverNode
Specifies the IPX node of the server.

Valid Values: X'1' to X'FFFFFFFFFFFE'

Default Value: None

serverSocket
Specifies the socket number of the server.

Valid Values: X'0' to X'FFFF'

Default Value: 451

hops
Indicates the number of hops between the server and this router.

Valid Values: 0 to 14

Default Value: 0

Example:

```
add sap-static
Sap type: (0-ffff) [4]? 4
IPX circuit number [1]? 2
IPX net address: (1-fffffffe) [1]? 40
IPX node address, in hex:  []? 000000000001
IPX socket:  (0-ffff) [451]?
Hops: (0-14) [0] 4
```

Delete

Use the **delete** command to delete an IPX broadcast or IPXWAN point-to-point circuit, a global IPX filter (access control), a global SAP filter, a static route or a static service.

Syntax:

delete access-control . . .
circuit . . .
filter . . .
route-static . . .
sap-static . . .

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`

circuit *ipx-circuit#*
Deletes the IPX broadcast or IPXWAN point-to-point circuit. It will also delete all of the static routes, static services and circuit filters that are associated with the specified *ipx-circuit#*.

Example: **`delete circuit`**

```
IPX circuit number [1]? 2
You are about to delete IPX broadcast circuit 2 on interface 4.
All associated static routes, static services and circuit filters
will be deleted as well.  Are you sure? [Yes]: yes
```

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').

Default Value: None

Example: **`delete filter 2 039B NOTES-CHICAGO`**

route-static *dest-net ipx-circuit# nextHop*

Deletes a static route.

dest-net

Specifies the destination IPX network number.

Valid Values: X'1' to X'FFFFFFFE'

Default Value: 1

ipx-circuit#

Specifies the IPX circuit on which the static route is configured.

Valid Values: existing IPX circuit number

Default Value: 1

nextHop

Specifies the IPX host number of the next-hop router through which the destination network can be reached.

Valid Values: X'1' to X'FFFFFFFFFFFE'

Default Value: none

Example:

```
delete route-static
IPX net address: (1-ffffffe) [1]? 30
IPX circuit number [1]? 2
IPX node address (in hex) []? 020000002030
```

sap-static *serviceType serviceName ipx-circuit#*

Deletes a static SAP service.

serviceType

Specifies the hexadecimal service class of the service.

Valid Values: X'0' to X'FFFF'

Default Value: 4

serviceName

Specifies the ASCII name of the service.

Valid Values: up to 47 of the following ASCII characters: ’A’–’Z’, ’a’–’z’, ’0’–’9’, ’_’, ’-’, ’@’.

Default Value: None

ipx-circuit#

Specifies the IPX circuit on which the SAP static service is configured.

Valid Values: existing ipx-circuit number

Default Value: 1

Example:

```
delete sap-static
Sap type: (0-ffff) [4]?
Sap name: (0-ffff) []? filesrv1
IPX circuit number [1]? 2
```

Disable

Use the **disable** command to disable globally or on specific IPX circuits, globally disables the use of IPX static routes and services. Also, use the **disable** command to disable replies to SAP to get-nearest-server , RIP-SAP Broadcast Pacing, RIP, or SAP on specific circuits.

Syntax:

disable
- circuit . . .
- ipx
- keepalive-filtering . . .
- nebios-broadcast . . .
- reply-to-get-nearest-server . . .
- rip . . .
- rip-sap-pacing . . .
- route-static . . .
- sap . . .
- sap-static . . .

circuit *ipx-circuit#*

Disables the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit.*

Example: `disable circuit`

```
IPX circuit number [1]? 2
```

ipx Globally disables the IPX protocol.

Example: **`disable ipx`**

keepalive-filtering *ipx-circuit#*
Disables Keepalive-filtering on the IPX broadcast circuit or IPXWAN point-to-point circuits specified by *ipx-circuit#*.

Example: **`disable keepalive-filtering`**

```
IPX circuit number [1]? 2
```

netbios-broadcast *ipx-circuit#*
Disables receiving and sending Novell NetBIOS broadcasts (packet type 20) on the IPX circuit specified by *ipx-circuit#*. The default is value is enabled. Receiving and sending Novell NetBIOS broadcasts is automatically disabled on IPXWAN static routing circuits, even if it is enabled in the configuration.

Example: **`disable netbios-broadcast`**

```
IPX circuit number [1]? 2
```

reply-to-get-nearest-server *ipx-circuit#*
Prevents the router from responding to SAP get-nearest-server requests on the IPX broadcast circuit or IPXWAN point-to-point circuit specified by *ipx-circuit#.*

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-to-get-nearest`**

```
IPX circuit number [1]? 2
```

rip *ipx-circuit#*
Disables RIP on the IPX broadcast circuit or IPXWAN point-to-point circuit specified by *ipx-circuit#*. The default is for RIP to be enabled on all circuits. RIP will automatically be disabled on circuits using IPXWAN Static Routing, even if it is configured as enabled.

Example: **`disable rip 1`**

rip-sap-pacing *ipx-circuit#*
Prevents RIP/SAP Broadcast Pacing on the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit#*. When pacing is disabled, RIP and SAP periodic broadcasts are transmitted on the circuit with a 55 msec interpacket gap (the default setting). Enable pacing only on circuits where RIP and SAP broadcasts might cause congestion (for example, you can enable pacing on frame-relay or X.25 circuits with many virtual circuits).

Example: **`disable rip-sap-pacing`**

```
IPX circuit number [1]? 2
```

route-static
Globally disables the use of static routes.

Example: **`disable route-static`**

sap *ipx-circuit#*
Disables SAP on the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit*. The default is for SAP to be enabled on all circuits.

SAP will automatically be disabled on RLAN circuits and on IPXWAN Static Routing, even if SAP is configured as enabled.

Example: `disable sap`

```
IPX circuit number [1]? 2
```

sap-static
Globally disables the use of static services.

Example: `disable sap-static`

Enable

Use the **enable** command to enable IPX globally or on specific circuits. The enable command can also be used to globally enable the use of IPX static routes or services, enables keepalive filtering, RIPS-SAP broadcast pacing, SAP reply to get-nearest-server, RIP or SAP on specific circuits.

Syntax:

enable	circuit . . .
	ipx
	keepalive-filtering . . .
	nebios-broadcast . . .
	reply-to-get-nearest-server . . .
	rip . . .
	rip-sap-pacing . . .
	route-static . . .
	sap . . .
	sap-static . . .

circuit *ipx-circuit# network#*
Enables the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit#* and specifies the IPX network number for the IPX circuit. The IPX circuit will be enabled if a valid IPX network number is configured.

Example: enable circuit

```
IPX circuit number [1]?
IPX network number in hex
(0 is allowed only on IPXWAN unnumbered circuits) [1]?
```

ipx-circuit#
Specifies the IPX broadcast or IPXWAN point-to-point circuit to be enabled.

Valid Values: any valid ipx-circuit number

Default Value: 0

network#
Specifies the IPX network to be used on the circuit. IPX network number 0 is valid only on IPXWAN unnumbered RIP or static routing circuits. IPX network number FFFFFFFF is not a valid IPX network number. IPX network number FFFFFFFE is reserved for the IPX Default Route and may not be used as an IPX network number.

Valid Values: X'0' to X'FFFFFFFD'

Default Value: 1

Example:

ipx Globally enables the IPX protocol.

Example: **`enable ipx`**

keepalive-filtering *ipx-circuit#*
Enables Keepalive filtering on the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit#.*

Example: **`enable keepalive-filtering`**

```
IPX circuit number [1]? 2
```

netbios-broadcast *ipx-circuit#*
Enables receiving and sending Novell NetBIOS broadcasts (packet type 20) on the IPX circuit specified by *ipx-circuit#*. The default value is enabled. Receiving and sending Novell NetBIOS broadcast is automatically disabled on IPXWAN static routing circuits, even if enabled in the configuration.

Example: **`enable netbios-broadcast`**

```
IPX circuit number [1]? 2
```

reply-to-get-nearest-server *ipx-circuit#*
Enables the router to respond to SAP get-nearest-server requests on the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit#*.

Example: **`enable reply-to-get-nearest`**

```
IPX circuit number [1]? 2
```

rip *ipx-circuit#*
Enables RIP on the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit#*. The default is for RIP to be enabled on all IPX circuits. RIP is automatically disabled on RLAN circuits and on IPXWAN static routing circuits, even if RIP is enabled in the configuration.

Example: **`enable rip`**

```
IPX circuit number [1]? 2
```

rip-sap-pacing *ipx-circuit#*
Enables RIP/SAP Broadcast Pacing on the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit#*.

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 circuits where RIP and SAP broadcasts might cause congestion (for example, on frame-relay or X.25 circuits with many virtual circuits).

Example: **`enable rip-sap-pacing`**

```
IPX circuit number [1]? 2
```

route-static
Globally enables the use of static routes.

Example: **`enable route-static`**

sap *ipx-circuit#*
Enables SAP on the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit#*.

Example: **`enable sap`**

sap-static

Globally enables the use of static services.

Example: `enable sap-static`

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 Circuit-Filter Configuration Commands" on page 678.

Syntax:

filter-lists	router-lists rip-lists sap-lists ipx-lists

Example: `filter-lists router-lists`

Frame

Use the **frame** command to specify the packet format for IPX circuits. (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 . . .

ethernet_II *ipx-circuit#*

Sets the frame type to ethernet_II on the IPX broadcast circuit specified by *ipx-circuit#*. The ethernet_II encapsulation uses ethernet version 2.0 with protocol type 8137. This is the NetWare 4.0 and greater default.

Example: `frame ethernet_II`

```
IPX circuit number [1]?
```

ethernet_8022 *ipx-circuit#*

Sets the frame type to ethernet_8022 on the IPX broadcast circuit specified by *ipx-circuit#*. The ethernet_8022 encapsulation uses LLC encapsulation with SAP E0.

Example: `frame ethernet_8022`

```
IPX circuit number [1]?
```

ethernet_8023 *ipx-circuit#*

Sets the frame type to ethernet_8023 on the IPX broadcast circuit specified by *ipx-circuit#*. The ethernet_8023 encapsulation uses ethernet 802.3 encapsulation with no LLC header. This is the pre-NetWare 4.0 default. It is also the router default.

Example: **`frame ethernet_8023`**

```
IPX circuit number [1]?
```

ethernet_SNAP *ipx-circuit#*
Sets the frame type to ethernet_SNAP on the IPX broadcast circuit specified by *ipx-circuit#*. The ethernet_SNAP encapsulation uses SNAP encapsulation with a PID of 0000008137.

Example: **`frame ethernet_SNAP`**

```
IPX circuit number [1]?
```

token-ring MSB *ipx-circuit#*
Sets the frame type to token-ring MSB on the IPX broadcast circuit specified by *ipx-circuit#*. The token-ring MSB encapsulation uses LLC encapsulation with SAP E0, and uses noncanonical MAC addresses. This is the NetWare default. It is also the router default.

Example: **`frame token-ring MSB`**

```
IPX circuit number [1]?
```

token-ring LSB *ipx-circuit#*
Sets the frame type to token-ring LSB on the IPX broadcast circuit specified by *ipx-circuit#*. The token-ring LSB encapsulation uses LLC encapsulation with SAP E0, and uses noncanonical MAC addresses.

Example: **`frame token-ring LSB`**

```
IPX circuit number [1]?
```

token-ring_SNAP MSB *ipx-circuit#*
Sets the frame type to token-ring_SNAP MSB on the IPX broadcast circuit specified by *ipx-circuit#*. The token-ring_SNAP MSB encapsulation uses SNAP encapsulation with PID 0000008137, and uses canonical MAC addresses.

Example: **`frame token-ring_SNAP MSB`**

```
IPX circuit number [1]?
```

token-ring_SNAP LSB *ipx-circuit#*
Sets the frame type to token-ring LSB on the IPX broadcast circuit specified by *ipx-circuit#*. The token-ring LSB encapsulation uses SNAP encapsulation with PID 0000008137 and uses noncanonical MAC addresses.

List

Use the **list** command to display the current IPX configuration.

Syntax:

list	access-controls all circuit filters route-static sap-static summary

access-controls
Lists the global IPX filters (access-controls). This command displays the information that is displayed in the “Access Control Configuration” section of the **list all**.

all Lists the entire IPX configuration.

Example:

```
 list all

IPX Globals
------------
IPX Globally                Enabled
Host Number (serial line)   020000003024
Maximum Services                  32
Maximum Networks                  32
Maximum Routes                    32
Maximum Routes per Destination    1
Maximum Local Cache entries       64
Maximum Remote Cache entries      64
Keepalive-Filtering Table Size    32


IPX Configuration:

------------------
                                  NetBIOS    Keepalive
Circ   Ifc    NetNum    IPX       Broadcast  Filtering  Encapsulation
 1      0      400       Enabled   Enabled    Disabled   ETHERNET_II
 2      1      411       Enabled   Enabled    Disabled   N/A
 3      2      412       Enabled   Enabled    Disabled   N/A
        Frame Relay PVC circuit number: 16
 4      3      413       Enabled   Enabled    Disabled   N/A
        Frame Relay SVC circuit name: Indianapolis



RIP Configuration:
------------------
                                  Update    Split      Broadcast  RIP
Circ   Ifc    NetNum    RIP       Interval  Horizon    Pacing     Ticks
1      0      400       Enabled   1         Enabled    Disabled   0
2      1      411       Enabled   1         Enabled    Disabled   3
3      2      412       Enabled   1         Enabled    Disabled   0
4      3      413       Enabled   1         Enabled    Disabled   0



SAP Configuration:
------------------
                                  Update    Split      Broadcast  Get Nearest
Circ   Ifc    NetNum    SAP       Interval  Horizon    Pacing     Reply
1      0      400       Enabled   1         Enabled    Disabled   Enabled
2      1      411       Enabled   1         Enabled    Disabled   Enabled
3      2      412       Enabled   1         Enabled    Disabled   Enabled
4      3      413       Enabled   1         Enabled    Disabled   Enabled


IPXWAN Configuration:
---------------------
Router Name    ipxwan-413
NodeID         413
                          Routing  Connect     Retry
Circ   Ifc    NetNum     Type     Time (sec)  Time (sec)
2      1      411        RIP      60          60
3      2      412        RIP      60          60
4      3      413        RIP      60          60

Static Route Configuration:
---------------------------
Static Routes:  Enabled
Dest Net  Hops   Ticks  Next Hop      Circ   Ifc
ABC       3      4      020000003044  3      2

Static Services Configuration:
------------------------------
Static Services:  Enabled
Type Service Name                    Srv Net  Host         Sock Hops Circ   Ifc
4    FILESRV01                       ABC      000000000001 451  3    3      2

SAP Filter Configuration:
-------------------------
IPX SAP Filters:  Enabled
Index Max Hops  Type  Service Name
1     5         4     FILESRV02

Access Control Configuration:
-----------------------------
IPX Access Controls:  Enabled
#   T Dest Net Host         Sock Sock Src Net  Host         Sock Sock
1   E 2        000000000000 0    FFFF 3        000000000000 0    FFFF
2   I 0        000000000000 452  453  0        000000000000 0    FFFF
```

circuit *ipx-circuit#*
Lists the IPX broadcast or IPXWAN point-to-point circuit specified by *ipx-circuit#.* This command displays the information shown in the “IPX Configuration,” “RIP Configuration,” “SAP Configuration,” and “IPXWAN Configuration” sections of the **list all** command example.

filters Lists the global SAP filters. This command displays the information shown in the “SAP Filter Configuration” section of the **list all** command example.

route-static
Lists the static routes. This command displays the information shown in the “Static Route Configuration” section of the **list all** command example.

sap-static
Lists the static services. This command displays the information shown in the “Static Services Configuration” section of the **list all** command example.

summary
Lists a summary of the IPX, RIP, SAP, IPXWAN, and Keepalive filtering configuration for all circuits on which IPX is enabled. This command displays the information shown in the “IPX Globals,” “IPX Configuration,” “RIP Configuration,” “SAP Configuration,” and “IPXWAN Configuration” sections of the **list all** command example.

IPX Globals
The following global information is displayed:
- Whether IPX is globally enabled or disabled
- IPX host number
- Maximum services
- Maximum networks
- Maximum routes
- Maximum routes per destination
- Maximum local cache entries
- Maximum remote cache entries
- Keepalive-filtering table size

IPX Configuration
The following is displayed for each circuit on which IPX is enabled:
- IPX circuit number
- Network interface number
- IPX network number (Netnum)
- IPX is enabled/disabled on circuit
- NetBIOS Broadcast
- Keepalive filtering
- Encapsulation

PVC circuit number
Displays the number of the Frame Relay PVC circuit.

SVC circuit name
Displays the name of the Frame Relay SVC circuit.

RIP Configuration
The following information is displayed for each circuit on which IPX is enabled:
- IPX circuit number
- Network interface number
- IPX network number (Netnum)
- Whether RIP is enabled or disabled
- RIP update interval timer

- Whether split-horizon is enabled or disabled
- Whether RIP broadcast pacing is enabled or disabled
- IPX route cost (in ticks)

SAP Configuration

The following information is displayed for each circuit on which IPX is enabled:

- IPX circuit number
- Network interface number
- IPX network number (Netnum)
- Whether SAP is enabled or disabled
- SAP update interval timer
- Whether split-horizon is enabled or disabled
- Whether SAP broadcast pacing is enabled or disabled
- Whether reply to SAP get-nearest-server request is enabled.

IPXWAN Configuration

The following global information is displayed:

- Router name
- Node ID

The following information is displayed for each IPXWAN circuit :

- IPX circuit number
- Network interface number
- IPX network number (Netnum)
- Routing type
- Connect timer
- Retry timer

Static Routes Configuration

Displays whether static routes are globally enabled or disabled. In addition, the following is displayed for each configured static route.

- IPX destination network number
- Hops
- Ticks
- Next hop node address
- IPX circuit number
- Network interface number

Static Services Configuration

Displays whether static services are globally enabled or disabled. In addition, the following is displayed for each configured static service:

- Service type
- Service name
- IPX network number of service
- IPX node address of Service (Host)
- Socket
- Hops
- IPX circuit number
- Network interface number

SAP Filter Configuration

Displays whether the global SAP filters are enabled or disabled. In addition, the following information is displayed for each configured global SAP filter:

- Index
- Max hops
- Service type
- Service name

Access Control Configuration

Displays whether the global IPX filters (access controls) are enabled or disabled. In addition, the following information is displayed for each configured global IPX filter (access control):

- Access control index (#)
- Filter type (include or exclude)
- Destination IPX network number
- Destination IPX node number (Host)
- Destination IPX socket range
- Source IPX network number
- Source IPX node number (Host)
- Source IPX socket range

Move

Use the **move** command to reorder the global IPX filter items (access control), or move an IPX circuit from one interface to another.

Syntax:

move access-control *srcLine# dstLine#*

circuit *ipx-circuit# interface# [use-PVC]]FR-circ#]*

access-control *srcLine# dstLine#*

srcLine#
Specifies the line number of the access control you want to move.

dstLine#
Specifies the line number of the access control after which the scrLine should be moved.

After the line is access control is moved, the lines are renumbered.

Example:

```
move access-control
Enter index of control to move [1]? 1
Move record AFTER record number [0]? 2
About to move:
#   T Dest Net Host         Sock Sock Src Net  Host         Sock Sock
1   E 2        000000000000 0    FFFF 3        000000000000 0    FFFF
to be after:
2   I 0        000000000000 452  453  0        000000000000 0    FFFF
Are you sure this is what you want to do? [Yes]: yes
```

circuit *ipx-circuit# interface# [use-PVC FR-circ#]*

Moves an IPX circuit from one network interface to another. This command also moves all of the static routes, static services, and IPX circuit filters associated with the given *ipx-circuit#* to the same *interface#*. If an IPXWAN circuit is being moved to a Frame Relay interface, you are prompted to specify whether the new circuit is a Frame Relay PVC or SVC circuit, and to provide the Frame Relay circuit number or circuit name, as appropriate.

ipx-circuit#
Specifies the IPX circuit that is to be moved.

Valid Values: an existing IPX circuit number

Default Value: 1

interface#
Specifies the network interface that the IPX circuit is moving to.

Valid Values: an existing network interface number.

Default Value: 0

use-PVC
Specifies whether the IPXWAN circuit is being moved to a Frame Relay PVC or SVC. 'Yes' means the IPXWAN circuit is being moved to a PVC. 'No' means the IPXWAN circuit is being moved to an SVC. This parameter is only required if the IPXWAN circuit is being moved to a Frame Relay interface.

Valid Values: Yes or No

Default Value: Yes

FR-circ#
Specifies the Frame Relay PVC circuit number or Frame Relay SVC circuit name. This parameter is required only if the IPX circuit is an IPXWAN circuit being moved to a Frame Relay interface.

Valid Values: existing PVC Frame Relay PVC circuit number or Frame Relay SVC circuit name

Default Value: 16 (PVC) or None (SVC)

Example: move circuit

```
IPX circuit number [1]?
Which interface do you want to move the IPX circuit to []? 5
Use Frame Relay PVC? [Yes]:
Frame Relay PVC circuit number [16]? 18
You are about to move IPXWAN circuit 1,
from Frame Relay interface 2 (FR circuit 16) to
Frame Relay interface 5 (FR circuit 18).
All associated static routes, static services and circuit filters
will be moved as well. Are you sure? [Yes]: Y
```

Set

Use the **set** command to configure the host number, IPXWAN router name and node ID, IPXWAN routing type, connection timeout and retry timer, IPX network numbers, maximum RIP and SAP table sizes, local and remote cache sizes, global IPX filter (access control) and global SAP filter states, RIP and SAP update intervals, IPX route cost (in ticks), Keepalive filter table size and split-horizon usage.

Syntax:

set
- access-control . . .
- filter . . .
- host-number . . .
- ipxwan . . .
- keepalive-table-size . . .
- local-cache size . . .
- maximum routes-per-destination . . .
- maximum networks . . .
- maximum services . . .
- maximum total-route-entries . . .
- name . . .
- net-number . . .
- node-id . . .
- remote-cache size . . .

rip-ticks . . .
rip-update-interval . . .
sap-update-interval . . .
split-horizon . . .

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 circuits running IPX. Each IPX router operating over serial circuits must have a unique host number. This is required because serial circuits do not have hardware node addresses from which to build a host number. It cannot be a multicast address.

Note: If you configure a mixture of IPX broadcast and IPXWAN circuits on the same interface, it is strongly recommended that you configure the host-number to be the IPXWAN node-id followed by X'0000'.

Valid Values: An 12-digit hexadecimal number in the range of X'000000000001' to X'FEFFFFFFFFFF'.

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.

ipxwan *ipx-circuit# routing-type timeout retryTimer*
Sets the IPXWAN routing type, connection timeout and retry timer. Before the **set ipxwan** command can be invoked, you must add an IPXWAN circuit.

ipx-circuit#
Specifies an existing IPXWAN point-to-point circuit on which the parameters will be set.

Valid Values: any existing IPXWAN point-to-point circuit number

Default Value: 1

routingType
Specifies the IPXWAN routing type to be negotiated.

- **u** for unnumbered RIP
- **r** for number RIP
- **b** for both unnumbered and numbered RIP
- **s** Static Routing

Valid Values: ’u’, ’U’, ’r’, ’R’, ’b’, ’B’, ’s’, ’S’

Default Value: 'u'

timeout

This value specifies the time limit, in seconds, within which the IPXWAN negotiation must be successfully completed. If it cannot be successfully completed before the connection timer expires, IPXWAN starts a retry timer. The device will not retry the negotiation until the retry timer expires.

Valid Values: An integer number of seconds in the range of 5 to 300.

Default Value: 60 seconds

retryTimer

This parameter specifies the amount of time to wait after a connection is timed out before trying to reestablish the connection.

Valid Values: An integer number of seconds in the range of 5 to 600.

Default Value: 60 seconds

Example: `set ipxwan`

```
IPX circuit number [1]? 3
Routing type ('u'=Unnumbered, 'r'=RIP, 'b'=Both, 's'=Static) [u]
Connection Timeout (in sec) [60]?
Retry timer (in sec) [60]?
```

keepalive-table-size *value*

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.

Valid Values: 1 to 250

Default: 32

Example: `set keepalive-table-size`

```
Number of entries [32]?
```

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 649 and "Remote Cache" on page 649.

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 to 64.

Default Value: 1. For additional information on multiple routes, see "Configuring Multiple Routes" on page 640.

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 to 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 to 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 640.

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 node id and 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 *ipx-circuit# network#*

Specifies the IPX network number fro the IPX broadcast or IPXWAN point-to-point circuit.

ipx-circuit#

Specifies an existing IPX broadcast or IPXWAN point-to-point circuit.

Valid Values: an existing circuit number

Default Value: 1

network#
Specifies the IPX network number to be used on the IPX circuit. IPX network number 0 is valid only on IPXWAN unnumbered RIP or static routing circuits. IPX network number FFFFFFFF is not a valid IPX network number. IPX network number FFFFFFFE is reserved for the IPX Default Route and may not be used as an IPX network number. The set command will be ignored if a valid IPX network number is not configured.

Valid Values: X'0' to X'FFFFFFFD'

Default Value: 1

Example: `set net-number`

```
IPX circuit number [1]? 2
IPX network number in hex
(0 is allowed only on IPXWAN unnumbered circuits) [1]?
```

node-id *network#*
Specifies the IPXWAN internal network number. A value of 0, FFFFFFFF or FFFFFFFE is not valid for the internal network number. IPXWAN will not be enabled unless a valid node ID is configured.

Default Value: 1

Example: `set node-id 2`

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-ticks *ipx-circuit# value*
Indicates the circuit cost, in ticks, associated with this circuit. The number of ticks represents the amount of time it takes to transmit a 576-byte IPX packet on this IPX circuit. Each tick is 55 milliseconds. A value of 0 indicates that the router will calculate the tick value. A configured non-zero value will override any calculated value, including IPXWAN.

ipx-circuit#
Specifies an existing IPX broadcast to IPXWAN point-to-point circuit.

Valid Values: any valid IPX circuit number

Default: 1

value
Specifies the tick value

Valid Values: The range is 1 to 30000.

Default Value: 0

Example: **`set rip-ticks`**

```
IPX circuit number [1]? 2
RIP ticks value (in 55mec ticks) [0]? 3
```

rip-update-interval *ipx-circuit# interval*

Specifies the interval in minutes at which RIP periodic broadcasts should occur on a specific IPX circuit.

Increasing the RIP interval reduces traffic on WAN lines and dial circuits. It also prevents dial-on-demand circuit 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 about them.

ipx-circuit#

Specifies an existing IPX broadcast to IPXWAN point-to-point circuit.

Valid Values: any valid IPX circuit number

Default: 1

interval

Specifies the interval in minutes

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 639.

Example: **`set rip-update-interval`**

```
IPX circuit number [1]? 2
RIP Timer Value (minutes) [1]? 2
```

sap-update-interval *ipx-circuit# interval*

Specifies the time delay in minutes at which SAP periodic broadcasts should occur on a specific IPX circuit.

Increasing the SAP interval reduces traffic on WAN lines and dial circuits. It also prevents dial-on-demand circuit from dialing out so often.

Note: While complete SAP advertisements are controlled by the interval, the router still propagates service changes as quickly as it learns about them.

ipx-circuit#

Specifies an existing IPX broadcast or IPXWAN point-to-point circuit.

Valid Values: any valid IPX number

Default: 1

interval

Specifies the interval in minutes.

Valid Values: The range is from 1 to 1440 minutes.

Default Value: 1 minute.

Example: **`set sap-update-interval`**

```
IPX circuit number [1]? 2
SAP Timer Value (minutes) [1]? 2
```

split-horizon *heuristic enabled disabled*

Specifies the type of split-horizon used on the IPX circuit.

If there is only a single Frame Relay VC on the circuit, split-horizon is enabled; otherwise split-horizon is disabled.

Generally, split-horizon should be set to *enabled*. It is sometimes necessary to disable split-horizon for partially-meshed broadcast circuits on Frame-Relay, ATM, and X.25 configurations. For additional information on split-horizon, see “Split-Horizon Routing” on page 650.

heuristic

Enables split-horizon on the IPX circuit, except for Frame Relay IPX broadcast circuits.

Valid Values: any valid IPX circuit number

Default: 1

enabled

Enables split-horizon on the IPX circuit.

Valid Values: 1–1440

Default: 1

disabled

Disables split-horizon IPX circuit.

Valid Values: 1–1440

Default: 1

Example: **`set split-horizon enabled 0`**

```
IPX circuit number [1]? 2
```

Accessing the IPX Circuit Filter Configuration Environment

To access the IPX circuit 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-lists, rip-lists, sap-lists,* and *ipx-lists*.

When creating a filter, an IPX circuit number is required.

IPX Circuit-Filter Configuration Commands

This section describes the commands used to configure the IPX circuit-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 43. IPX Filter Configuration Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
Attach	Attaches a specified filter-list to a specified filter.
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	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

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 ipx-circuit#*

Creates a filter for the specified direction on the specified circuit. Specify *input* to filter packets received on the specified circuit. Specify *output* to filter packets to be sent by the specified circuit.

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 circuit 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

```
Filtering: ENABLED

Filter Lists:
Name                           Action
------------------------------ ----------
ipx01                          EXCLUDE
ipx02                          INCLUDE
ipx03                          EXCLUDE

Filters:
Id    Circ  Ifc   Direction  State      Default    Cache
```

```
----- ----- ----- ---------- ---------- ---------- -----
1     3     2     INPUT      ENABLED    INCLUDE       10
2     2     1     INPUT      ENABLED    INCLUDE       10
```

filter *filter#*

Lists information about the specified filter. Use the list command to display a numbered list of configured filters.

Example: list filter 2

```
Filters:
Id    Circ  Ifc   Direction  State      Default    Cache
----- ----- ----- ---------- ---------- ---------- -----
2     2     1     INPUT      ENABLED    INCLUDE       10

Filter Lists:
Name                           Action
------------------------------ ----------
ipx01                          EXCLUDE
```

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 circuit filter; the ROUTER, RIP and SAP circuit filters do not support a cache.

Syntax:

set-cache *size* filter#

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 circuit filter (ROUTER, RIP, SAP, or IPX) being configured. For all types of circuit filter, the **add** command can be entered without parameters. You will then be prompted for the required parameters.

Add (ROUTER)

Syntax:
add *node-number mask*

node-number
Specifies the value to be compared against the source node number of the router which sent the RIP response packet (after being ANDed with the mask). If you want to match on a single node, set the node-number parameter to the address and set the mask to FFFFFFFFFFFF. If you want to match on all nodes, set the node-number parameter and the mask parameter to 000000000000.

Valid Values: X'000000000000' to X'FFFFFFFFFFFF'

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 FFFFFFFFFFFF. If you want to match on all addresses, set the address parameter and the mask parameter to 000000000000.

Valid Values: X'000000000000' to X'FFFFFFFFFFFF'

Default Value: X'FFFFFFFFFFFF'

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 X'00000001' and the net-range-end to X'FFFFFFFE'.

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 X'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').

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-node dst-mask dst-sck-range-start dst-sck-range-end src-net-range-start src-net-range-end src-node 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' to 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 X'00000001' and the dst-net-range-end to X'FFFFFFFE'.

Valid Values: X'00000000' to X'FFFFFFFF'

Default Value: X'00000000'

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 X'00000001' and the dst-net-range-end to X'FFFFFFFE'.

Valid Values: X'00000000' to X'FFFFFFFF'

Default Value: X'00000000'

dst-node
Specifies the value to be compared against the destination node number (after being ANDed with the dst-mask). If you want to match on a single node, set the dst-node parameter to the node number and set the dst-mask to X'FFFFFFFFFFFF'. If you want to match on all nodes, set the dst-node parameter and the dst-mask parameter to X'000000000000'.

Valid Values: X'000000000000' to X'FFFFFFFFFFFF'

Default Value: X'000000000000'

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 X'FFFFFFFFFFFF'. If you want to match on all addresses, set the dst-address parameter and the dst-mask parameter to X'000000000000'.

Valid Values: X'000000000000' to X'FFFFFFFFFFFF'

Default Value: X'000000000000'

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 X'0000' and the dst-sck-range-end to X'FFFF'.

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

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 X'0000' and the dst-sck-range-end to X'FFFF'.

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

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 X'00000001' and the src-net-range-end to X'FFFFFFFE'.

Valid Values: X'00000000' to X'FFFFFFFE'

Default Value: X'00000000'

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 X'00000001' and the src-net-range-end to X'FFFFFFFE'.

Valid Values: X'00000000' to X'FFFFFFFE'

Default Value: X'00000000'

src-node
Specifies the value to be compared against the source node number (after being ANDed with the src-mask). If you want to match on a single node, set the src-node parameter to the node number and set the src-mask to X'FFFFFFFFFFFF'. If you want to match on all nodes, set the src-node parameter and the src-mask parameter to X'000000000000'.

Valid Values: X'00000000' to X'FFFFFFFF'

Default Value: X'00000000'

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 X'FFFFFFFFFFFF'. If you want to match on all addresses, set the src-address parameter and the src-mask parameter to X'000000000000'.

Valid Values: X'000000000000' to X'FFFFFFFFFFFF'

Default Value: X'000000000000'

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 X'0000' and the src-sck-range-end to X'FFFF'.

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#*

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.

Example: `delete 4`

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`

Accessing the IPX Monitoring Environment

For information on how to access the IPX monitoring environment, refer to “Getting Started (Introduction to the User circuit)” in the *Software User’s Guide*

IPX Monitoring Commands

Table 44 lists the IPX monitoring commands. The IPX monitoring commands allow you to view the parameters and statistics of the circuits and networks that transmit IPX packets. Monitoring 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 monitoring commands at the `IPX>` prompt. Table 44 summarizes the IPX monitoring commands.

Table 44. IPX Monitoring Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page xxxii.
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.
Counters	Displays the number of routing errors and packet overflows.
Delete keepalive connection	Deletes a Keepalive filtering table entry.
Disable	Disables IPX globally or on specific IPX circuits.
Dump routing tables	Displays the contents of the routing table.
Enable	Enables IPX globally or on specific IPX circuits.
Filters	Displays whether global SAP filtering is enabled, the SAP filter statements, and a count of the SAP advertisements which have been filtered.
Filter-Lists	Accesses the IPX circuit filter console. This is where the RIP router, RIP SAP, and IPX circuit-based filters can be monitored.
IPXWAN	Lists IPXWAN information for IPXWAN point-to-point circuits.
Keepalive	Displays the status of each active client/server connection in the keepalive-filtering table.
List	Lists the current configuration or the IPX address of each enabled circuit.

Table 44. IPX Monitoring Command Summary (continued)

Command	Function
Ping	Sends IPXPING packets to another host and watches for a response. This command can be used to isolate trouble in an internetwork environment.
Recordroute	Sends IPXPING record route packets to another host and watches for a response. Use this command to record and display the round-trip route between this device and another host. Use this information to isolate trouble in an internetwork environment.
Reset	Resets specific IPX circuits, global SAP filters, global IPX filters (access controls), static routes, static services, or the router, RIP, SAP, or IPX circuit-based filters (filter lists).
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.
Traceroute	Sends IPXPING trace route packets to another host and watches for a response. Use this command to trace and display each hop a packet takes on its way from this device to a destination host. Use this information to isolate trouble in an internetwork environment.
Exit	Returns you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

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`

```
IPX Access Controls:  Enabled
#   T Dest Net Host         Sock Sock Src Net  Host         Sock  Sock Count
1   E 2        000000000000 0    FFFF 3        000000000000 0     FFFF 0
2   I 0        000000000000 452  453  0        000000000000 0     FFFF 0
```

Access control index number

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.

Dest-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          Circ  Ifc
     420               1                  412/000004200000  3     2
     412               1                  412/000000000000  3     2
     412/000004200000  1                  412/000004200000  3     2
```

The first entry shows that the remote network 420 can be reached over the serial circuit with IPX network number 412. The second entry is the IPX network 412. 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 1 packet to the IPX node number 0000 0420 0000 on net number 412.

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      circuit size exceeded

Destination errors
Count  Type
0      Unknown
0      Checksum error
0      Non-existent socket
0      Congestion

IPX input packet overflows
Circ   Ifc    Name        Count
1      0       Eth/0      0
2      1       PPP/0      0
3      2       PPP/1      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.

circuit 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 circuit 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 globally or on specific circuits.

Syntax:
disable circuit ...
ipx

circuit *ipx-circuit#*
Disables the IPX circuit specified by *ipx-circuit#*. IPX can be re-enabled using the **enable** command.

Example: `disable circuit 2`

ipx Disables IPX globally on all IPX circuits. IPX can be globally re-enabled using the **enable** command.

Example: `disable ipx`

Dump

Use the **dump** command to display the contents of the routing tables.

Syntax:

dump

Example: `dump`

```
Type Dest Net Hops Delay Age(M: S)   via Router        Circ  Ifc
Dir  412      0    6        0: 0      412/000004000000 3     2
Dir  400      0    1        0: 0      400/020000000400 1     0
Dir  411      0    3        0: 0      411/400000000400 2     1
Stat 1        3    2        0: 0      400/010101010101 1     0
RIP  420      1    7        0:30      412/000004200000 3     2
Stat 444      2    2        0: 0      400/400000000444 1     0
Stat FFFFFFFD 14   3000     0: 0      400/111111111111 1     0
```

Type

- 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.
- Stat - specifies that this is a static route.

Dest net
Specifies the destination network number.

Hops Specifies the number of 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 639.
- 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 circuit that transmits the packet.

Circ IPX circuit number

Ifc Network interface number

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 globally or on specific circuits.

Syntax:

enable	circuit ...
	ipx

circuit *ipx-circuit#*
Enables IPX on the circuit specified by *ipx-circuit#*. An IPX network number must have been configured for the circuit before IPX can be enabled.

Example: `enable circuit 2`

ipx Globally enables IPX on all enabled IPX circuits.

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 Filters:  Enabled
Count  Max Hops  Type  Service Name
0      5         4     FILESRV01
```

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-lists, rip-lists, sap-lists, and ipx-lists.

For information about the commands available from this prompt, see “IPX Circuit Filter Monitoring Commands” on page 705.

Syntax:

filter-lists	router-lists
	rip-lists
	sap-lists
	ipx-lists

Example: `filter-lists router-lists`

IPXWAN

Use the **ipxwan** command to list the IPXWAN information for IPXWAN point-to-point circuits.

Syntax:

ipxwan detailed . . .
summary

detailed *ipx-circuit#*

Lists the IPXWAN information for the IPX circuit specified.

Example: **`ipxwan detailed 3`**

```
Detailed information for IPXWAN link over circuit 3 interface 2, PPP/1
This side is the IPXWAN slave
Neighbor Name: ipxwan-420
Neighbor Node ID: 420
Negotiated Routing Type: RIP/SAP
Link Delay: 6 1/18th sec ticks
Common Net#: 412
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
```

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 are RIP/SAP, unnumbered RIP, and Static Routing. When either unnumbered RIP or static routing is the negotiated routing type, no common network number is required on the link.

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. When the negotiated routing type is either unnumbered RIP or Static Routing, the value 0 will be displayed as the Common Net# for both the **IPXWAN detailed** and **IPXWAN summary** commands. 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 IPXWAN summary information for all IPXWAN point-to-point circuits.

Example: `ipxwan summary`

```
Circ   Ifc    Name        Common Net#  NodeID   Neighbor Name
3      2       PPP/1      412          420      ipxwan-420
```

Circ IPX circuit number

Ifc Network interface number

Common Net#
Network number agreed upon by both ends of the link. This number must be unique to the entire internetwork. The common net# will be 0 if the negotiated routing type is either unnumbered RIP or Static Routing.

NodeID
Node ID (also known as the internal network number) of the neighbor.

Neighbor Name
Router name of the neighbor as received in the RIP/SAP Information Request Packet.

Keepalive

Shows the status of each active client/server connection in the keepalive-filtering table.

Syntax:

keepalive

Example:

```
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)
```

List

Use the **list** command to list the current configuration or the IPX address of each enabled IPX circuit.

Syntax:

list *addresses*

configuration

addresses
Lists the IPX address of each enabled IPX circuit.

Example:

```
Circ   Ifc    Name        Type                            Network/Address
1      0       Eth/0      Ethernet                        400/020000000400
2      1       PPP/0      SCC Serial Line                 411/400000000400
3      2       FR/0       FR PVC                          412/000004000000
        Frame Relay PVC circuit number: 16
4      3       FR/0       FR SVC                          413/000004000000
        Frame Relay SVC circuit name: Indianapolis
```

Configuration
List the current IPX configuration. This command displays the same information as the **list summary** configuration command. See “List” on page 667 for an example of the display and an explanation of the output.

Ping

Use the **ping** command to make the router send IPXPING packets to a given destination (“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 monitoring. 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.

Notes:

1. Care should be taken when specifying the broadcast address (FFFFFFFFFFFF), as this could generate a large number of IPXPING response packets, which would degrade network and routing software performance.
2. If you enter the **ping** command without any parameters, you will be prompted for all parameters. If you enter only ***destination network*** and ***destination node***, default values will be used for the remaining parameters.

Syntax:

ping *dest-net dest-node src-net src-node size rate*

dest-net
Specifies the destination IPX network number. This parameter is required.

Valid Values: X'1' to X'FFFFFFFD'

Default Value: 1

dest-node
Specifies the destination IPX node address. This parameter is required.

Valid Value: X'1' to X'FFFFFFFFFFFF'

Default Value: None

src-net
Specifies the source IPX network number. This is an optional parameter. The value must be a known network number that is associated with a direct attached IPX circuit. If a source network is not specified, the network number of the IPX circuit on which the IPXPING request packets are sent will be used as the source IPX node. If the IPX circuit is an IPXWAN unnumbered RIP or static routing circuit, the node address of the IPX circuit used for the source network number will be used as the source node.

Valid Value: X'1' - X'FFFFFFFD'

Default Value: 1

src-node
Specifies the source IPX node address. This is an optional parameter. The value must be a known node address that is associated with a direct attached IPX circuit. If a source node is not specified, the node address of the IPX circuit on which the IPXPING request packets are sent will be used as the source IPX node. If the IPX circuit is an IPXWAN unnumbered RIP or static routing circuit,, the node address of the IPX circuit used for the source network number will be used as the source node.

Valid Value: X'1' - X'FFFFFFFFFFFE'

Default Value: None

size
Specifies the number of data bytes to be appended to the ping request. This is an optional parameter. The data includes the time the request is first sent so the amount specified cannot be smaller than 4 bytes. It also cannot be larger than the maximum packet size supported by the router or the output circuit. This value can vary depending on the configuration.

Valid Value: 4 to Router Maximum

Default Value: 56 bytes

rate
Specifies the number of seconds between ping requests. This is an optional parameter.

Valid Value: 1 to 60

Default Value: 1

Example: ping

```
Destination network number [1]? 20
Destination node number []? 00000001c200
Source network number [1]? 10
Source node number []? 000000019a00
Data size: [56]?
Rate in seconds [1]?

IPXPING 20/00000001C200: 56 data bytes
56 data bytes from 20/00000001C200: hops=3 time=0 ms
56 data bytes from 20/00000001C200: hops=3 time=40 ms
56 data bytes from 20/00000001C200: hops=3 time=0 ms

----20/00000001C200 IPXPING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms) min/ave/max = 0/13/40
```

RecordRoute

Use the **recordroute** command to report every forwarding circuit on the path to the destination and back again. If recordroute is invoked with no parameters, you will be prompted for all of them. Only the destination IPX network number and destination IPX node address are required.

There are two events that will end a recordroute. The first is when you press a key. The second is when the maximum number of recordroute request packets have been sent.

Syntax:

recordroute *dest-net dest-node src-net src-node rate number*

dest-net
Specifies the destination IPX network number. This parameter is required.

Valid Values: X'1' to X'FFFFFFFD'

Default Value: 1

dest-node
Specifies the destination IPX node address. This parameter is required.

Valid Values: X'1' to X'FFFFFFFFFFFE'

Default Value: None

src-net
Specifies the source IPX network number. This is an optional parameter. The value must be a known network number that is associated with a direct attached IPX circuit. If a source network is not specified, the network number of the IPX circuit on which the recordroute packets are sent will be used as the source IPX address. If the IPX circuit is an IPXWAN unnumbered RIP or static routing circuit, the network number of some other numbered IPX circuit will be used as the source address, since IPXWAN unnumbered RIP and static routing circuits are not assigned an IPX network number.

Valid Values: X'1' to X'FFFFFFFD'

Default Value: 1

src-node
Specifies the source IPX node address. This is an optional parameter. The value must be a known node address that is associated with a direct attached

IPX circuit. If a source node is not specified, the node address of the IPX circuit on which the recordroute packets are sent will be used as the source IPX node. If the IPX circuit is an IPXWAN unnumbered RIP or static routing circuit, the node address of IPX circuit used for the source network number will be used as the source node.

Valid Values: X'1' to X'FFFFFFFFFFFE'

Default Value: None

rate

Specifies the number of seconds between recordroute requests. This is an optional parameter.

Valid Values: 1 to 60

Default Value: 1

number

Specifies the maximum number of recordroute requests to be sent. This is an optional parameter. A value of zero will cause the recordroute to continue until a key is pressed.

Valid Values: 0 to 60

Default Value: 0

Example: `recordroute`

```
Destination network number [1]? 20
Destination node number []? 00000001c200
Source network number [1]? 10
Source node number []? 000000019a00
Rate in seconds [1]?
Number of packets to send [0]?

RECORDROUTE 20/00000001C200: 784 data bytes
784 data bytes from 20/00000001C200: seq_no=0 time=0 ms
Recorded Routes (in hex):
                10/000000019A00
                500/0000100A0000
                500/0000100C0000
                10/000000019000
                10/000000019A00 (Final Destination)

784 data bytes from 20/00000001C200: seq_no=1 time=30 ms  (same route)
784 data bytes from 20/00000001C200: seq_no=2 time=10 ms  (same route)
...
784 data bytes from 20/00000001C200: seq_no=18 time=0 ms
Recorded Routes (in hex):
                10/000000019A00
                0/0000100A0000
                20/00000001AE00
                20/00000001C200
                0/0000100B0000
                10/000000019000
                10/000000019A00 (Final Destination)

784 data bytes from 20/00000001C200: seq_no=19 time=0 ms  (same route)
784 data bytes from 20/00000001C200: seq_no=20 time=70 ms  (same route)
784 data bytes from 20/00000001C200: seq_no=21 time=0 ms  (same route)
...
784 data bytes from 20/00000001C200: seq_no=48 time=0 ms
Recorded Routes (in hex):
                10/000000019A00
                500/0000100A0000
                500/0000100C0000
                10/000000019000
                10/000000019A00 (Final Destination)
```

```
784 data bytes from 20/00000001C200: seq_no=49 time=0 ms  (same route)
784 data bytes from 20/00000001C200: seq_no=50 time=0 ms  (same route)

----20/00000001C200 RECORDROUTE Statistics----
53 packets transmitted, 38 packets received, 28% packet loss
5 unreachables, 0 no usable source addresses, 0 buffer unavailable
round-trip (ms) min/ave/max = 0/23/100
```

The entire path is reported only once on the first response or when the path changed. In the above example, the path changed twice.

Reset

Use the **reset** command to reset specific IPX circuits, global SAP filters, global IPX filters (access controls), static routes, static services, or the Router, RIP, SAP, or IPX circuit-based filters (filter lists).

Syntax:

reset	access-controls
	circuit . . .
	filters
	filter-lists
	route-static
	sap-static

access-controls
Resets the global IPX filters (access-controls) based on the configuration parameter stored in the configuration memory. Changes made to the global IPX filter configuration will be activated.

Example: reset access-controls

circuit *ipx-circuit#*
Resets IPX on the specified IPX circuit using configuration parameter values stored in the configuration memory. Changes made to the IPX configuration on the IPX circuit will be activated.

Example: reset circuit 2

filters
Resets the global SAP filters based on the configuration parameter values stored in the configuration memory. Changes made to the global SAP filter configuration will be activated.

Example: reset filters

filter-lists *filter-type*
Resets the circuit-based filter based on configuration parameter values stored in the configuration memory. Changes made to the circuit-based filter configuration will be activated. Valid ***filter-types*** are router,rip,sap, and ipx.

Example: reset filter-lists rip

route-static
Resets the static routes based on the configuration parameter values stored in the configuration memory. Changes made to the static route configuration will be activated.

Example: reset route-static

sap-static
Resets the static services based on the configuration parameter values stored in the configuration memory. Changes made to the static services configuration will be activated.

Example: reset sap static

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:

sizes

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`

```
9 entries used out of 32

State Typ Service Name                  Hops   Age     Net  /   Host     /Sock
SAP  4    PCS12                          3     0:50        1/000000000048/0451
SAP  4    ACMPCS                         3     0:50        1/00000000004A/0451
SAP  4    DEVEL2                         1     0:50       11/0000000000B4/0451
SAP  4    PLANNING                       2     0:50       BB/0000000000B7/0451
SAP  4    DEVEL                          2     0:50       BB/0000000000EE/0451
SAP  4    SOFT2                          1     0:30      704/000000000094/0451
SAP  4    SKYSURF1                       2     0: 5 2C39ABE9/000000000001/0451
SAP  278  DIRTREE                        2     0: 5 2C29ABE9/000000000001/4005
Stat 26B  DIRTREE                        2     0: 0      444/000000000001/0045
```

State Specifies one of the following parameters:
SAP - indicates that this service was obtained by the SAP routing protocol.
Del - 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.
Stat - indicates that this service is a static service.

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 Del 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 circuit 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.

Traceroute

Use the **traceroute** command to report each hop a ping request takes on its way to a final destination. If traceroute is invoked with no parameters, you will be prompted for all of them. Only the destination IPX network number and destination IPX node address are required.

There are three events that will end a traceroute. The first is when you press a key. The second is when a response is received from the destination address. The third is when the maximum number of hops has been reached.

Syntax:

traceroute *dest-net dest-node src-net src-node size probes rate hops*

dest-net
Specifies the destination IPX network number. This parameter is required.

Valid Values: X'1' to X'FFFFFFFD'

Default Value: 1

dest-node
Specifies the destination IPX node address. This parameter is required.

Valid Values: X'1' to X'FFFFFFFFFFFE'

Default Value: None

src-net
Specifies the source IPX network number. This is an optional parameter. The value must be a known network number that is associated with a direct attached IPX circuit. If a source network is not specified, the network number of the IPX circuit on which the traceroute packets are sent will be used as the source IPX address. If the IPX circuit is an IPXWAN unnumbered RIP or static routing circuit, the network number of some other numbered IPX circuit will be used as the source address, since IPXWAN unnumbered RIP and static routing circuits are not assigned an IPX network number.

Valid Value: X'1' to X'FFFFFFFD'

Default Value: 1

src-node
Specifies the source IPX node address. This is an optional parameter. The value must be a known node address that is associated with a direct attached IPX circuit. If a source node is not specified, the node address of the IPX circuit on which the traceroute packets are sent will be used as the source IPX node. If the IPX circuit is an IPXWAN unnumbered RIP or static routing circuit, the node address of IPX circuit used for the source network number will be used as the source node.

Valid Values:X'1' to X'FFFFFFFFFFFE'

Default Value: None

size
Specifies the number of data bytes to be appended to the traceroute request. This is an optional parameter. The data includes the time the request is first sent, so the number specified cannot be smaller than 4 bytes. It also cannot be larger than the maximum packet size of the router or the output circuit. This value can vary depending on the configuration.

Valid Values: 4 to router maximum

Default Value: 56

probes
Specifies how many traceroute requests to send per hop. This is an optional parameter.

Valid Values: 1 to 10

Default Value: 3

rate
Specifies the number of seconds to wait between probes, when there is not an answer to the traceroute request. This is an optional parameter.

Valid Values: 1 to 60

Default Value: 1

hops
Specifies the maximum number of hops to send traceroute requests. This is an optional parameter. Without NLSP, a packet can traverse a maximum of 16 nodes (hence the default of 16). With NLSP or the IBM 6611 half-router solution, the limit is no longer 16.

Valid Values: 1 to 255

Default Value: 16

Example: `traceroute`

```
Destination network number [1]? 20
Destination node number []? 00000001c200
Source network number [1]? 10
Source node number []? 000000019a00
```

```
Data size:  [56]?
Number of probes per hop [3]?
Wait time between retries in seconds [1]?
Maximum Hops [16]?

TRACEROUTE 20/00000001C200: 56 data bytes
1 10/000000019000: 0 ms * 500/0000100B0000 20 ms
2 * * *
3 20/00000001C200: 10 ms 60 ms 20 ms
```

The source IPX address of a traceroute response is reported only once as long as it does not change. In the above example, two different routers responded to the one hop traceroute request. This would happen if the route to the destination changed between probes.

There is other information reported by traceroute besides the round trip time of a probe:

- '*' - No response packet was received in the time specified.
- 'H!' - The destination network is unreachable. This would be reported if the route to the destination was lost after traceroute was started.
- 'BF' - No buffers available.

IPX Circuit Filter Monitoring Commands

Table 45 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-lists, rip-lists, sap-lists, and ipx-lists.

Table 45. IPX circuit Filter Command Summary

Command	Function
Cache	Displays the contents of the filter cache for the specified circuit. 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 you to the previous command level. See “Exiting a Lower Level Environment” on page xxxii.

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    Circ  Ifc   Direction  State      Default    Cache
----- ----- ----- ---------- ---------- ---------- -----
1     1     0     INPUT      ENABLED    INCLUDE    10
2     1     0     OUTPUT     ENABLED    INCLUDE    10
3     2     1     INPUT      DISABLED   INCLUDE    10
4     2     1     OUTPUT     DISABLED   INCLUDE    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    Circ  Ifc   Direction  State      Default    Cache
----- ----- ----- ---------- ---------- ---------- -----
1     1     0     INPUT      ENABLED    INCLUDE    10

Filter Lists:
Name                           Action     Count
------------------------------ ---------- ----------
ipx01                          EXCLUDE    43
ipx02                          INCLUDE    23453
```

IPX Dynamic Reconfiguration Support

This section describes dynamic reconfiguration (DR) as it affects Talk 6 and Talk 5 commands.

CONFIG (Talk 6) Delete Interface

Internet Packet Exchange (IPX) supports the CONFIG (Talk 6) **delete interface** command with the following consideration:

When the network interface is deleted, all IPX circuits, static routes, static services and circuit-filters (RIP router, RIP, SAP and IPX) configured on the network interface are deleted from the configuration memory.

GWCON (Talk 5) Activate Interface

IPX supports the GWCON (Talk 5) **activate interface** command with the following considerations:

- IPX must be globally enabled currently before IPX can be activated on a new network interface.
- IPX must be enabled currently on at least one network interface before IPX can be activated on a new network interface.

- The GWCON, protocol IPX, reset route-static command must be used to activate static routes on the network interface.
- The GWCON, protocol IPX, reset sap-static command must be used to activate static services on the network interface.
- The GWCON, protocol IPX, reset filter-lists command must be used to activate circuit filters on the network interface.

The following table summarizes the IPX configuration changes that are activated when the GWCON (Talk 5) **activate interface** command is invoked:

Commands performed as part of the GWCON (Talk 5) activate interface command
GWCON, protocol IPX, reset circuit **Note:** When the network interface is activated, all IPX circuits configured on the network interface are activated.

GWCON (Talk 5) Reset Interface

IPX supports the GWCON (Talk 5) **reset interface** command with the following considerations:

- The GWCON, protocol IPX, reset route-static command must be used to reset static routes on the network interface.
- The GWCON, protocol IPX, reset sap-static command must be used to reset static services on the network interface.
- The GWCON, protocol IPX, reset filter-lists command must be used to reset circuit filters on the network interface.

The following table summarizes the IPX configuration changes that are activated when the GWCON (Talk 5) **reset interface** command is invoked:

Commands performed as part of the GWCON (Talk 5) reset interface command
GWCON, protocol IPX, reset circuit **Note:** When the network interface is reset, all IPX circuits configured on the network interface are reset.

GWCON (Talk 5) Component Reset Commands

IPX supports the following IPX-specific GWCON (Talk 5) **reset** commands:

GWCON, Protocol IPX, Reset Circuit Command

Description:

The specified IPX circuit is reset.

Network Effect:

The IPX circuit is deleted and recreated using the IPX circuit parameters in the configuration memory. All routes and services learned through the IPX circuit are advertised as down, are deleted, and then re-learned when the IPX circuit is recreated. If the IPX network number on the circuit has changed, sessions involving the clients on that circuit will be lost and the clients may have to be reset.

Limitations:

- The **GWCON, protocol IPX, reset route-static** command must be used to reset static routes on the reset circuit.
- The **GWCON, protocol IPX, reset sap-static** command must be used to reset static services on the reset circuit.

- The **GWCON, protocol IPX, reset filter-lists** command must be used to reset circuit filters on the reset circuit.

The following table summarizes the IPX configuration changes that are activated when the **GWCON, protocol IPX, reset circuit** command is invoked:

Commands whose changes are activated by the GWCON, protocol IPX, reset circuit command
CONFIG, protocol IPX, add broadcast-circuit
CONFIG, protocol IPX, add ipxwan-circuit
CONFIG, protocol IPX, delete circuit
CONFIG, protocol IPX, disable circuit
CONFIG, protocol IPX, disable keepalive-filtering
CONFIG, protocol IPX, disable netbios-broadcast
CONFIG, protocol IPX, disable reply-to-get-nearest serve
CONFIG, protocol IPX, disable rip
CONFIG, protocol IPX, disable rip-sap-pacing
CONFIG, protocol IPX, disable sap
CONFIG, protocol IPX, enable circuit
CONFIG, protocol IPX, enable keepalive-filtering
CONFIG, protocol IPX, enable netbios-broadcast
CONFIG, protocol IPX, enable reply-to-get-nearest serve
CONFIG, protocol IPX, enable rip
CONFIG, protocol IPX, enable rip-sap-pacing
CONFIG, protocol IPX, enable sap
CONFIG, protocol IPX, frame
CONFIG, protocol IPX, move circuit
CONFIG, protocol IPX, set ipxwan
CONFIG, protocol IPX, set net-number
CONFIG, protocol IPX, set rip-ticks
CONFIG, protocol IPX, set rip-update-interval
CONFIG, protocol IPX, set sap-update-interval
CONFIG, protocol IPX, set split-horizon

GWCON, Protocol IPX, Reset Route-Static Command

Description:

All IPX static routes, including the default route, and whether IPX static routing is globally enabled or disabled, are reset.

Network Effect:

No network disruption.

Limitations:

None.

The following table summarizes the IPX configuration changes that are activated when the **GWCON, protocol IPX, reset route-static** command is invoked:

Commands whose changes are activated by the GWCON, protocol IPX, reset route-static command
CONFIG, protocol IPX, add route-static
CONFIG, protocol IPX, delete route-static
CONFIG, protocol IPX, disable route-static
CONFIG, protocol IPX, enable route-static

GWCON, Protocol IPX, Reset Sap-Static Command

Description:
All IPX static services, including whether the use of IPX static services is globally enabled or disabled, are reset.

Network Effect:
No network disruption.

Limitations:
None.

The following table summarizes the IPX configuration changes that are activated when the GWCON, protocol IPX, reset sap-static command is invoked:

Commands whose changes are activated by the GWCON, protocol IPX, reset sap-static command
CONFIG, protocol IPX, add sap-static
CONFIG, protocol IPX, delete sap-static
CONFIG, protocol IPX, disable sap-static
CONFIG, protocol IPX, enable sap-static

GWCON, Protocol IPX, Reset Filters Command

Description:
All global SAP filters, including whether the use of SAP filtering is globally enabled or disabled, are reset.

Network Effect:
No network disruption.

Limitations:
None.

The following table summarizes the IPX configuration changes that are activated when the **GWCON, protocol IPX, reset filters** command is invoked:

Commands whose changes are activated by the GWCON, protocol IPX, reset filters command
CONFIG, protocol IPX, add filter
CONFIG, protocol IPX, delete filter
CONFIG, protocol IPX, set filter

GWCON, Protocol IPX, Reset Access-Controls Command

Description:
All global IPX filters (access-controls), including whether the use of IPX filtering is globally enabled or disabled, are reset.

Network Effect:
No network disruption.

Limitations:
None.

The following table summarizes the IPX configuration changes that are activated when the **GWCON, protocol IPX, reset access-controls** command is invoked:

Commands whose changes are activated by the GWCON, protocol IPX, reset access-controls command
CONFIG, protocol IPX, add access-control
CONFIG, protocol IPX, delete access-control
CONFIG, protocol IPX, move access-control
CONFIG, protocol IPX, set access-control

GWCON, Protocol IPX, Reset Filter-Lists Router Command

Description:
All RIP router circuit filters, filter-lists and items, including whether the use of RIP router circuit filters is globally enabled or disabled, are reset.

Network Effect:
No network disruption.

Limitations:
None.

The following table summarizes the IPX configuration changes that are activated when the **GWCON, protocol IPX, reset filter-lists router** command is invoked:

Commands whose changes are activated by the GWCON, protocol IPX, reset filter-lists router command
CONFIG, protocol IPX, filter-lists router, attach
CONFIG, protocol IPX, filter-lists router, create
CONFIG, protocol IPX, filter-lists router, default
CONFIG, protocol IPX, filter-lists router, delete
CONFIG, protocol IPX, filter-lists router, detach
CONFIG, protocol IPX, filter-lists router, disable
CONFIG, protocol IPX, filter-lists router, enable
CONFIG, protocol IPX, filter-lists router, move
CONFIG, protocol IPX, filter-lists router, update, add
CONFIG, protocol IPX, filter-lists router, update, delete
CONFIG, protocol IPX, filter-lists router, update, move
CONFIG, protocol IPX, filter-lists router, update, set-action

GWCON, Protocol IPX, Reset Filter-Lists RIP Command

Description:
All RIP circuit filters, filter-lists and items, including whether the use of RIP circuit filters is globally enabled or disabled, are reset.

Network Effect:
No network disruption.

Limitations:
None.

The following table summarizes the IPX configuration changes that are activated when the **GWCON, protocol IPX, reset filter-lists rip** command is invoked:

Commands whose changes are activated by the GWCON, protocol IPX, reset filter-lists rip command
CONFIG, protocol IPX, filter-lists rip, attach
CONFIG, protocol IPX, filter-lists rip, create
CONFIG, protocol IPX, filter-lists rip, default
CONFIG, protocol IPX, filter-lists rip, delete
CONFIG, protocol IPX, filter-lists rip, detach
CONFIG, protocol IPX, filter-lists rip, disable
CONFIG, protocol IPX, filter-lists rip, enable
CONFIG, protocol IPX, filter-lists rip, move
CONFIG, protocol IPX, filter-lists rip, update, add
CONFIG, protocol IPX, filter-lists rip, update, delete
CONFIG, protocol IPX, filter-lists rip, update, move
CONFIG, protocol IPX, filter-lists rip, update, set-action

GWCON, Protocol IPX, Reset Filter-Lists SAP Command

Description:

All SAP circuit filters, filter-lists and items, including whether the use of SAP circuit filters is globally enabled or disabled, are reset.

Network Effect:

No network disruption.

Limitations:

None.

The following table summarizes the IPX configuration changes that are activated when the **GWCON, protocol IPX, reset filter-lists sap** command is invoked:

Commands whose changes are activated by the GWCON, protocol IPX, reset filter-lists sap command
CONFIG, protocol IPX, filter-lists sap, attach
CONFIG, protocol IPX, filter-lists sap, create
CONFIG, protocol IPX, filter-lists sap, default
CONFIG, protocol IPX, filter-lists sap, delete
CONFIG, protocol IPX, filter-lists sap, detach
CONFIG, protocol IPX, filter-lists sap, disable
CONFIG, protocol IPX, filter-lists sap, enable
CONFIG, protocol IPX, filter-lists sap, move
CONFIG, protocol IPX, filter-lists sap, update, add
CONFIG, protocol IPX, filter-lists sap, update, delete
CONFIG, protocol IPX, filter-lists sap, update, move
CONFIG, protocol IPX, filter-lists sap, update, set-action

GWCON, Protocol IPX, Reset Filter-Lists IPX Command

Description:
All IPX circuit filters, filter-lists and items, including whether the use of IPX circuit filters is globally enabled or disabled, are reset.

Network Effect:
No network disruption.

Limitations:
None.

The following table summarizes the IPX configuration changes that are activated when the **GWCON, protocol IPX, reset filter-lists ipx** command is invoked:

Commands whose changes are activated by the GWCON, protocol IPX, reset filter-lists ipx command
CONFIG, protocol IPX, filter-lists ipx, attach
CONFIG, protocol IPX, filter-lists ipx, create
CONFIG, protocol IPX, filter-lists ipx, default
CONFIG, protocol IPX, filter-lists ipx, delete
CONFIG, protocol IPX, filter-lists ipx, detach
CONFIG, protocol IPX, filter-lists ipx, disable
CONFIG, protocol IPX, filter-lists ipx, enable
CONFIG, protocol IPX, filter-lists ipx, move
CONFIG, protocol IPX, filter-lists ipx, set-cache
CONFIG, protocol IPX, filter-lists ipx, update, add
CONFIG, protocol IPX, filter-lists ipx, update, delete
CONFIG, protocol IPX, filter-lists ipx, update, move
CONFIG, protocol IPX, filter-lists ipx, update, set-action

GWCON (Talk 5) Temporary Change Commands

IPX supports the following GWCON commands that temporarily change the operational state of the device. These changes are lost whenever the device is reloaded, restarted, or you execute any dynamically reconfigurable command.

Commands
GWCON, protocol IPX, disable circuit
GWCON, protocol IPX, disable ipx
GWCON, protocol IPX, enable circuit
GWCON, protocol IPX, enable ipx

Non-Dynamically Reconfigurable Commands

The following table describes the IPX configuration commands that cannot be dynamically changed. To activate these commands, you need to reload or restart the device.

Commands
CONFIG, protocol IPX, set keepalive-table-size
CONFIG, protocol IPX, set local-cache size

IPX circuit Filter Monitoring Commands (Talk 5)

CONFIG, protocol IPX, set maximum networks
CONFIG, protocol IPX, set maximum routes-per-destination
CONFIG, protocol IPX, set maximum services
CONFIG, protocol IPX, set maximum total-route-entries
CONFIG, protocol IPX, set name
CONFIG, protocol IPX, set node-id
CONFIG, protocol IPX, set remote-cache size

Part 3. Appendixes

Appendix A. Interoperating with the IBM 6611 Router

A number of configuration considerations must be addressed for IBM Nways Multiprotocol Routing 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 Routing 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

The following are bridge configuration considerations:

- The LAN identification (Segment number) of the DLSw must match on both the IBM 2210 and IBM 6611 routers. If a mismatch persistently exists, enter the IBM Nways Multiprotocol Routing Services Configurator (Talk 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 2210 and IBM 6611 routers.

DLSw-Related Considerations

DLSw-related interoperability considerations are as follows:

- The IBM Nways Multiprotocol Routing 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 Routing Services DLSw implementation upon receipt.
- The IBM 6611 DLSw implementation processes SSP_ENTER_BUSY/EXIT_BUSY messages received from the IBM Nways Multiprotocol Routing Services DLSw implementation but will not generate similar flow control related SSP messages.
- The IBM Nways Multiprotocol Routing 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 Routing 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 Routing 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 Routing Services DLSw group feature has implications for the selection of RIP/OSPF:
  - To utilize DLSw groups on an 2210, you must also configure OSPF/MOSPF. But because these DLSw groups are not interoperable with the 6611, it is possible to configure the 2210 with only RIP enabled and no OSPF configuration.
  - Although OSPF and RIP can both be enabled on the IBM 2210 side, MOSPF (if selected through the OSPF configuration) is not supported by the IBM 6611.
- Within the IBM Nways Multiprotocol Routing 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 Routing 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 priority assigned by the IBM 2210 hardware for BRS can be implemented in an outbound direction, the priority 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 Routing 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 Routing 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 Routing 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 Routing 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 2210'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 Routing 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 B. Interoperating with the IBM 6611 Bridge

A number of configuration issues must be addressed before implementing bridging on the IBM 2210 to interoperate with bridging on the IBM 6611.

This appendix provides an overview of these issues, and indicates which features of IBM 2210's bridge implementation are ***not*** interoperable with the IBM 6611's bridge implementation.

To help avoid building an incompatible network, the following bridge configuration issues should be considered when using the IBM 6611 and the IBM 2210 as the two end-bridges over PPP and frame-relay serial links.

For PPP, the IBM 2210 bridge supports different MAC types (Ethernet and token-ring) as described in RFC 1638, *PPP Bridging Control Protocol*. For frame-relay, the IBM 2210 supports RFC 1490/2427 *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 2210 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 2210 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.

Other PPP Considerations

You should keep the following in mind when configuring a 2210 and a 6611 in a bridged network:

- To bridge traffic over a PPP link, the negotiated maximum receive unit (MRU) must be large enough to contain a bridged frame. The bridged frame contains the data and the MAC layer header from the original LAN.

  For example, an Ethernet frame can contain 1500 bytes of data. When bridged over a WAN link, an additional 14 bytes of Ethernet MAC header are included in the bridged traffic bringing the packet size to 1514. This means that the negotiated PPP MRU must be at least 1514 to bridge the frame.

  You should consider an MRU size that will be more than large enough to hold any bridged frame. Try using 2000 or 2048 as a starting value for the MRU.
- Make sure that both ends of the PPP link are configured for the same size MRU. If you use the default MRU for the 2210, make sure the MRU for the 6611 matches the 2210 MRU value.

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 2210 (SR-TB) - PPP (TB) - IBM 6611 (TB) - Ethernet

Alt: Token Ring (SR) - IBM 2210 (SRB) - PPP (SR) - IBM 6611 (SR-TB) - Ethernet

Example 2: Token Ring (TB) - IBM 2210 (TB) - PPP (TB) - IBM 6611 (TB) - ETH/TKR

Alt: Token Ring (SR) - IBM 2210 (SRB) - PPP (SR) - IBM 6611 (SR-TB) - ETH

Alt: Token Ring (SR) - IBM 2210 (SRB) - PPP (SR) - IBM 6611 (SRB) - TKR

Alt: Token Ring (TB) - IBM 2210 (SR-TB) - PPP (SR) - IBM 6611 (SRB) - TKR

Alt: Token Ring (TB) - IBM 2210 (SR-TB) - PPP (SR) - 6611 (SR-TB) - ETH

The LAN frames generated by Boundary Access Node (BAN) and DLSw are source-routed token-ring frames. Based on the media type and the bridge configuration behavior of the associated outgoing bridge port, the IBM 2210 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 acknowledgment

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 ATM real interface

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

AVI ATM virtual interface

ayt are you there

BAN Boundary Access Node

BBCM Bridging Broadcast Manager

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

CIP Classical IP

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

DCS Directly connected server

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

E&M Ear & Mouth

E1 2.048 Mbps transmission rate

EDEL end delimiter

EDI error detected indicator

EGP Exterior Gateway Protocol

EIA Electronics Industries Association

ELAN Emulated LAN

ELAP EtherTalk Link Access Protocol

ELS Event Logging System

ESI End system identifier

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

FXO Foreign Exchange Office

FXS Foreign Exchange Station

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

LAC L2TP Network Access Concentrator

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

LIS Logical IP subnet

LLC logical link control

LLC2 logical link control 2

LMI local management interface

LNS L2TP Network Server

LRM LAN reporting mechanism

LS link state

LSA link state advertisement

LSB least significant bit

LSI LAN shortcuts interface

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

MOSDBG
Micro Operating System Debugging Tool

MOSPF
Open Shortest Path First with multicast extensions

MSB most significant bit

MSDU MAC service data unit

MRU maximum receive unit

MTU maximum transmission unit

nak not acknowledged

NBMA Non-Broadcast Multiple Access

NBP Name Binding Protocol

NBR neighbor

NCP Network Control Protocol

NCP Network Core Protocol

NetBIOS
Network Basic Input/Output System

NHRP Next Hop Resolution Protocol

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
nonvolatile configuration

OOS Out of Service

OPCON
Operator Console

OSI open systems interconnection

OSICP
OSI Control Protocol

OSPF Open Shortest Path First

OUI organization unique identifier

PC personal computer

PCR peak cell rate

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

SCR Sustained cell rate

SCSP Server Cache Synchronization Protocol

sdel start delimiter

SDLC SDLC relay, synchronous data link control

seqno sequence number

SGID sever group id

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

VCC Virtual Channel Connection

VINES VIrtual NEtworking System

VIR variable information rate

VL virtual link

VNI Virtual Network Interface

VoFR Voice over Frame Relay

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. (1) Operational. (2) Pertaining to a node or device that is connected or is available for connection to another node or device.

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.

agent. A system that assumes an agent role.

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. Asynchronous 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)

Attribute Value Pair (AVP). A uniform method of encoding message types and bodies. This method maximizes the extensibility while permitting interoperability of L2TP.

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)

CCITT. International Telegraph and Telephone Consultative Committee. This was an organization of the International Telecommunication Union (ITU). On 1 March 1993 the ITU was reorganized, and responsibilities for standardization were placed in a subordinate organization named the Telecommunication Standardization Sector of the Telecommunication Union (ITU-TS). "CCITT" continues to be used for recommendations that were approved before the reorganization.

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 database (CDB). A database that stores the configuration parameters of one or several devices. It is prepared and updated using the configuration program.

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*.

connection. In data communication, an association established between functional units for conveying information. (I) (A)

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)

dependent LU requester (DLUR). An APPN end node or an APPN network node that owns dependent LUs, but requests that a dependent LU server provide the SSCP services for those dependent LUs.

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.

EIA unit. A unit of measure, established by the Electronic Industries Association, equal to 44.45 millimeters (1.75 inches).

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

fax. Hardcopy received from a facsimile machine. Synonymous with ***telecopy.***

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.

flash memory. A data storage device that is programmable, erasable, and does not require continuous power. The chief advantage of flash memory over other programmable and erasable data storage devices is that it can be reprogrammed without being removed from the circuit board.

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.

front-end processor. A processor such as the IBM 3745 or 3174, that relieves a main frame from the communication control tasks.

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.

hub (intelligent). A wiring concentrator, such as the IBM 8260, that provides bridging and routing functions for LANs with different cables and protocols.

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 Digital Network Exchange (IDNX). A processor integrating voice, data, and image applications. It also manages the transmission resources, and connects to multiplexers and network management support systems. It allows integration of equipment from different vendors.

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).

interleaving. (1) The alternating of two or more operations or functions through the overlapped use of a computer facility. (2) In data transmission, the alternating of packets from one data stream with packets from another.

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.

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)*.

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.

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.

IPPN. The interface that other protocols can use to transport data over IP.

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

L2TP Access Concentrator (LAC). A device attached to one or more public service telephone network (PSTN) or ISDN lines capable of handling both PPP operation and of the L2TP protocol. The LAC implements the media over which L2TP operates. L2TP passes the traffic to one or more L2TP Network Servers (LNS). L2TP can tunnel any protocol carried by the PPP network.

L2TP Network Server (LNS). An LNS operates on any platform capable that can be a PPP end station. The LNS handles the server side of the L2TP protocol. Since L2TP relies only on the single media over which L2TP tunnels arrive, the LNS has only a single LAN or WAN interface, yet is still able to terminate calls arriving from any the full range of PPP interfaces supported by a LAC. These include asynchronous ISDN, synchronous ISDN, V.120, and other types of connections.

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. (1) MIB module. (2) Management Information Base.

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 Access Server (NAS). A device providing temporary, on-demand network access to users. This access is point-to-point using PSTN or ISDN lines.

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*.

Non-Return-to-Zero Changes-on-Ones Recording (NRZ-1). A recording method in which the ones are represented by a change in the condition of magnetization, and zeros are represented by the absence of change. Only the one signals are explicitly recorded. (Previously called ***non-return-to-zero inverted***, NRZI, recording.)

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 loss ratio. The probability that a packet will not reach its destination or not reach it within a specified time.

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.

private branch exchange (PBX). A private telephone exchange for transmission of calls to and from the public telephone network.

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)

pulse code modulation (PCM). A standard adopted for the digitalization of an analog voice signal. In PCM, the voice is sampled at a rate of eight kHz and each sample is coded in an 8-bit frame.

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.

real-time processing. The manipulation of data that are required, or generated, by some process while the process is in operation. Usually the results are used to influence the process, and perhaps related processes, while it is occurring.

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

SAP. See service access point.

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.

Serial Line Internet Protocol (SLIP). A protocol used over a point-to-point connection between two IP hosts over a serial line, for example, a serial cable or an RS232 connection into a modem, over a telephone line.

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. (3) In L2TP, L2TP creates a session when an end-to-end PPP connection is attempted between a dial user and the LNS; regardless of whether the user initiates the session or the LNS initiates an outbound call. The datagrams for th session are sent over the tunnel between the LAC and LNS. The LNS and LAC maintain the state information for each user attached to an LAC.

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. (1) An endpoint for communication between processes or application programs. (2) The abstraction provided by the University of California's Berkeley Software Distribution (commonly called Berkeley UNIX or BSD UNIX) that serves as an endpoint for communication between processes or applications.

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. In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions. (I) (A)

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

TCP/IP. (1) Transmission Control Protocol/Internet Protocol. (2) A UNIX-like/Ethernet-based system-interconnect protocol originally developed by the US Department of Defense. TCP/IP facilitated ARPANET (Advanced Research Projects Agency Network), a packet-switched research network for which Layer 4 was TCP and Layer 3, IP.

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) 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.

Tunnel. A tunnel is defined by an LNS-LAC pair. The tunnel carries PPP datagrams between the LAC and the LNS. A single tunnel can multiplex many sessions. A control connection operating over the same tunnel controls the establishment, release, and maintenance of all sessions and of the tunnel itself.

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.34. An ITU-T Recommendation for modem communication over standard commercially available voice-grade 33.6-Kbps (and slower) channels.

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.

version. A separately licensed program that usually has significant new code or new function.

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 connection. In frame relay, the return path of a potential connection.

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

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Readers' Comments — We'd Like to Hear from You

Nways Multiprotocol Routing Services
Protocol Configuration and Monitoring
Reference Volume 1
Version 3.4

Publication No. SC30-3680-10

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC30-3680-10

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

IBM Corporation
Design & Information Development
Department CGF/Bldg. 656
PO Box 12195
Research Triangle Park, NC
27709-9990

Fold and Tape | **Please do not staple** | Fold and Tape

SC30-3680-10

Cut or Fold
Along Line

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC30-3680-10

Spine information:

IBM

Nways Multiprotocol Routing Services

MRS V3.4 Protocol Reference V1