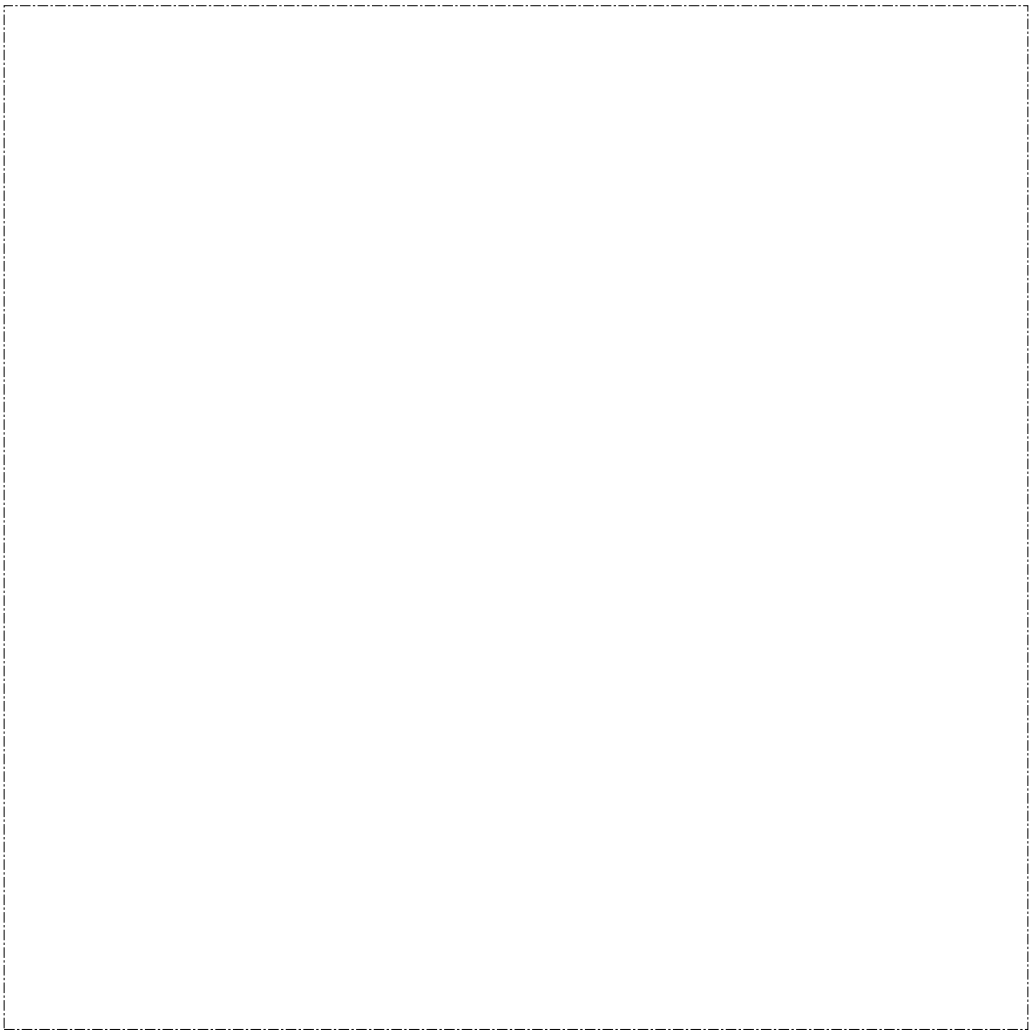
Multiprotocol Switched Services (MSS) Server

SC30-3819-01

Command Line Interface Volume 2
User's Guide and Protocol Reference



Multiprotocol Switched Services (MSS) Server

SC30-3819-01

Command Line Interface Volume 2
User's Guide and Protocol Reference

Note

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

Second Edition (January 1997)

This edition applies to Version 1 Release 1 of the IBM Multiprotocol Switched Services (MSS) Server and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

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

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

International Business Machines Corporation
Design and Information Development
Department CGF
P.O. Box 12195
Research Triangle Park, NC 27709-9990
U.S.A.

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

Contents

Part 2. Configuring and Monitoring Router Protocols

Figures

Tables

Notices

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

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

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

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

Trademarks

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

Advanced Peer-to-Peer Networking	CUA	Operating System/2
AIX	IBM	RISC System/6000
AIXwindows	Micro Channel	System/370
APPN	NetView	VTAM
BookManager	Nways	Web Explorer
Common User Access	OS/2	PS/2

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

Preface

This manual contains the information you will need to use the command interface for configuration and operation of the IBM 8210 Nways Multiprotocol Switched Services (MSS) Server or your IBM Nways Multiprotocol Switched Services (MSS) Server module, hereafter referred to as “the router,” installed on your IBM Multiprotocol Switched Services (MSS) Server. With the help of this manual, you should be able to perform the following processes and operations:

- Configure, monitor, and use the Multiprotocol Switched Services (MSS) Server base code on your IBM 8210 Nways Multiprotocol Switched Services (MSS) Server or your IBM Nways Multiprotocol Switched Services (MSS) Server module
- Configure, monitor, and use the interfaces and Link Layer software supported by your router.

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

Conventions Used in This Manual

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

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

   res**tart**

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

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

   t**ime host ...**

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

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

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

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

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

   Ctrl-P

   Ctrl P

MSS Server Library

The following hard copy publications are shipped with the product. The manuals in this list are also included in displayable softcopy form on the Multiprotocol Switched Services (MSS) Softcopy Library CD-ROM (SK2T-0378). This CD-ROM is shipped with initial orders for the MSS Server.

The reference cards, the International Program License Agreement, and the safety information booklet are shipped in hard copy only and are not included on the CD-ROM.

- *IBM 8210 Nways Multiprotocol Switched Services (MSS) Server Setup and Problem Determination Guide,* GA27-4140
- *IBM 8210 Nways Multiprotocol Switched Services (MSS) Server Operations Reference Card,* GX27-4017
- *IBM Multiprotocol Switched Services (MSS) Server Configuration and Operations Guide,* SC30-3821
- *CAUTION: Safety Information - Read This First,* SD21-0030
- *International Program License Agreement*
- *IBM Nways Multiprotocol Switched Services (MSS) Server Module Reference Card,* GX27-4018
- *IBM Nways Multiprotocol Switched Services (MSS) Server Module Setup and Problem Determination Guide,* GA27-4141

The following publications are not shipped in hard copy, but are offered in soft copy form on the Multiprotocol Switched Services (MSS) Softcopy Library CD-ROM (SK2T-0378). All of these manuals can be separately ordered in hard copy form through your IBM marketing representative.

- *IBM Multiprotocol Switched Services (MSS) Server Introduction and Planning Guide,* GC30-3820
- *IBM Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1: User's Guide and Protocol Reference,* SC30-3818
- *IBM Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 2: User's Guide and Protocol Reference,* SC30-3819
- *Event Logging System Messages Guide,* SC30-3682
- *IBM 8210 Nways Multiprotocol Switched Services (MSS) Server Service Manual,* GY27-0354

Summary of Changes For Version 1 Release 1.1

The following are the hardware enchancements for the IBM 8210 in this release:

- FDDI adapter
- V33.6 V/D/F Modem
- ATM adapter upgraded to enhance performance

The following are the software enhancements for the IBM 8210 in this release:

- Support for the FDDI adapter and the V33.6 modem
- Improvements to ELAN and LEC to support ATM UNI 3.0 and 3.1 signalling
- IBM LEC support
- Super ELAN support
- Support for bridging as described in RFC 1483
- Quality of Service (QoS) feature
- Next Hop Resolution Protocol (NHRP)
- Improved BUS frame throughput
- Support for virtual ATM interfaces
- AppleTalk 2 support
- Redundant IP gateway support
- ARP Server Redundancy support
- Helps for the World Wide Web interface
- Command History for the Command Line Interface

The technical changes and additions are indicated by a vertical line (|) to the left of the change.

Part 1. Configuring and Monitoring Bridge Functions

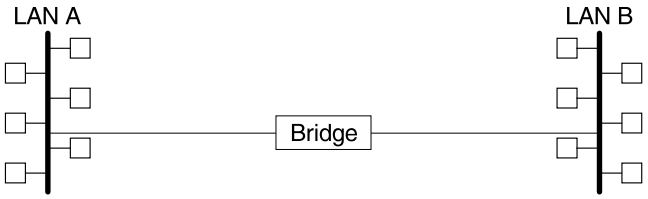
Chapter 1. Bridging Basics

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

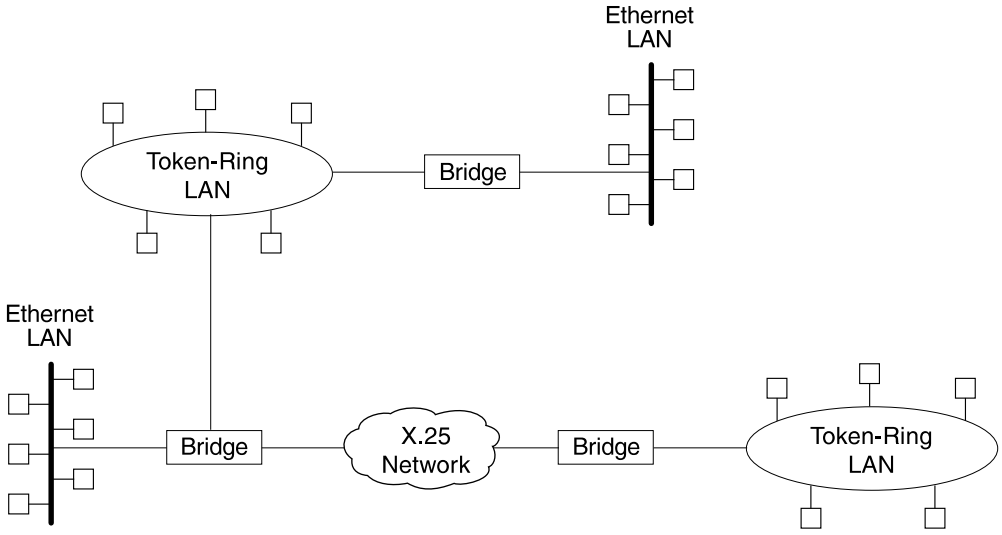
Bridging Overview

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 MAC header contained in the frame. Bridges originally linked two or more homogeneous networks. The term *homogeneous* means that the connected networks use the same bridging method and media types. Examples of these would be networks supporting the source routing bridging method ***only*** or transparent bridging algorithm ***only*** (these methods will be explained later).

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



Simple Bridge Connecting Two Homogeneous (Ethernet) LANs



Complex Bridging Configuration Connecting Different LAN Technologies

Figure 1-1. Simple and Complex Bridging Configuration

Bridges versus Routers

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

Router Connections

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

Bridge Connections

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

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

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

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

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

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

Types of Bridges

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

Simple Bridges

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

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

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

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

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

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

Complex Bridges

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

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

The adaptive source routing transparent (ASRT) bridge is the IBM 8210's implementation of bridge technology. The ASRT Bridge is a collection of software components capable of several of the bridging options just described and more. All of these functions are explained in greater detail later in this chapter.

Local Bridges

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

Remote Bridges

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

Remote bridges can differ from local bridges in several ways. One major difference is found in the speed at which data is transmitted. WAN connections may be slower than LAN connections. This difference in speed can make quite a difference when running time-sensitive applications. Another difference is found in the physical way in which remote and local bridges are connected to LANs. In local

bridges, the connections are made through local cabling media (for example, Ethernet, Thinet). Remote bridge connections are made over the serial lines.

Basic Bridge Operation

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

Operation Example 1: Local Bridge Connecting Two LANs

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

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

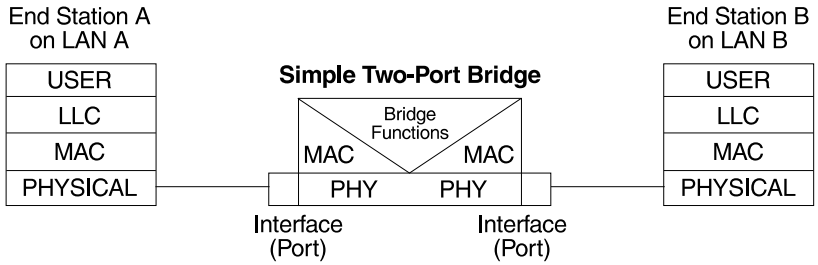


Figure 1-2. Two-Port Bridge Connecting Two LANs

Operation Example 2: Remote Bridging Over a Serial Link

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

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

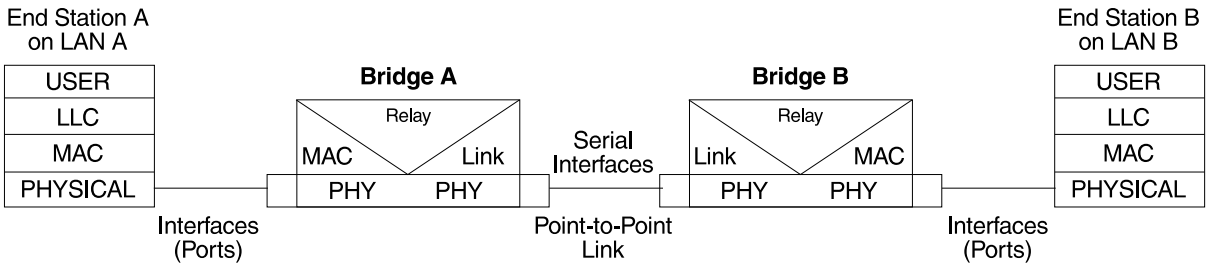


Figure 1-3. Bridging Over a Point-to-Point Link

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

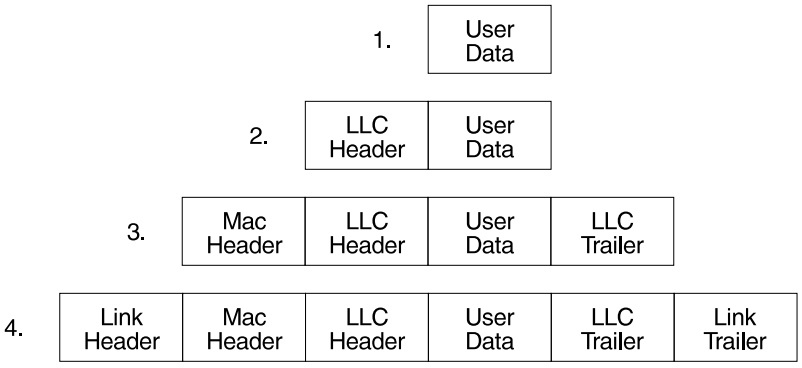


Figure 1-4. Data Encapsulation Over a Point-to-Point Link

Encapsulation proceeds as follows:

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

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

MAC Bridge Frame Formats

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

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

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

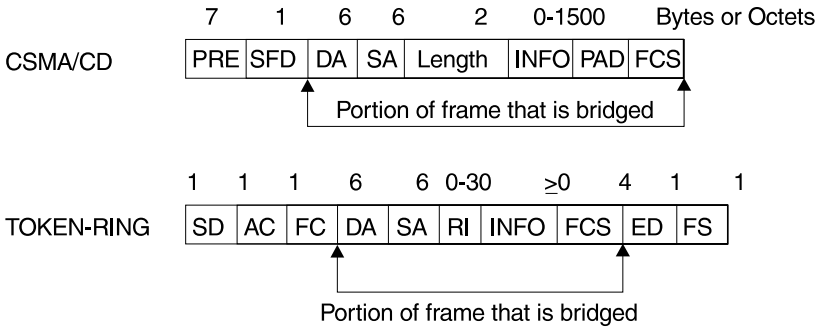


Figure 1-5. Examples of MAC Frame Formats

CSMA/CD (Ethernet) MAC Frames

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

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

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

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

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

Token-Ring MAC Frames

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

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

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

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

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

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

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

Chapter 2. Bridging Methods

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

Transparent Bridging

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

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

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

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

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

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

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

Routers and Transparent Bridges

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

During this operation, the following actions occur:

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

Network Requirements

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

Transparent Bridge Operation

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

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

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

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

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

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

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

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

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

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

Shaping the Spanning Tree

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

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

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

Table 2-1. Spanning Tree Default Values

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

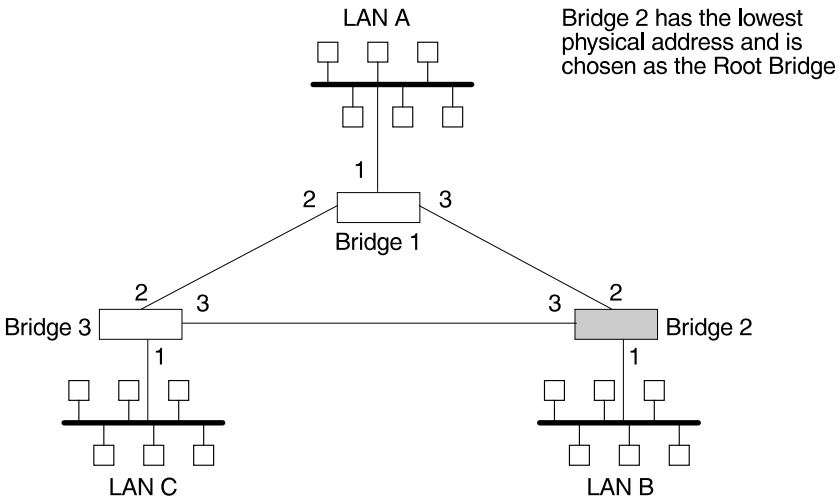


Figure 2-1. Networked LANs Before Spanning Tree

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

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

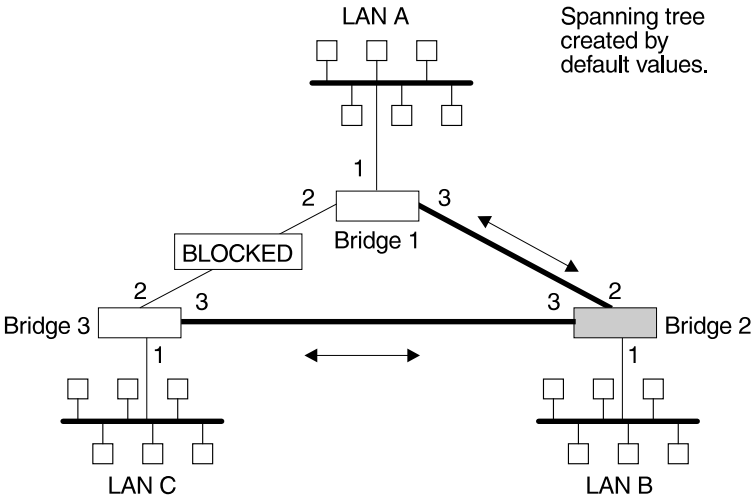


Figure 2-2. Spanning Tree Created With Default Values

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

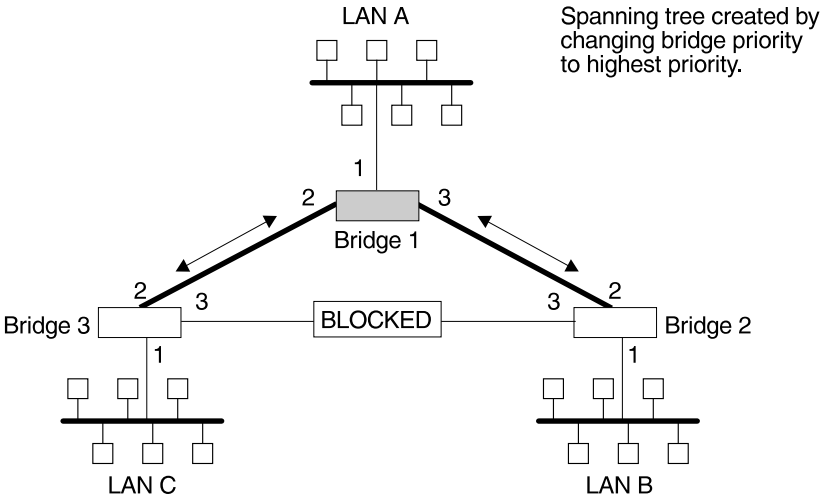


Figure 2-3. User-Adjusted Spanning Tree

Spanning Tree Bridges and Ethernet Packet Format Translation

The 8210 Spanning Tree Bridge protocol provides packet forwarding for the bridging routers in accordance to IEEE Standard 802.1D-1990 Media Access Control (MAC) bridges. The protocol also provides appropriate header translation for Ethernet packets.

An Ethernet/IEEE 802.3 network can simultaneously support the Ethernet data link layer and the IEEE 802.2 data link layer, based on the value of the length/type field in the MAC header. The bridge must translate to and from Ethernet format to

provide transparency across mixed LAN types. The algorithm used is based on emerging IEEE standards.

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

IBM RT Feature for SNA Traffic

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

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

UB Encapsulation of XNS Frames

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

Transparent Bridging and ATM

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

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

Transparent Bridge Terminology and Concepts

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

Aging Time

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

Bridge

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

Bridge Address

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

Bridge Hello Time

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

Bridge Forward Delay

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

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

Bridge Identifier

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

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

Bridge Maximum Age

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

Bridge Priority

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

Designated Bridge

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

Designated Port

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

Filtering and Permanent Databases

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Root Port

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

Spanning Tree

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

Transparent Bridging

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

Source Routing Bridging (SRB)

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

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

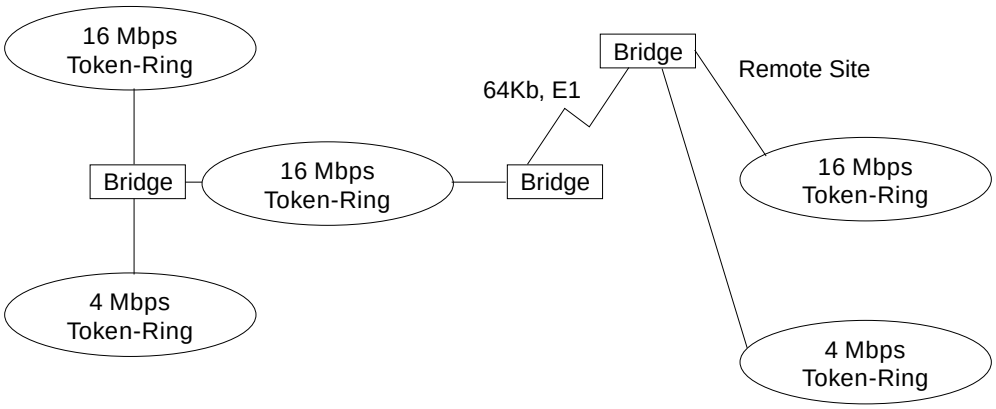


Figure 2-4. Example of Source Routing Bridge Connectivity

Among its features, the source routing bridge provides:

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

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

Source Routing Bridge Operation

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

1. A source station sends out a frame and finds that the frame's destination is not on its own (local) segment or ring.
2. The source station builds a *route discovery* broadcast frame and transmits it onto the local segment.
3. All bridges on the local segment capture the route discovery frame and send it over their connected networks.

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

   When the broadcast frame finally reaches its destination, it contains the exact sequence of addresses from source to destination.
4. When the destination end station receives the frame, it generates a response frame including the route path for communication. Frames that wander to other parts of the bridged network (accumulating irrelevant routing information in the meantime) never reach the destination end station and no station ever receives them.
5. The originating station receives the learned route path. It can then transmit information across this established path.

Source Routing Frames

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

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

Because source routing MAC frames contain routing information necessary for data communication over multi-ring environments, they differ slightly in format from the

typical token-ring MAC frames. The presence of a “1” in the RII within the source address field indicates that a RIF containing routing information follows the source address. Figure 2-5 on page 2-12 provides a closer look at the format of the source address field of a source routing frame.

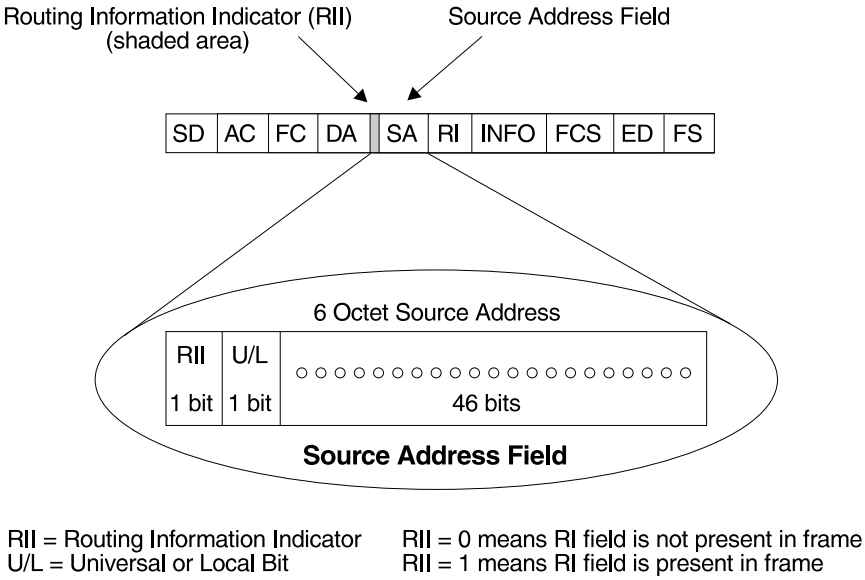


Figure 2-5. 802.5 Source Address Format

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

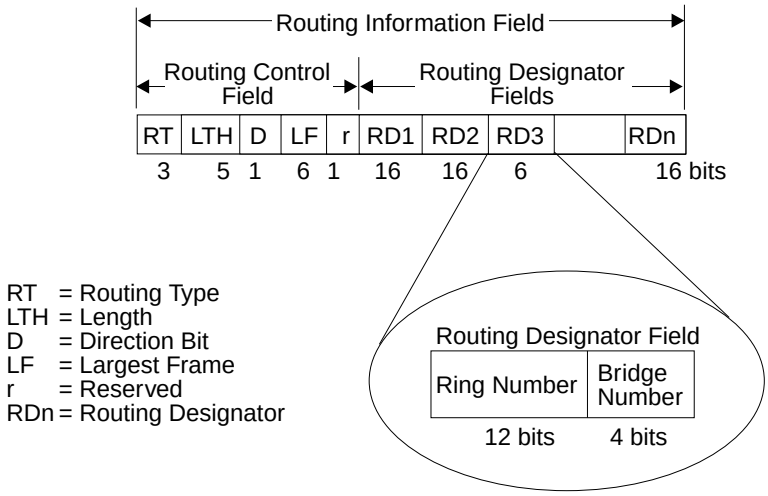


Figure 2-6. 802.5 Routing Information Field

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

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

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

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

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

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

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

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

  LF bit encoding is made up of a 3-bit base encoding and a 3-bit extended encoding (6 bits total). The SRT bridge (explained in a later section) contains

an LF mode indicator allowing the bridge to select either base or extended LF bits. When the LF mode indicator is set to the *base mode*, the bridge sets the LF bits in explorer frames with the largest frame base values. When the LF mode indicator is set to *extended mode*, the bridge sets the LF bits in explorer frames in with the largest frame extended values.

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

The Spanning Tree Explore Option

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

- Simulate a spanning tree network
- Balance traffic loads.

Simulating a Spanning Tree Network

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

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

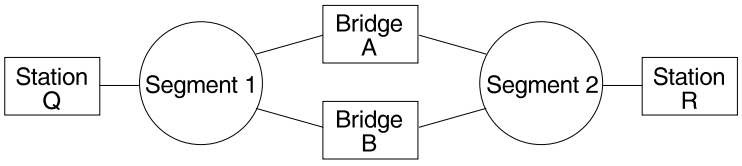


Figure 2-7. Example of Parallel Bridges

Balancing Traffic Loads

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

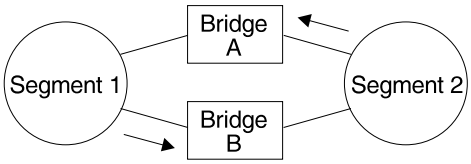


Figure 2-8. Using Spanning Tree Explore for Load Balancing

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

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

Protocol Filtering

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

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

Table 2-2. Route/Bridge Decision Table

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

Source Route Bridging and ATM

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

Source Routing Bridge Terminology and Concepts

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

Bridge Instance

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

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

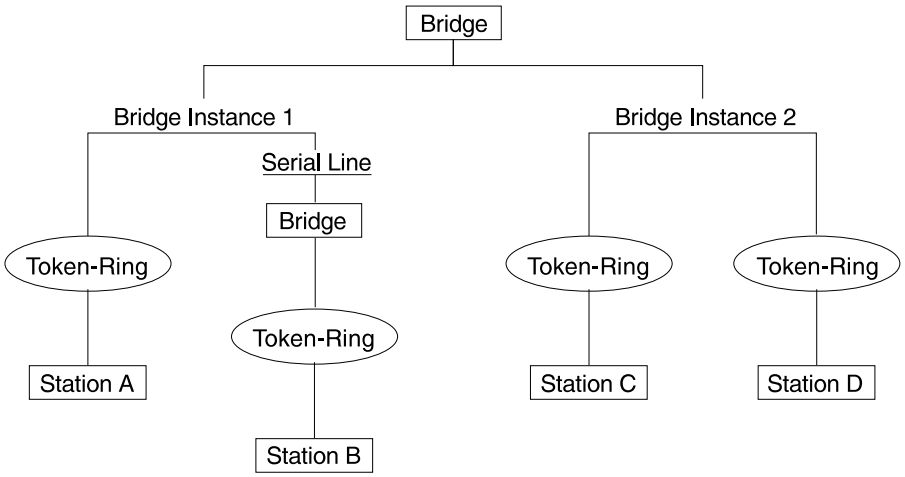


Figure 2-9. Bridge Instances within a Bridge

Bridge Number

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

Explorer Frames

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

Interface Number

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

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

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

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

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

Source Routing Transparent (SRT) Bridge

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

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

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

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

The following sections describe SRT Bridging in more detail:

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

General Description

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

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

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

Source Routing Transparent Bridge Operation and Architecture

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

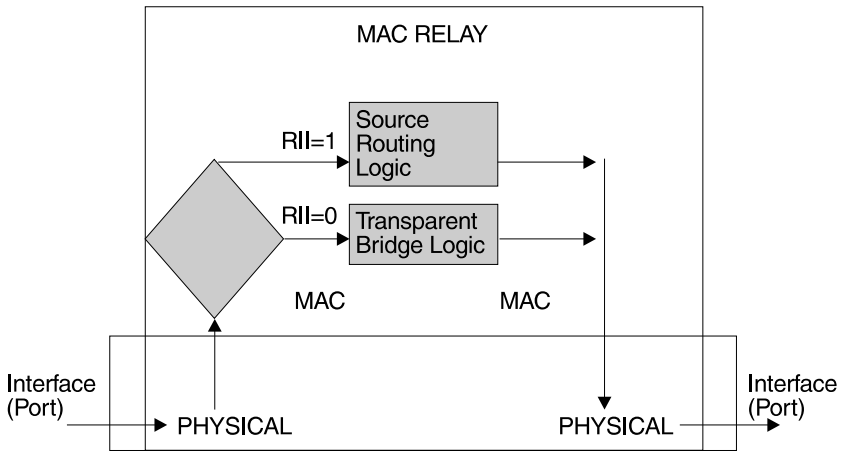


Figure 2-10. SRT Bridge Operation

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

SRT Bridge and ATM

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

Source Routing Transparent Bridge Terminology

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

Explorer Frames

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

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

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

Routing Information Field (RIF)

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

Routing Information Indicator (RII)

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

Source Routing

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

Spanning Tree

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

Transparent Bridging

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

ASRT Bridge Overview

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

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

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

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

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

The following sections describe the ASRT bridge in detail:

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

General Description

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

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

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

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

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

Source Routing—Transparent Bridge Operation

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

- Source routing domains
- Transparent bridging domains.

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

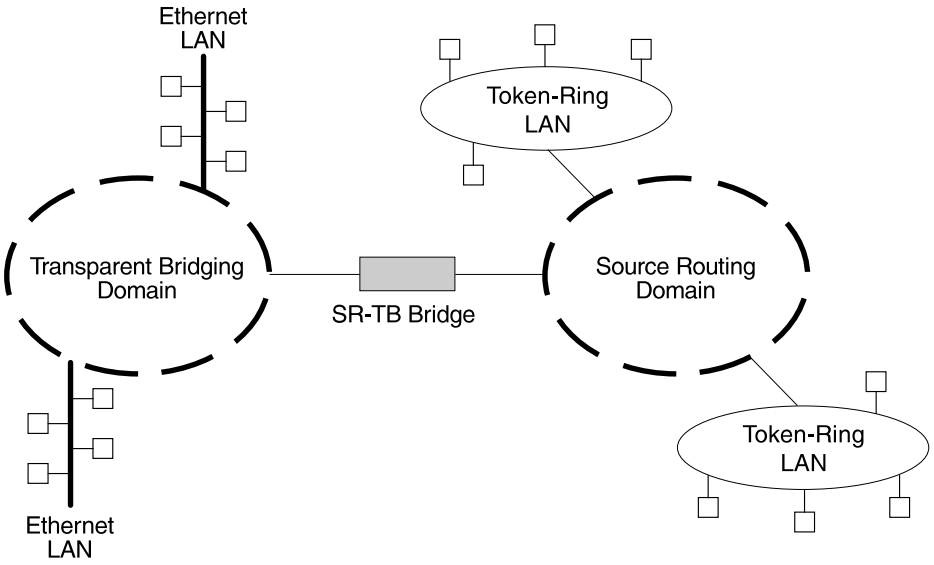


Figure 2-11. SR-TB Bridge Connecting Two Domains

Specific Source Routing and Transparent Bridging Operations

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

- Transparent bridging operations
- Source routing bridging operations.

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

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

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

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

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

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

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

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

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

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

SR-TB Bridging: Four Examples

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

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

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

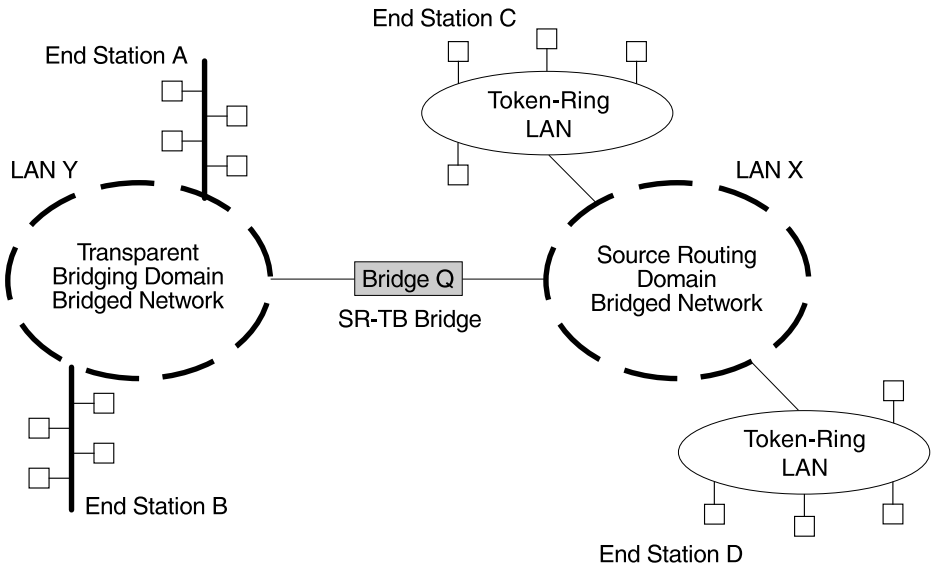


Figure 2-12. SR-TB Bridging Examples

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

SR-TB and 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 PVC.

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

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

All Routes Broadcast

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

All Stations Broadcast

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

Bridge

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

Bridge Number

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

Explorer Frames

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

Ring Number

The unique number identifying a ring in a bridged network.

Route

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

Route Designator

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

Route Discovery

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

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

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

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

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

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

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

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

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

Hardware Address Filtering
Another key feature provided by the ASRT bridge is hardware address filtering. Hardware address filtering solves the conflict in packet acknowledgement methods that exists in the Ethernet and token-ring LAN technologies. It occurs in the MAC layer and is the only technique that accurately sets acknowledgment bits based on the destination MAC address. The ASRT bridge uses content-addressable memories (CAMs) to implement hardware address filtering. This technology

effectively gives the bridge a higher level of intelligence by providing instantaneous lookup of MAC addresses without creating any performance penalty.

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

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

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

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

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

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

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

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

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

ASRT Configuration Considerations

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

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
STB	No	TB	TB	TB
SRB	No	SR	--	SR
STB & SRB	No	SR	TB	TB or SR
SR TB	Yes	SR	TB	TB
SR TB	Yes	SR	TB	SR
SRT	No	SR & TB	TB	SR & TB
ASRT	Yes	SR & TB	TB	SR & TB
ASRT	Yes	SR	TB	SR & TB
ASRT	Yes	SR or TB	TB	SR & TB

Bridge Personality Key:

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

Bridging Method Key:

SR = Source Routing TB = Transparent Bridging

Chapter 3. Bridging Features

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

Bridging Tunnel

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

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

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

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

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

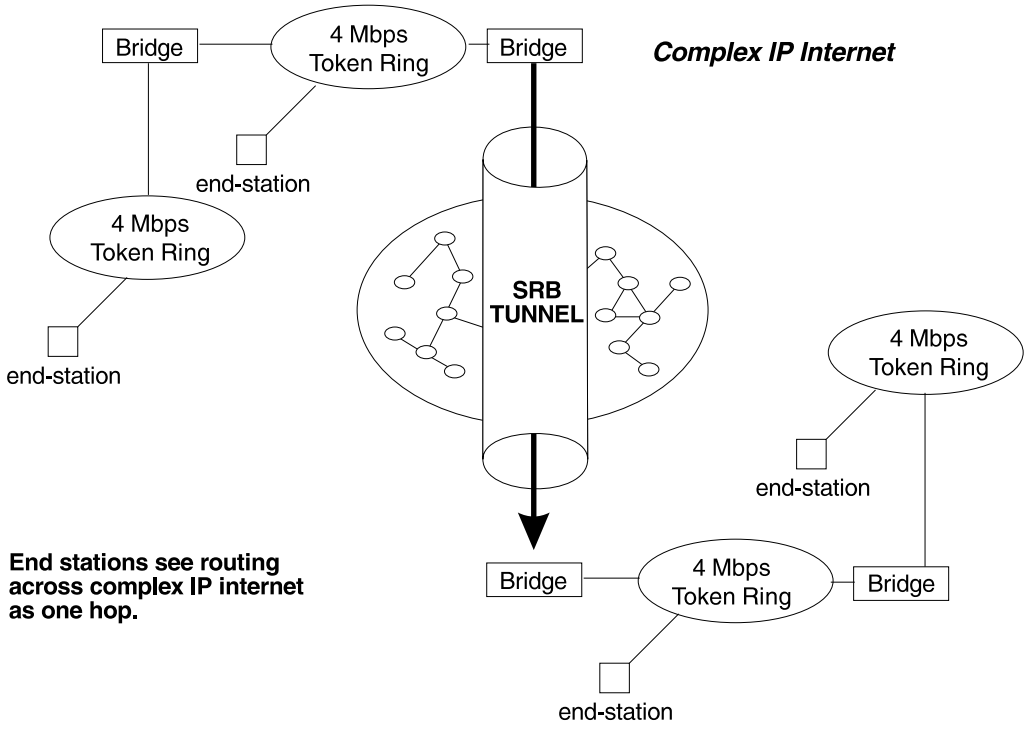


Figure 3-1. Example of the Bridge Tunnel Feature

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

Encapsulation and OSPF

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

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

With OSPF, tunnels automatically manage paths inside the internetwork. If a line or bridge fails along the path then the tunnel bridge automatically reroutes traffic along a new path. If a path is restored, the tunnel automatically updates to the best path. This rerouting is completely transparent to the end-stations. For more information on OSPF, see the configuration and monitoring chapters beginning at Chapter 16, “Using and Configuring OSPF” on page 16-1.

TCP/IP Host Services (Bridge-Only Management)

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

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

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

Bridge-only management function is activated by assigning an IP address to the bridge and enabling TCP/IP Host Services (see Chapter 11, "Configuring TCP/IP Host Services" on page 11-1). This IP address is associated with the bridge as a whole, instead of being associated with a single interface. When booting over the network, the bridge's IP address and a default gateway can be learned automatically through the ROMCOMM interface with the boot PROMs. Default gateway assignments can also be user-configured.

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

Bridge-MIB Support

For Bridge Management via SNMP, the IBM 8210 Multiprotocol Switched Services Server supports the management information bases (MIBs) as specified by RFC 1493 and RFC 1525, *except* for the following MIBs:

- dot1dStaticTable
- dot1dTpFdbTable
- dot1dPortPairTable

For additional information on the MIBs, refer to *NCE Configuration and Operations*.

NetBIOS Name Caching

The NetBIOS name caching feature enables the bridging router to significantly reduce the number of Name-Query frames that leave an originating ring and are forwarded through a bridge. Configuring for NetBIOS name caching is part of the NetBIOS configuration. Details are in "NetBIOS Name Caching and Route Caching" on page 8-6.

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 8-5.

NetBIOS Name and Byte Filters

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

Types of NetBIOS Filtering

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

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

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

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

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

Configuration and monitoring of NetBIOS is described at Chapter 8, "Using, Configuring, and Monitoring NetBIOS" on page 8-1. The remainder of this section describes NetBIOS host name filtering and NetBIOS byte filtering.

NetBIOS Host Name Filtering

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

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

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

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

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

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

NetBIOS Byte Filtering

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

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

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

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

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

Building a Filter

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

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

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

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

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

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

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

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

Simple and Complex Filters

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

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

Multiple Spanning Tree Protocol Options

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

Background: Problems with Multiple Spanning Tree Protocols

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

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

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

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

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

STP/8209

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

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

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

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

Threading (Router Discovery)

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

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

IP Threading with ARP

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

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

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

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

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

IPX Threading

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

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

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

AppleTalk 2 Threading

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

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

Chapter 4. Basic Bridging Configurations

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

- "Accessing the ASRT Configuration Environment"
- "Basic Bridging Configuration Procedures"

If you need more information about the ASRT bridge configuration commands, refer to the Chapter 6, "Configuring Bridging."

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

For examples of setting up NetBIOS filtering, see "NetBIOS Host Name and Byte Filtering Configuration Procedures" on page 8-8.

Accessing the ASRT Configuration Environment

For information on how to access the ASRT configuration environment, see "Getting Started" in the *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

Basic Bridging Configuration Procedures

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

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

Bridging Interfaces

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

- Ethernet
- Token-Ring
- Serial Line

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

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

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

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

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

Enabling the Transparent Bridge

Use the following commands to enable transparent bridging:

- **Enable bridge** to enable transparent bridging on all Local Area Network (LAN) interfaces.
- **Disable transparent** *port#* to exclude specified token-ring interfaces from participating in transparent bridging. Repeat the command for all interfaces you want excluded from the transparent bridging configuration.

Enabling the Source Routing Bridge

Use the following commands to enable source-routing bridging:

- **Enable bridge** to enable bridging on all local area network interfaces.
- **Disable transparent** *port#* to disable transparent bridging on all ports.
- **Enable source-routing** *port# segment# [bridge#]* to enable source routing for given ports. When source routing is enabled on more than two ports, an additional segment number is required to assign an internal virtual segment needed for 1:N SRB configurations.

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

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

Enabling the SR-TB Bridge

Use the following commands to enable SR-TB bridging:

- **Enable bridge** to enable bridging on all local area network interfaces.
- **Disable transparent** *port#* to disable transparent bridging on all underlying source routing interfaces.
- **Enable source routing bridge** *port# segment# [bridge#]* to enable source routing for given ports. When source routing is enabled on more than two

ports, an additional segment number is required to assign an internal virtual segment needed for 1:N SRB configurations.

- **Enable sr-tb-conversion** *segment#* to enable conversion of source-routed frames to transparent frames and vice versa. You are also required to assign a domain segment number and a domain MTU size to represent the entire transparent (Ethernet) bridging domain.

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

For more information on all of the commands just mentioned, refer to Chapter 6, “Configuring Bridging.”

Chapter 5. Overview of Routing and Bridging Over ATM

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

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 server supports IP and IPX routing as illustrated in Figures 5-1 and 5-2.

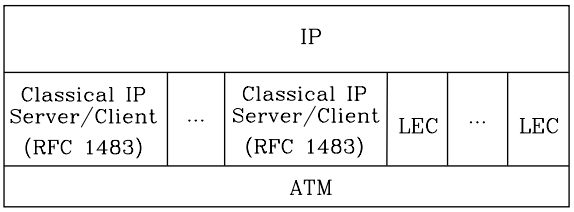


Figure 5-1. IP Routing in the server

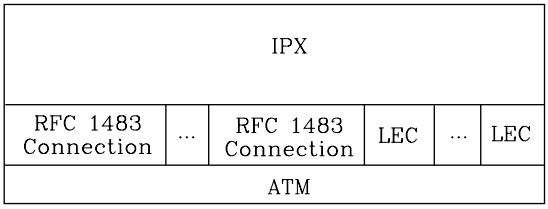


Figure 5-2. IPX Routing in the server

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

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

Overview of Bridging

The 8210 supports bridging over emulated Ethernet and token-ring interfaces as well as over native ATM (RFC 1483) as illustrated in Figure 5-3 on page 5-3 and Figure 5-4 on page 5-3. The operational characteristics of bridging over these emulated interfaces are identical to those over legacy LAN interfaces. Emulated interfaces have interface numbers and bridge port numbers. In the case of bridging

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

over native ATM, multiple (virtual) ports may be configured on a single interface. Each bridge port has a particular behavior, and the bridge has an overall behavior.

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

Bridging Behaviors

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

Pure TB

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

Pure SR

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

SR and TB

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

SR-TB

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

SRT

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

ASRT

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

Resolving LAN Destination Addresses via LE_ARP_REQUESTS

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

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

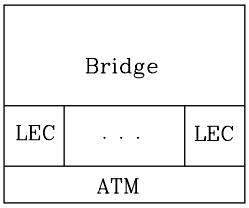


Figure 5-3. Bridging Over the Emulated LAN Interface

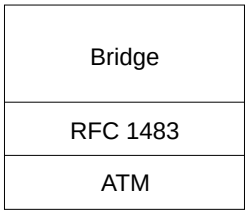


Figure 5-4. Bridging Over Native ATM

Overview of RFC 1483 Support

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

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

Overview of RFC 1483 Support for Routing

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

RFC 1483 Support for IPX Routing

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

The 8210, like other routers that support RFC 1483 LLC/SNAP encapsulation on ATM, can be interconnected in full or partial meshes using manually configured RFC 1483 connections.

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

Both permanent virtual circuits (PVCs) and configured switched virtual circuits (SVCs) are supported. However, virtual channel connections (VCCs) to IPX routers must be dedicated to IPX; they cannot be shared with other protocols, such as IP. 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 server 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 server 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 server sets the IPX host number to the ESI component of the associated ATM address. Whenever you do not explicitly configure the ESI, it defaults to the MAC address that is burned into the ATM interface hardware.

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

Routers that are interconnected in a partial mesh and are providing intermediate routing between routers on the same ATM interface should disable IPX split-horizon on the ATM interface. Doing this ensures that RIP and SAP properly inform the

interconnected routers of all available routes and services. Routers that are interconnected in a full mesh need not disable split-horizon.

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

Note: Though the implementation limits IPX to one address per ATM interface, a single physical ATM interface can still support multiple IPX addresses using the ATM Virtual Interface. You will still be limited to one IPX address per interface, but since you can add several ATM Virtual Interfaces on each physical interface, you can avoid the single address limitation.

Refer to *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1* for additional information about ATM Virtual Interfaces.

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 two 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 and 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.

The main difference between configuring bridging on a port on an ATM interface and configuring one on a legacy LAN is that a VCC must be specified when a bridge port is configured on the ATM interface. In addition, the software implementation allows multiple bridge ports to be configured on a single physical ATM interface. Thus, bridge ports configured on an ATM interface are treated as virtual ports. For the initial implementation of Bridging support, the following restrictions apply:

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

- A maximum of 32 bridge ports may be configured on a single ATM interface
- Only PVCs may be associated with a bridge port.
- Support is not currently provided for ATM virtual interfaces as described in "ATM Virtual Interface Configuration Concepts" in the "Using and Configuring ATM" chapter of *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*. Only "real" interfaces are supported.

To configure a bridge port on an ATM interface, the VPI and VCI must be specified for the PVC that will be associated with that port.

Once a port has been added on an ATM interface (see "Add" on page 6-3) the bridging configuration commands that require a port number as a parameter can be used with this port number.

Refer to Chapter 22, "Using and Configuring ARP" on page 22-1 for additional information on configuring bridging over ATM.

Chapter 6. Configuring Bridging

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

Accessing the ASRT Configuration Environment

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

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

ASRT Configuration Commands

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

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

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

- Enter the configuration commands for super ELAN using the `ASRT config>set super-elan-id` command.
- Enter the configuration commands for IP Bridging Broadcast Manager (BBCM) at the `IP B-BCM config>` prompt. Select the BBCM protocol, either IP or NetBIOS, using the ASRT `config> broadcast-manager` command. The BBCM commands are explained later in this chapter.
- Enter the configuration commands for dynamic protocol filtering (Virtual LANs) at the `VLAN config>` prompt. The VLAN prompt is accessed by entering the ASRT **VLANS** command explained later in this chapter.
- Enter configuration commands for IP tunnels at the `TNL config>` prompt. The tunnel prompt is a subset of the major ASRT commands and is accessed by entering the ASRT **tunnel** command explained later in this chapter.
- Enter configuration commands for NetBIOS at the `NetBIOS config>` prompt. The NetBIOS prompt is a subset of the major ASRT commands and is accessed by entering the ASRT **netbios** command explained later in this chapter.

- Enter configuration commands for NetBIOS Filtering at the `NetBIOS Filter config>` prompt. This prompt is a subset of the NetBIOS commands.
- Enter bridging configuration commands for ATM at the `ASRT config>` prompt.

Table 6-1 shows the ASRT configuration commands.

Table 6-1 (Page 1 of 2). ASRT Configuration Command Summary

Command	Function
? (Help)	Lists all of the ASRT configuration commands, or lists the options associated with specific commands.
Add	Adds station address entries to the permanent database, specific address mapping, LAN/WAN ports, protocol filters, and a tunnel between end stations across an IP internetwork.
Ban	Allows access to the boundary access node (BAN) configuration prompt so that BAN configuration commands can be entered.
Broadcast-Manager	Allows the user to configure Bridging Broadcast-Manager for either IP or NetBIOS.
Change	Allows the user to change bridge and segment numbers.
Delete	Deletes station address entries, specific address mapping, LAN/WAN ports, protocol filters, and a tunnel between end stations across an IP internetwork.
Disable	Disables the following functions: • Bridging • Duplicate frames • Mapping between group and functional addresses • Propagation of Spanning Tree Explorer Frames • Source routing on a given port • Reception of spanning tree explorer frames over a tunnel • Conversion of source routed to transparent frames • Transparent (spanning tree) bridging function on a given port • Tunnel between bridges
Enable	Enables the following functions: • Bridging • Duplicate frames • Mapping between group and functional addresses • Propagation of Spanning Tree Explorer Frames • Source routing on a given port • Reception of spanning tree explorer frames over a tunnel • Conversion of source routed to transparent frames • Transparent (spanning tree) bridging function on a given port • Tunnel between bridges
List	Displays information about the complete bridge configuration or about selected configuration parameters.
NetBIOS	Displays the NetBIOS configuration prompt.

Table 6-1 (Page 2 of 2). ASRT Configuration Command Summary

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
Tunnel	Allows access to the tunnel configuration prompt so that tunnel configuration commands can be entered.
VLANS	Allows the user to configure dynamic protocol filtering
Exit	Exits the ASRT configuration process and returns to the CONFIG environment.

? (Help)

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

Syntax: ?

Example: **?**

```
ADD
BAN
BROADCAST-MANAGER protocol
CHANGE
DELETE
DISABLE
ENABLE
LIST
NETBIOS
SET
TUNNEL
VLANS
EXIT
```

Example: **set ?**

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

Add

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

- Station address entries to the permanent database
- Specific address mapping for a given protocol
- LAN/WAN ports

- Protocol filters that selectively filter packets based on their protocol type
- IP tunnel between end-stations and across IP network segments

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

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

address *addr-value*

Adds unique station address entries to the permanent database. These entries are copied into the filtering database as permanent entries when the bridge is restarted. The *addr-value* is the MAC address of the desired entry. It can be an individual address, multicast address, or broadcast address. You are also given the option to specify the outgoing forwarding port map for each incoming port. Permanent database entries are not destroyed by the power off/on process and are immune to the aging settings. Permanent entries cannot be replaced by dynamic entries.

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

Default Value: none

Example: `add address`

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

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

Exclude destination address ... This prompt lets you set destination address filtering for that entry. Answering "Yes" to the prompt causes filtering of any frames that

contain this address as a destination address no matter which port it came from.

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

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

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

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

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

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

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

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

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

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

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

Enabling Destination Address Filtering For Entry

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

type-field Protocol type field.

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

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

Common values are:

Protocol - SAP (hexadecimal value)

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

DSAP Default Value: 1

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

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

Protocol - Ethernet type (hex value)

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

Ethernet Default Value: 1

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

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

Common values are:

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

SNAP Default Value: 00 0000 0800

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

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

Default Value: none

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

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

Default Value: none

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

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

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

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

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

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

port *interface port-num VPI VCI*

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

If you are adding a port on an ATM interface you need to specify the VPI and VCI for the PVC that will be associated with that port.

Port Number Valid Values: 1 to 254

Port Number Default Value: none

VPI Valid Values: 0 to 255

VPI Default Value: 0

VCI Valid Values: 0 to 65535

VCI Default Value: 0

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.

Refer to Chapter 22, "Using and Configuring ARP" on page 22-1 for ATM ARP client configuration information.

Example 1: add a port

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

Example 2: add a port on an ATM interface

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

prot-filter snap ether dsap

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

This parameter specifies protocol identifiers for which the received frames of that specific protocol are discarded exclusively without applying bridge logic. ARP packets for this protocol type will also be discarded. The protocol filter is

applied only on the received packets. The protocol filters available include the following:

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

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

DSAP Types

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

SNAP Protocol Identifiers

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

Ethernet Types

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

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

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

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

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

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

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

tunnel *port#*

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

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

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

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

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

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

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

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

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

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

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

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

Example: `add tunnel 3`

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

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

Broadcast-Manager protocol

Use the **broadcast-manager** command to configure Bridging Broadcast-Manager for IP or NetBIOS.

Syntax: broadcast-manager *protocol*

Example: `broadcast-manager ip`

```
ASRT config>broad
Enter Bridge Broadcast Manager Protocol: (IP or NetBIOS) [IP]?  IP
IP Bridge Broadcast Manager User Configuration
IP B-BCM config>?
DISABLE ip b-bcm
ENABLE ip b-bcm
LIST configuration
SET cache age timeout
EXIT
```

Example 2: `broadcast-manager netbios`

```
NetBIOS Support User Configuration
NetBIOS config>
```

Note: The NetBIOS option takes you to the NetBIOS configuration option.

| See “Bridging Broadcast Manager Configuration Commands” on page 6-45 for a
| description of Bridging Broadcast Manager configuration commands.

Change

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

Syntax: change bridge . . .
segment . . .

bridge *new-bridge#*

Changes bridge numbers in the bridging configuration.

Example: `change bridge 3`

segment *old-segment# new-segment#*

Changes bridge numbers in the bridging configuration.

Example: `change segment 2 3`

Delete

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

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

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

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

address *addr-value*

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

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

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

| **Default Value**: none

Example: `delete address`

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

Deletes specific address mapping for given protocol.

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

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

port *port#*

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

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

Example: `delete port 2`

prot-filter snap ether dsap

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

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

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

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

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

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

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

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

Disable

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

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

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

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

bridge

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

Example: `disable bridge`

duplicate *frame-type*

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

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

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

Example: `disable duplicate TSF`

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

ethertype-ibmrt-pc

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

Example: `disable ethertype-ibmrt-pc`

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

fa-ga-mapping

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

Example: `disable fa-ga-mapping`

ibm8209_Spanning_Tree

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

Example: `disable IBM8209_spanning_tree`

spanning-tree-explorer *port#*

Disables a port from allowing propagation of spanning tree explorer frames if source routing is enabled. This command is used only if transparent bridging is

not enabled on the port. In that case, it is automatically known in conformance with the transparent spanning tree.

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

source-routing *port#*

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

Example: **`disable source-routing 2`**

sr-tb-conversion

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

Example: **`disable sr-tb-conversion`**

stp

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

Example: **`disable stp`**

| super-elan-bridging *port#*

| Disables the Super ELAN participation on the bridge port.

| **Example:** **`disable super`** 1

transparent *port#*

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

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

Example: **`disable transparent 2`**

tree *port#*

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

Example: **`disable tree 1`**

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

ub-encapsulation

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

Example: **`disable ub-encapsulation`**

Enable

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

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

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

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

bridge

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

Example: **`enable bridge`**

duplicate *frame-type*

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

Example: **`enable duplicate STE`**

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

ethertype-ibmrt-pc

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

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

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

fa-ga-mapping

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

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

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

IBM8209_Spanning_Tree

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

Example: **`enable IBM8209_spanning_tree`**

spanning-tree-explorer *port#*

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

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

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

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

port# — Valid port participating in the bridge configuration.

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

Default Value: 1

segment# — 12-bit number that represents the LAN/WAN to which media is attached. All the media on other bridges attached to this LAN/WAN must be configured with the same value. For correct operation of the source routing function, it is very important that all the bridges attached to this LAN/WAN have the same perspective of the LAN/WAN identification value.

bridge# 4-bit value unique among all the bridges attached to the same LAN/WAN. This value is required when source routing is enabled on the first interface. For later interfaces, this input is optional. It is recommended that the bridge# be unique on the segment.

Valid Values: X'0' to X'F'

Default Value: 1

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

Example: **`enable source-routing 2 1 1`**

sr-tb-conversion

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

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

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

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

TB-Domain Segment Number Default Value: 1

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

TB-Domain MTU Valid Values: 576 to 18000 bytes

TB-Domain MTU Default Value: 2048

Example: **`enable sr-tb-conversion`**

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

stp

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

Example: **`enable stp`**

super-elan-bridging *port# super-elan-id*

This option enables the bridge port for super ELAN support. Super ELAN support allows LAN Emulation clients on different ELANs to communicate directly with each other. Clients may establish data direct VCCs with any other client in the super ELAN, even though the clients may be defined on different ELANs. Once the data direct VCC is established, bridge support is not required to forward data frames.

The Super ELAN function can be enabled only for Transparent bridge ports that have an ATM physical interface and are identified with Ethernet or Token-Ring LAN Emulation Clients.

Example: **`enable super`** `1 1`

transparent *port#*

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

Example: **`enable transparent`**

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

tree *port#*

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

Example: **`enable tree 1`**

ub-encapsulation

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

Example: **`enable ub-encapsulation`**

List

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

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

address *addr value*

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

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

Default Value: none

Example: **`list address 000000123456`**

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

Address Address entry in 12-digit hexadecimal format.

Entry Type

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

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

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

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

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

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

bridge

Lists all general information regarding the bridge.

Example: **`list bridge`**

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

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

STP Participation:      IEEE802.1d and IBM-8209
                 +-------------------------+
-----------------| TRANSLATION INFORMATION |----------------------------
                 +-------------------------+
FA<=>GA Conversion:     Enabled                UB-Encapsulation:  Disabled
                 +------------------+
-----------------| PORT INFORMATION |-----------------------------------
                 +------------------+
Number of ports added:  3
Port:   1      Interface:     0      Behavior:   STB only    STB:  Enabled
 VPI:   0      VCI:          48
Port:   2      Interface:     1      Behavior:   STB & SRB   STB:  Enabled
Port:   3      Interface:     2      Behavior:   STB & SRB   STB:  Enabled
```

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

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

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

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

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

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

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

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

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

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

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

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

SRB LF-bit interpretation
Indicates the largest Frame (LF) bit encoding interpretation mode if source routing is enabled in this bridge. This is listed as either BASIC or EXTENDED.

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

Spanning Tree Protocol Participation
Displays the types of spanning tree protocols that the bridge participates in.

Number of ports added
Number of bridge ports added to the bridging configuration.

Port Number
A user-defined number assigned to an interface by the Add Port command.

Interface Number
Identifies devices connected to a network segment through the bridge. You must add at least two interfaces to participate in bridging. An interface number of 255 is used for bridging.

Port Behavior
Indicates method of bridging being used by that port. The values include STB for Transparent, SRB for Source Routing, and SR-TB for Source Routing Transparent conversion bridging.

filtering *datagroup-option*

The following general data groups can be displayed under the **list filtering** command:

All Displays all filtering database entries.

Ethertype Displays Ethernet protocol type filter database entries.

SAP Displays SAP protocol filter database entries.

SNAP Displays SNAP protocol identifier filter database entries.

The following examples illustrate each of the **list filtering** display options.

Example 1: `list filtering all`

```
Ethernet type 0800 is routed on ports 1
IEEE 802.2 destination SAP 42 is routed on ports 1
IEEE 802 SNAP PID 00-00-00-08-00 is routed on ports 2-3
```

Descriptors used in explaining how packets are communicated include the following:

Routed
Describes packets passed to routing forwarder to be forwarded.

Filtered
Describes packets which are administratively filtered by the user setting protocol filters.

Bridged and routed
This describes a protocol identifier for which there is a protocol entity within the system which is not a forwarder. For example a link level echo protocol. Unicast packets from this protocol are bridged or locally processed if being sent to a registered address. Multicast packets are forwarded and locally processed for a registered multicast address.

All of these descriptors also apply to ARP packets with this Ethertype.

Example 2: `list filtering ethertype`

```
Ethernet type (in hexadecimal), 0 for all [0]? 0800
Ethernet type 0800 is routed on ports 1
```

Example 3: `list filtering sap`

```
SAP (in hexadecimal), 100 for all [100]? 42
IEEE 802.2 destination SAP 42 is routed on ports 1
```

Example 4: `list filtering snap`

```
SNAP Protocol ID, return for all [00-00-00-00-00]?
IEEE 802 SNAP PID 00-00-00-08-00 is routed on ports 2-3
```

mapping *add-type type-field*

Lists specific address mapping for a given protocol.

Example: `list mapping SNAP`

```
PROTOCOL TYPE          GROUP ADDRESS          FUNCTIONAL ADDRESS
=============          =============          ==================
123456-7890            12-34-56-78-90-12      12:34:56:78:90:12
```

add-type
Choice of either DSAP, Ether (Ethernet), or SNAP.

type-field
Protocol type field:

- Destination Service Access Point (DSAP) protocol type is entered in a range of 1–FE (hexadecimal).
- Ethernet (Ether) protocol type is entered in a range of 5DD–FFFF (hexadecimal).
- Subnetwork Access Protocol (SNAP) protocol type is entered in 10-digit hexadecimal format.

permanent

Displays the number of entries in the bridge's permanent database.

Example: `list permanent`

```
Number of Entries in Permanent Database:  17
```

port *port#*

Displays port information related to ports that are already configured. Port# selects the port you want to list. Specifying no number selects all ports.

Example: `list port`

```
Port Id (dec)    : 128: 5, (hex): 80-05
Port State       : Enabled
STP Participation: Enabled
Port Supports: NO Bridging
Assoc Interface : 1
Path Cost       : 0
    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    Port ID (dec)    : 128:02, (hex): 80-02
    Port State       : Enabled
    STP Participation: Enabled
    Port Supports    : Transparent Bridging Only
    Assoc Interface  : 0  VPI  0  VCI:    78
    Path Cost        : 0
    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    Port ID (dec)    : 128:03, (hex): 80-03
    Port State       : Enabled
    STP Participation: Enabled
    Port Supports    : Transparent Bridging Only
    Assoc Interface  : 2
    Super ELAN bridging: Enabled       Super ELAN ID: 1
```

Port ID
: The ID consists of two parts: the port priority and the port number. In the example, 128 is the priority. 1, 2, and 3 are the port numbers. In hexadecimal format, the low-order byte denotes the port number and the high-order byte denotes the priority.

Port state
: Displays current state of the specified port or ports. This can be either ENABLED or DISABLED.

Port supports
: Displays bridging method supported by that port (for example, transparent bridging, source routing bridging).

SRB
: Displayed only when SRB is enabled and lists source routing bridging information. This includes the SRB segment number (in hex), the Maximum Transmission Unit size, and whether the transmission of spanning tree explorer frames is enabled or disabled.

Duplicate Frames Allowed
: Displays a breakdown and count of the types of duplicate frames allowed.

Assoc interface
: Displays interface number associated with the displayed port, and the VPI/VCI for the associated PVC if an ATM port.

Path Cost
: Cost associated with the port which is used for possible root path cost. The range is 1 to 65535.

Super ELAN Bridging
: Displays Super ELAN status for the bridge port and Super ELAN ID to which the port is assigned. The range of values for Super ELAN ID is 1 to 65535.

prot-filter *port#*

Reads a current list of the filter protocol types. Filters can be listed selectively by port or all ports can be displayed at once. Port# selects the bridge port that you want to list.

Example: `list prot-filter 1`

```
PORT 1
Protocol Class   :  DSAP
Protocol Type    :  01
Protocol State:  :  Filtered
Port Map         :  1, 2, 3
= = = = = = = = = = = = = = = = = = = = = = =
```

Port Number	Port number is displayed for each port if all ports are selected to be displayed.
Protocol Class	Displays protocol class either SNAP, Ether, or DSAP.
Protocol Type	Displays protocol ID in hexadecimal format.
Protocol State	Denotes that protocol is being filtered for selected port.
Port Map	Displays the numbers of the ports where this type of protocol filter is present.

protocol

Displays bridge information related to the spanning tree protocol.

Example: `list protocol`

```
Bridge Identifiers: 32768/00000000003
Bridge-Max-Age (in seconds): 20
Bridge-Hello-Time (in seconds): 2
Bridge-Forward-Delay (in seconds): 15
```

Note: Each of these bridge related parameters is also described in detail in the previous chapter.

Bridge Identifier	8-byte value in ASCII format. If you did not set the bridge address prior to displaying this information, the low order 6 bytes will be displayed as zero, denoting that the default MAC address of a port is being used. When a bridge has been selected as the root bridge, the bridge max age and bridge hello time are transmitted by it to all the bridges in the network via the HELLO BPDUs.
Bridge-Max-Age	Maximum age (period of time) that should be used to time out spanning tree protocol related information.
Bridge-Hello-Timer	Time interval between HELLO BPDUs.
Bridge-Forward-Delay	Time interval used before changing to another state (should this bridge become the root).

range *start-index stop-index*

Reads a range of address entries from the permanent database. To do this, first determine the size of the database by using the **list permanent** command. From this value you can then determine a “start index” value for your entry range. The start index is in the range from 1 to the size of the database. You can then choose a “stop index” for displaying a limited number of entries. This input is optional. If the stop index is not provided the default value is the size of the database.

Address entries contain the following information:

Example: `list range`

```
Start-Index [1]? 1
Stop-index [17]? 6
ADDRESS                 ENTRY TYPE      PORT MAP

=======                 ==========      ========
01-80-C2-00-00-00       REGISTERED      Input Port:  ALL PORTS
                                        Output ports:

01-80-C2-00-00-01       RESERVED        NONE/DAF
01-80-C2-00-00-02       RESERVED        NONE/DAF
01-80-C2-00-00-03       RESERVED        NONE/DAF
01-80-C2-00-00-04       RESERVED        NONE/DAF
01-80-C2-00-00-05       RESERVED        NONE/DAF
```

Address

6-byte MAC address of the entry.

Type of Entry

Specifies one of the following types:

- Reserved - entries reserved by the IEEE 802.1d committee
- Registered - entries consist of unicast addresses belonging to proprietary communications hardware attached to the box or multicast addresses enabled by protocol forwarders
- Permanent - entries entered by the user in the configuration process which survive power on/offs or system resets
- Static - entries entered by the user in the console process which do not survive power on/offs or system resets and are ageless
- Dynamic - entries “learned” by the bridge “dynamically” which do not survive power on/offs or system resets and which have an “age” associated with the entry
- Free - locations in database that are free to be filled by address entries

Port Map

Displays outgoing port map for all incoming ports.

NetBIOS

Displays the NetBIOS configuration prompt. Enter **netbios** at the `ASRT config>` prompt to display the NetBIOS configuration prompt. See “NetBIOS Commands” on page 8-15 for an explanation of each of the NetBIOS configuration commands.

Syntax: netbios

Example: `netbios`

```
NetBIOS Support User Configuration
NetBIOS config>
```

Set

Use the **set** command to set certain values, functions, and parameters associated with the bridge configuration. These include the following:

- Aging time for dynamic address entries in the filtering database
- Bridge address
- Largest Frame (LF) bit encoding interpretation for source routing
- MAC service data unit (MSDU) size

- Spanning tree protocol bridge and port parameters
- Route Descriptor (RD) limit
- Size of the bridge filtering database.

Syntax: set
- age
- bridge
- filtering
- lf-bit-interpretation . . .
- maximum-packet-size . . .
- port
- protocol bridge
- protocol port . . .
- route-descriptor-limit . . .
- super-elan-id

age *seconds resolution*

Sets the time for aging out dynamic entries in the filtering database when the port with the entry is in the forwarding state. This age is also used for aging RIF entries in the RIF table in the case of an SR-TB bridge personality.

Enter the desired value after each prompt and press **Return**.
It can be an individual address, multicast address, or broadcast address. Permanent database es are not destroyed by the power off/on process and are immune to the aging settings. Permanent entries cannot be replaced by dynamic entries.

| **Aging Time Valid Values**: X'0000 0000 0000' to X'FFFF FFFF FFFF'

| **Aging Time Default Value**: none

The resolution value specifies how often dynamic entries in the filtering database should be scanned to determine if they have expired their age limit as set by the aging timer.

| **Resolution Valid Values**: 1 to 60 seconds

| **Resolution Default Value**: 5 seconds

Example: **`set age`**

```
seconds [300]  ?  400
resolution [5] ?   6
```

bridge *bridge-address*

Sets the bridge address. This is the low-order 6-octet bridge address found in the bridge identifier. By default, the bridge-addr-value is set to the medium access control (MAC) address of the lowest-numbered port at initialization time. You can use this command to override the use of the default address and enter your own unique address.

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

Caution: In cases where a serial line interface (or tunnel) is the lowest numbered port, it is mandatory to use this command so that the bridge will have a unique address when restarted. This process is necessary because serial lines do not have their own MAC address.

At the prompt, enter the bridge address in 12-digit hexadecimal format and press **Return**.

If you enter the address in the wrong format you will receive the message `Illegal Address`. If you enter no address at the prompt you will receive the message `Zero length address supplied` and the bridge will maintain its previous value. To return the bridge address to the default value, enter an address of all zeroes.

Valid Values: 12 hexadecimal digits

Do not use dashes or colons to separate each octet. Each bridge in the network must have a unique address for the spanning tree protocol to operate properly.

Default Value: 000000000000

Example: **`set bridge`**

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

filtering *database-size*

Sets the number of entries that can be held in the bridge filtering database.

Default Value: 1024 times the number of bridge ports.

For more information, see the **list filtering** command on page 6-24.

Example: **`set filtering`**

```
database-size [2048]?
```

lf-bit-interpretation *encode-mode*

Sets the Largest Frame (LF) bit encoding interpretation if source routing is enabled in this bridge.

Example: **`set lf-bit-interpretation basic`**

Encode-mode	Entered as either **basic** or **extended**. In the basic mode only 3 bits of the routing control field are used. This is the common practice in source routing bridges that exist today. In extended mode, 6 bits of the routing control field are used to represent the maximum data unit that the bridge supports. The default value is **extended**. Extended and Basic nodes are compatible.

maximum-packet-size *port# msdu-size*

Sets the largest MAC service data unit (MSDU) size for the port, if source routing is enabled on this port. The MSDU value setting has no implication on traditionally transparent media. An MSDU value greater than the packet size configured in the router will be treated as an error.

If this parameter is not set, the default value used is the size configured as the packet size for that interface.

Valid Values: Specify an integer in the range of 16 - 65535

Default Value: packet size set for the port

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

port *block* or *disable*

Begins the port's participation in the spanning tree protocol. This is done by entering a status value of "block." This places the port in the "blocked" status as a starting point. The actual state of the port will later be determined by the spanning tree protocol as it determines its topology. Entering a status value of "disable" removes the port from participating in the spanning tree.

Example: **`set port block`**

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

protocol *bridge* or *port*

Modifies the spanning tree protocol bridge or port parameters for a new configuration, or tunes the configuration parameters to suit a specific topology.

Enter "bridge" as the option to modify bridge parameters. The bridge-related parameters that can be modified with this command are described below.

Enter **srb** or **tb** to specify whether the source routing bridge (srb) or transparent bridge (tb) spanning tree protocol parameters are to be affected.

When setting these values, make sure that the following relationships exist between the parameters or the input will be rejected:

2 X (Bridge Forward Delay - 1 second) ≥ *Bridge Maximum Age*
Bridge Maximum Age ≥ *2 X (Bridge Hello Time + 1 second)*

Example: **`set protocol bridge tb`**

```
Bridge Max-Age [20]  25
Bridge Hello Time [2]   3
Bridge Forward Delay [15] 20
Bridge Priority [32768]    1
```

Bridge Maximum Age

Maximum age (period of time) that should be used to time out spanning tree protocol related information.

When this bridging router is selected as the root bridge in a spanning tree, the value of this parameter specifies how long other active bridges are to store the configuration bridge protocol data units (BPDUs) they receive. When a BPDU reaches its maximum age limit without being replaced, the active bridges in the network discard it and assume that the root bridge has failed. A new root bridge is then selected.

Dependencies

The setting of this parameter may be affected by the setting of the Bridge Hello Time parameter. In addition, the setting of this parameter may affect the setting of the Bridge Forward Delay parameter.

Valid Values: 6 to 40 seconds

Default Value: 20 seconds

Bridge Hello Timer	Time interval between HELLO BPDUs. When this bridging router is selected as the root bridge in a spanning tree, this parameter specifies how often this bridge transmits configuration bridge protocol data units (BPDUs). BPDUs contain information about the topology of the spanning tree and reflect changes to the topology. **Dependencies** The setting of this parameter may affect the setting of the Max age parameter. **Valid Values**: 1 to 10 seconds **Default Value**: 2 seconds
Bridge Forward Delay	Time interval used before changing to another state (should this bridge become the root). When this bridging router is selected as the root bridge in a spanning tree, the value of this parameter specifies how long active ports in all bridges remain in a *listening state*. When the forward delay time expires, ports in the listening state go into the *forwarding state*. State changes occur as a result of changes in the topology of the spanning tree, such as when an active bridge fails or is shut down. The root bridge conveys this value to all bridges. This process ensures that all bridges are consistent between changes. **Dependencies** The setting of this parameter may be affected by the setting of the SRB Bridge Max Age parameter. **Valid Values**: 4 to 30 seconds **Default Value**: 15
Bridge Priority	A high-order 2-octet bridge address found in the Bridge Identifier - either the MAC address obtained from the lowest-numbered port or the address set by the **Set Bridge** command. The bridge priority indicates the chances that this bridge will become the root bridge of the spanning tree. The lower the numerical value of the bridge priority parameter, the higher the priority of the bridge and the more likely it is to be chosen. The spanning tree algorithm chooses the bridge with the lowest numerical value of this parameter to be the root bridge. **Valid Values**: 0 to 65535 **Default Value**: 32768

Enter “port” as the option to modify the spanning tree protocol port parameters. Enter the desired value at each prompt and press **Return**.

Example: `set protocol port`

```
Port Number [1] ?
Port Path-Cost (0 for default) [0] ?   1
Port Priority [128] ?   1
```

Port Number
Bridge port number; selects the port for which the path cost and port priority will be changed.

Path Cost
Cost associated with the port which is use for possible root path cost.

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

This parameter specifies the cost associated with passing frames through this port interface, should this bridging router become the root bridge. Factor this value in when determining spanning tree routes between any two stations. A value of 0 instructs the bridging router to automatically calculate a path cost for this port using its own formula.

Valid Values: 1 to 65535

Default Value: 0 (means the cost will be calculated automatically)

Port Priority
Identifies port priority for the specified port sed by the spanning tree algorithm in making comparisons for port selection (which port offers the lowest cost path to the root bridge) and blocking decisions.

Valid Values: 0 to 255

Default Value: 128

route-descriptor-limit *limit-type RD-limit-value*

Allows the user to associate a maximum Route Descriptor (RD) length for all route explorer (ARE) or spanning tree explorer (STE) frames forwarded by the bridge if source routing is enabled.

Example: `set route-descriptor-limit ARE 14`

Limit-type
Entered either as ARE or STE depending on whether the RD-limit-value is applied to all route explorer (ARE) or spanning tree explorer (STE) frames.

RD-limit-value
Specifies the maximum number of RDs that might be contained in the routing information field (RIF) of the frame type specified by the RD limit type.

The hop count for each frame is the number of bridges through which the frame has traveled so far. One RD is added to the Routing Information Field each time the frame passes through a bridge. Therefore, the number of RDs equals the number of hops. When the number of RDs (hops) exceeds the number of hops allowed by this parameter, the frame is discarded.

Valid Values: 0 to 14

Default Value: 14

super-elan-id *super-elan-identifier*

The identifier designates the super ELAN to which the bridge port is attached. Bridge ports with the same super ELAN Id comprise a single super ELAN. Super ELAN Ids have only local significance. Thus, two bridge ports in different physical systems can belong to the same super ELAN but have difference local super ELAN identifiers. Data frames are not bridged between bridge ports with different super ELAN identifiers.

Valid Values: 1 - 65535

Example: `set super-elan-id 1`

Tunnel

Use the **tunnel** command to access the Tunnel configuration prompt for a specific tunnel. Tunnel configuration commands are entered at this prompt. See "Tunnel Configuration Commands" for an explanation of each of these commands.

Syntax: tunnel *tunnel-id*

Example: **`tunnel 2`**

Once a port is configured, all other commands that need a port number as a parameter can function with this port.

VLANS

Use the **VLANS** command to access the VLAN configuration prompt. VLAN configuration commands are entered at this prompt. See "Dynamic Protocol Filtering (VLANS) Configuration Commands" on page 6-38 for an explanation of each of these commands.

Syntax: VLANS

Example: **`VLANS`**

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Tunnel Configuration Commands

This section summarizes and then explains the Tunnel configuration commands. The Tunnel configuration commands allow you to specify network parameters for specified tunnels that transmit bridging frames over IP.

Configuration commands for specifically defined tunnels are entered at the `TNL config>` prompt. This prompt is accessed by entering the **tunnel** command at the

ASRT config> prompt. Table 6-2 on page 6-35 shows the tunnel configuration commands.

Table 6-2. Tunnel Configuration Commands

Command	Function
? (Help)	Lists all of the Tunnel configuration commands, or lists the options associated with specific commands.
Add	Adds the IP address of destination bridges participating in an IP unicast or multicast addressing configuration for bridging over IP.
Delete	Deletes the IP address of a destination bridge participating in an IP unicast or multicast addressing configuration for bridging over IP.
Join	Configures the router as a member of one or more multicast groups.
Leave	Removes the router as a member of multicast groups.
List	Displays the IP addresses of end-stations participating in an IP unicast or multicast addressing configuration for bridging over IP. Also displays the size (in number of bytes) of bridging packets being routed through an IP tunnel and whether or not multicast addressing is enabled or disabled.
Set	Sets a base multicast IP address for multicast tunneling on the router.
Exit	Exits the tunnel configuration process and returns to the ASRT environment.

Tunneling and Multicast Packets

For tunnel configurations where multicast packets are involved, the source address of the multicast packets must lie on a network segment that is capable of the Internet Group Management Protocol (IGMP).

IGMP is not defined on ATM configurations. This means that when you run multicast applications on the router (for example, the MOSPF tunnel), you must ensure that one of the following conditions exists:

- The source is one of the LAN segment addresses
- The source is the internal IP address

The first condition can be ensured by using the IP **set router-id** configuration command. The second condition can be ensured by using the IP **set internal-ip-address** configuration command.

In all cases, the second option is preferred and the first should be used only if some of the routers in the network do not like host addresses (this would happen in mixed vendor networks).

? (Help)

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

Syntax: ?

Example: **?**

or

```
list ?
```

Add

Use the **add** command to add the IP address of end stations participating in a unicast or multicast IP addressing configuration.

For IP unicast addressing, the tunneling configuration requires that you supply IP addresses of destination bridges. This record will be used by the router software to convert the segment number in the routing information field (RIF) in a source routed frame to the corresponding IP address of the destination bridge. For transparent bridging frames, it identifies the other endpoint of the tunnel.

For IP multicast addressing, the tunneling configuration requires only the IP multicast address reserved for tunneling. Encapsulation uses three groups of IP multicast addresses. The first group is for sending all-routes explorer (ARE) frames, the second group for sending spanning tree explorer (STE) frames, and the third group for specifically routed frames (SRF).

Note: The bridging router software transparently differentiates between unicast and multicast addresses.

Syntax: add address *IP-address*

Valid Values: a valid IP address

Default Value: none

Example: `add address 128.185.144.37`

Delete

Use the **delete** command to delete the IP address of bridges participating in a unicast or multicast IP addressing configuration.

Syntax: delete address *IP-address*

Valid Values: a valid IP address

Default Value: none

Example: `delete address 128.185.144.37`

Join

Use the **join** command to establish the router as a member of one or more multicast groups. A tunnel group may be one of three types: peer, client, or server. The tunnel group is defined by an integer tag. A bridge can belong to only one group type for each tag. A bridge cannot belong to both *peer group 1* and *server group 1*, for example.

Syntax: join client-group *group-number*
peer-group *group-number*
server-group *group-number*

client-group *group-number*

Joins the client group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `join client-group 3`

peer-group *group-number*

Joins the peer group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `join peer-group 5`

server-group *group-number*

Joins the server group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `join server-group 7`

Leave

Use the **leave** command to remove the router as a member of multicast groups.

Syntax: leave server-group *group-number*
client-group *group-number*
peer-group *group-number*

server-group *group-number*

Leaves the server group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `leave server-group 7`

client-group *group-number*

Leaves the client group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `leave client-group 3`

peer-group *group-number*

Leaves the peer group with the given group number.

Valid Values: 1 to 64

Default Value: 0

Example: `leave peer-group 5`

List

Use the **list** tunnel command to display the IP addresses of bridges participating in an IP unicast or multicast addressing configuration for tunneling over IP. This command can also be used to display the current size of IP packets being sent through the tunnels and displays whether or not IP is enabled or disabled.

Syntax: list address
all

address

Lists the IP addresses of bridges participating in an IP unicast or multicast addressing configuration for tunneling over IP.

Example: **`list address`**

```
IP Tunnel Addresses
128.185.179.51      128.185.170.51      128.185.142.39
128.185.143.39       224.0.0.5
```

all

Lists all unicast IP addresses, configured multicast addresses, and the tunnel packet size.

Example: **`list all`**

```
IP Tunnel Addresses
128.185.179.51      128.185.170.51      128.185.142.39
128.185.143.39       224.0.0.5
Frame size for the tunnel 2120
```

Set

Use the **set** command to set the base multicast address of the router.

Syntax: set base-multicast-address

base-multicast-address

Sets the base multicast IP address for multicast tunneling.

Valid Values: any valid IP address

Default Value: none

Example: **`set base-multicast-address 224.10.0.0`**

Exit

Use the **exit** command to return to the previous ASRT prompt level.

Syntax: exit

Example: **`exit`**

Dynamic Protocol Filtering (VLANS) Configuration Commands

This section explains all of the VLAN configuration commands. These commands let you configure VLAN filtering for IP, IPX, and NetBIOS.

Configuration commands are entered at the `VLAN config>` prompt. This prompt is accessed by entering the **VLANS** command at the `ASRT config>` prompt. The following table shows the VLAN filtering configuration commands.

Table 6-3. VLAN Configuration Command Summary

Command	Function
? (Help)	Lists all of the VLAN filtering configuration commands, or lists the options associated with specific commands.
Add	Adds a VLAN filter for an IP subnet, IPX network, or NETBIOS
Change	Changes VLAN filtering parameters for an IP subnet, IPX network, or NETBIOS
Delete	Deletes the selected VLAN filter(s)
Disable	Disables VLAN filtering on the selected subnet(s)
Enable	Enables VLAN filtering on the selected subnet(s)
List	Displays all information associated with the selected VLAN filter(s)
Exit	Exits the VLAN filtering configuration process and returns to the ASRT environment

? (Help)

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

Syntax: ?

Example: ?

```
ADD
CHANGE
DELETE
DISABLE
ENABLE
LIST
EXIT
```

Add

Use the Add command to add a VLAN filter for a particular IP subnet, an IPX Network, or NetBIOS traffic.

Syntax:

add ip

ipx

netbios

Example 1: add ip

```
IP Address [0.0.0.0]? 9.2.3.4
Subnet Mask [255.0.0.0]?
Configure Specific Ports? [No]:
Age (expiration in minutes,0=infinity) [5000]? 0
Enable IP-Cut-Through from this VLAN? [Yes]:
Enable IP-Cut-Through to this VLAN? [Yes]:
Enable This Filter? [Yes]:
VLAN Name (32 chars max) []? IP 9.x.x.x
VLAN 'IP 9.x.x.x' (IP subnet 9.0.0.0) successfully added
```

If some ports should not be configured as Auto-Detect and Include, then the port can be manually configured.

Example 2: add ipx

```
Network Number (in 8-digit hex) (1 - FFFFFFFE) [1]? 2FF
Configure Specific Ports? [No] y
Configure VLAN on port 1 (Include, Exclude, or Auto-Detect) [A]?
Configure VLAN on port 2 (Include, Exclude, or Auto-Detect) [A]? e
Age (expiration in minutes,0=infinity) [5000]?
Enable IP-Cut-Through from this VLAN? [Yes]:
Enable IP-Cut-Through to this VLAN? [Yes]: n
Enable This Filter? [Yes]:
VLAN Name (32 chars max) []? IPX 2FF
VLAN 'IPX 2FF' (IPX network 0x2FF) successfully added
```

A description of each parameter follows:

IP Address

This prompt allows you to enter the IP address of the IP subnet whose traffic will be dynamically filtered to create this VLAN. This value, after the subnet mask is applied, is what will be saved and referenced in other VLAN commands.

Subnet Mask

This is the subnet mask that will be applied to the input IP Address to create the IP subnet value used to detect traffic for this VLAN.

Network Number

This prompt allows you to enter the IPX network ID number whose traffic will be dynamically filtered to create this VLAN.

Configure

Answering "No" to this prompt causes all bridge ports to be set to the default value of Auto-Detect and Include. Answering "Yes" to this prompt causes further prompting to select the desired port inclusion mode for each bridge port.

The modes are:

- Auto-Detect and Include (the default mode that requires that broadcast traffic from this subnet be received on the port before being included in the VLAN forwarding domain)
- Include Always (to always include this port in the forwarding domain regardless of received traffic)
- Exclude Always (to always exclude this port from the forwarding domain regardless of received traffic).

Age

The amount of time, in minutes, that an Auto-Detect port will remain in the forwarding state in the absence of traffic received from that port for

this VLAN. Entering a value of zero means that ports auto-detected will never expire and be removed from the forwarding domain.

Valid Values: 0 to 4 294 967 295

Default Value: 5000

Enable IP-Cut-Through Transmission Status

Answering "Yes" will allow forwarding of IP traffic from devices on this VLAN to devices on other VLANs that have IP-Cut-Through reception enabled.

Enable IP-Cut-Through Reception Status

Answering "Yes" will allow IP traffic to be forwarded to devices on this VLAN from devices on other VLANs that have IP-Cut-Through transmission enabled.

VLAN Filter Status

Answering "Yes" will enable dynamic filtering for this VLAN. Answering "No" means that no filtering will be done on traffic bound for this subnet.

VLAN Name

This prompt lets you define an optional name for this VLAN that can be used with all VLAN commands.

This name must be unique among all VLANs for all protocols and consists of up to 32 characters.

Change

Use the change command to change the configuration parameters associated with a particular VLAN. The VLAN to change can be chosen by explicitly specifying the subnet or by selecting the VLAN from a list with the *by-name* option. This command invokes the same prompts used with the add command. The current parameter values will be displayed as the default and can be maintained by simply pressing Return.

Syntax:

change by-name

ip subnet address

ipx network number

netbios

Example: change ip

```
IP Address [9.0.0.0]?
Configure Specific Ports? [No]:
Age (expiration in minutes,0=infinity) [0]? 300
Enable IP-Cut-Through from this VLAN? [Yes]:
Enable IP-Cut-Through to this VLAN? [Yes]:
Enable This Filter? [Yes]:
VLAN Name (32 chars max) [IP 9.x.x.x]?
VLAN 'IP 9.x.x.x' (IP subnet 9.0.0.0) successfully changed
```

Delete

Use the delete command to delete a particular VLAN filter, all VLAN filters for a particular protocol, or all defined VLAN filters. If deleting a single filter, the VLAN to be deleted can be chosen by specifying the subnet or by selecting the VLAN from a list using the *by-name* option.

Syntax:

```
delete    by-name
          ip        all
          ip        subnet   subnet address
          ipx       all
          ipx       network  network number
          netbios
          all
```

Example 1: del ip subnet 9.0.0.0

```
VLAN 'IP 9.x.x.x' (IP subnet 9.0.0.0) deleted
```

Example 2: del ipx all

```
Are you sure you want to delete ALL IPX VLANS? [No]: y
All IPX VLANS deleted
```

Disable

Use the disable command to disable a particular VLAN filter, all VLAN filters for a particular protocol, or all defined VLAN filters. If disabling a single filter, the VLAN to be disabled can be chosen by specifying the subnet or by selecting the VLAN from a list using the *by-name* option.

Syntax:

```
disable   by-name
          ip   all
          ip   subnet   subnet address
          ipx  all
          ipx  network  network number
          netbios
          all
```

Example: disable ip subnet 220.5.3.0

```
VLAN 'Building #4' (IP subnet 220.5.3.0) now disabled
```

Enable

Use the enable command to enable a particular VLAN filter, all VLAN filters for a particular protocol, or all defined VLAN filters. If enabling a single filter, the VLAN to be enabled can be chosen by specifying the subnet or by selecting the VLAN from a list using the *by-name* option.

Syntax:

```
enable    by-name
          ip   all
          ip   subnet   subnet address
          ipx  all
          ipx  network  network number
          netbios
          all
```

Example: enable by-name

```
      Choice of VLAN:
          Subnet      VLAN Name
          ======      =========
      (1) 9.0.0.0     IP 9.x.x.x
      (2) 220.5.3.0   Building #4
      (3) 0x2FF       Token Ring A
      (4) 0x3FF       Token Ring B
   Enter Selection [1]? 3
   VLAN 'Token Ring A' (IPX Network 0x2FF) now enabled
```

List

Use the list command to list the configuration information about a particular VLAN filter, all VLAN filters for a particular protocol, or all defined VLAN filters. If listing a single filter, the VLAN to be listed can be chosen by specifying the subnet or by selecting the VLAN from a list using the *by-name* option.

Syntax:

```
list      by-name
          ip   all
          ip   subnet   subnet address
          ipx  all
          ipx  network  network number
          netbios
          all
```

Example 1: list ip subnet 9.0.0.0

```
Subnet Address              = 9.0.0.0
Subnet Mask                 = 255.0.0.0
Bridge Port 1 (Interface 0) = Auto-Detect and Include
Bridge Port 2 (Interface 1) = Always Exclude
Age (expiration in minutes) = 300
IP-Cut-Through Status:
   Transmit From This VLAN  = Enabled
   Reception By This VLAN   = Enabled
VLAN Filter State           = Enabled
VLAN Name                   = IP 9.x.x.x
```

Example 2: list ipx all

```
----------- IPX VLANS -------------------------------
IPX Network Number          = 0x2FF
Bridge Port 1 (Interface 0) = Auto-Detect and Include
Bridge Port 2 (Interface 1) = Always Exclude
Age (expiration in minutes) = Never Expires
IP-Cut-Through Status:
   Transmit From This VLAN  = Enabled
   Reception By This VLAN   = Disabled
VLAN Filter State           = Enabled
VLAN Name                   = Token Ring A
+++++++++++++++++++++++++++++++++++++++++++++++++++++++
IPX Network Number          = 0x3FF
Bridge Port 1 (Interface 0) = Auto-Detect and Include
Bridge Port 2 (Interface 1) = Auto-Detect and Include
Age (expiration in minutes) = 5000
IP-Cut-Through Status:
   Transmit From This VLAN  = Enabled
   Reception By This VLAN   = Enabled
VLAN Filter State           = Disabled
VLAN Name                   = Token Ring B
```

Exit

Use the exit command to return to the previous prompt level.

Syntax: exit

Example: exit

Bridging Broadcast Manager Configuration Commands

Bridging Broadcast Manager (BBCM) can transform many broadcast frames into unicast frames, thus lessening their effects on network performance. BBCM snoops on packets to learn bindings between layer 3 and layer 2 network addresses. Future broadcast packets to any learned layer 3 address can be transformed into unicast packets and forwarded by the bridge as any other unicast packet. If the transformed unicast address is in the bridge database, then the packet can be forwarded to its destination segment. The packet will not cause an interruption at every station, nor will it utilize bandwidth on every network segment. Refer to the LAN Emulation overview chapter in *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1* for additional discussion of Bridging Broadcast Manager.

This section explains all of the BBCM (bridging broadcast-manager) configuration commands. These commands let you configure BBCM for IP. See Chapter 8, "Using, Configuring, and Monitoring NetBIOS" on page 8-1 for configuring NetBIOS Filtering and Name Caching.

Configuration commands are entered at the `IP B-BCM config>` prompt. This prompt is accessed by entering the broadcast-manager command at the `ASRT config>` prompt. Table 6-4 shows the BBCM configuration commands.

Table 6-4. BBCM Configuration Commands

Command	Function
? (Help)	Lists all of the BBCM configuration commands, or lists the options associated with specific commands.
Enable	Enables bridging broadcast-manager.
Disable	Disable bridging broadcast-manager.
List	Displays general information concerning BBCM configuration.
Set *cache age timeout*	Sets the bridging broadcast-manager cache age timeout. Cache entries that are not refreshed in this amount of time are aged out.
Exit	Exits the BBCM configuration process and returns to the ASRT environment.

? (Help)

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

Syntax: ?

Example: ?

```
IP B-BCM config>?
DISABLE ip b-bcm
ENABLE ip b-bcm
LIST configuration
SET cache age timeout
EXIT
```

Enable

Use the **enable** command to enable BBCM.

Syntax: enable

Example: enable

```
IP B-BCM config>enable
IP Bridge Broadcast Manager is   ENABLED
```

Disable

Use the **disable** command to disable BBCM.

Syntax: disable

Example: disable

```
IP B-BCM config>disable
IP Bridge Broadcast Manager is   DISABLED
```

List

Use the **list** command to list general information about BBCM.

Syntax: list

Example: list

```
IP B-BCM config>list
IP Bridge Broadcast Manager is   ENABLED
IP B-BCM cache age timeout is    9 minutes
```

Set

Use the **Set** command to set the cache age timeout for the BBCM cache.

Syntax: set

Example: set

```
IP B-BCM config>set
IP B-BCM cache age timeout in minutes (system default=5) [9]? 10
IP B-BCM cache age timeout set to 10 minutes
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: exit

```
IP B-BCM config>exit
ASRT config>exit
Config>
```

Sample Super ELAN Configuration

```
The Standalone Configuration Process.  You are here because
No network devices configured.

Config (only)>add dev atm 1    1
Adding CHARM ATM Adapter device in slot 1  port 1 as interface #0
Use "net 0" to configure CHARM ATM Adapter parameters
Config (only)>net 0
ATM user configuration
ATM Config>int    2
ATM interface configuration
ATM Interface Config>add esi    3
ESI in 00.00.00.00.00.00 form []? 333333333333
ATM Interface Config>exit
ATM Config>le-s    4
LAN Emulation Services user configuration
LE Services config>les
ELAN Name (ELANxx) []? elan1
LES-BUS configuration
LES-BUS config for ELAN 'elan1'>add    5
Turn on Standard Event Logging for LES [yes]
Select ELAN type
        (1) Token Ring
        (2) Ethernet

Enter Selection:  [1]? 2
Select ESI
        (1) Use burned in ESI
        (2) 33.33.33.33.33.33

Enter Selection:  [1]? 2    6


Selector x00 is generally reserved for use by the LECS,
Selector x01 is generally reserved for use by the LECS Interface.

Enter selector (in hex) [2]?
Selection "Add LES-BUS" Complete
LES-BUS config for ELAN 'elan1'>exit
LE Services config>exit
ATM Config>le-s
LAN Emulation Services user configuration
LE Services config>les elan2
LES-BUS configuration
```

```
LES-BUS config for ELAN 'elan2'>add    7
Select ELAN type
        (1) Token Ring
        (2) Ethernet

Enter Selection:  [1]? 2
Select ESI
        (1) Use burned in ESI
        (2) 33.33.33.33.33.33

Enter Selection:  [1]? 2


Selector x00 is generally reserved for use by the LECS,
Selector x01 is generally reserved for use by the LECS Interface.

Enter selector (in hex) [3]?
Selection "Add LES-BUS" Complete
LES-BUS config for ELAN 'elan2'>exit
LE Services config>exit
ATM Config>le-c    8
ATM LAN Emulation Clients configuration
LE Client config>add eth forum    9
Added Emulated LAN as interface 1
LE Client config>add eth forum    10
Added Emulated LAN as interface 2
LE Client config>add eth forum    11
Added Emulated LAN as interface 3
LE Client config>config 1
ATM LAN Emulation Client configuration
Ethernet Forum Compliant LEC Config>set les    12
LES ATM address in 00.00.00.00.00.00:... form []? 399999999999999000099990101
33333333333302
Ethernet Forum Compliant LEC Config>set mac    13
Use adapter address for MAC? [Yes]: n
MAC address [00.00.00.00.00.00]? 100000000001
Ethernet Forum Compliant LEC Config>exit
LE Client config>config 2
ATM LAN Emulation Client configuration
Ethernet Forum Compliant LEC Config>set les    14
LES ATM address in 00.00.00.00.00.00:... form []? 399999999999999000099990101
33333333333303
Ethernet Forum Compliant LEC Config>set mac
Use adapter address for MAC? [Yes]: n
MAC address [00.00.00.00.00.00]? 100000000002
Ethernet Forum Compliant LEC Config>exit
```

```
LE Client config>config 3
ATM LAN Emulation Client configuration
Ethernet Forum Compliant LEC Config>set les    15
LES ATM address in 00.00.00.00.00.00:... form []? 399999999999990000999901013
33333333333302
Ethernet Forum Compliant LEC Config>set mac
Use adapter address for MAC? [Yes]: n
MAC address [00.00.00.00.00.00]? 100000000003
Ethernet Forum Compliant LEC Config>exit
LE Client config>exit
ATM Config>exit
Config (only)>p ip    16
Internet protocol user configuration
IP config>add address    17
Which net is this address for [0]? 3
New address [0.0.0.0]? 1.1.1.1
Address mask [255.0.0.0]? 255.255.255.0
IP config>add address    18
Which net is this address for [0]? 3
New address [0.0.0.0]? 2.2.2.1
Address mask [255.0.0.0]? 255.255.255.0
IP config>list address
IP addresses for each interface:
   intf  0                                        IP disabled on this interface
   intf  3   1.1.1.1          255.255.255.0      Local wire broadcast, fill 1
             2.2.2.1          255.255.255.0      Local wire broadcast, fill 1
IP config>exit
Config (only)>p asrt    19
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>enable bridge    20
ASRT config>del port 3    21
```

```
ASRT config>list bridge

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

Bridge:                 Enabled             Bridge Behavior: STB
                    +----------------------------+
-------------------| SOURCE ROUTING INFORMATION |-------------------------------
                    +----------------------------+
Bridge Number:          N/A                 Segments:          0
Max ARE Hop Cnt:        00                  Max STE Hop cnt:   00
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
                    +------------------+
-------------------| PORT INFORMATION |-----------------------------------------
                    +------------------+
Number of ports added: 2
Port:   1      Interface:       1      Behavior:    STB Only   STP:  Enabled
Port:   2      Interface:       2      Behavior:    STB Only   STP:  Enabled

ASRT config>en super    22
Port Number [1]?
Super ELAN ID [0]? 22
Port priority of Super ELAN enabled ports should be less than 128.
Enter New Port Priority [64]?
Bridge priority with Super ELAN enabled ports should be less than 32768.
Enter New Bridge Priority [16384]?
ASRT config>en super
Port Number [1]? 2
Super ELAN ID [22]?
Port priority of Super ELAN enabled ports should be less than 128.
Enter New Port Priority [64]?
```

```
ASRT config>list port
Port ID (dec)    : 64:01, (hex): 40-01
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface  : 1
Path Cost        : 0
Super ELAN bridging: Enabled       Super ELAN ID: 22
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Port ID (dec)    : 64:02, (hex): 40-02
Port State       : Enabled
STP Participation: Enabled
Port Supports    : Transparent Bridging Only
Assoc Interface  : 2
Path Cost        : 0
Super ELAN bridging: Enabled       Super ELAN ID: 22
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ASRT config>broadcast    23
Enter Bridge Broadcast Manager Protocol:  IP or NetBIOS  [IP]?

IP Bridge Broadcast Manager User Configuration

IP B-BCM config>enable    24
IP Bridge Broadcast Manager is   ENABLED
IP B-BCM config>list
IP Bridge Broadcast Manager is   ENABLED
IP B-BCM cache age timeout is    5 minutes
IP B-BCM config>exit
```

```
ASRT config>broadcast
Enter Bridge Broadcast Manager Protocol:  IP or NetBIOS  [IP]? net

NetBIOS Support User Configuration

NetBIOS config>enable dup

Duplicate frame filtering is            ON

NetBIOS config>enable route    25

Route caching is                        ON

NetBIOS config>list general

Bridge-only Information:

Bridge duplicate filtering is           ON
Bridge duplicate frame filter t/o      1.5 seconds

DLS-Bridge Common Information:

Route caching is                        ON
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
NetBIOS config>exit
```

```
ASRT config>vlans    26

VLAN filter configuration
VLAN config>add ip    27
IP Address [0.0.0.0]? 1.1.1.1
Subnet Mask [255.0.0.0]? 255.255.255.0
Configure Specific Ports? [No]:
Age (expiration in minutes,0=infinity) [5000]?
Enable IP-Cut-Through from this VLAN? [Yes]:
Enable IP-Cut-Through to this VLAN? [Yes]:
Enable This Filter? [Yes]:
VLAN Name (32 chars max) []? Dept A
VLAN 'Dept A' (IP Subnet 1.1.1.0) successfully added
VLAN config>add ip    28
IP Address [0.0.0.0]? 2.2.2.1
Subnet Mask [255.0.0.0]? 255.255.255.0
Configure Specific Ports? [No]:
Age (expiration in minutes,0=infinity) [5000]?
Enable IP-Cut-Through from this VLAN? [Yes]:
Enable IP-Cut-Through to this VLAN? [Yes]:
Enable This Filter? [Yes]:
VLAN Name (32 chars max) []? Dept B
VLAN 'Dept B' (IP Subnet 2.2.2.0) successfully added
VLAN config>add net    29
Configure Specific Ports? [No]:
Age (expiration in minutes,0=infinity) [5000]?
Enable This Filter? [Yes]:
VLAN Name (32 chars max) []? Microsoft NT Users
VLAN 'Microsoft NT Users' (NetBIOS) successfully added
```

```
VLAN config>list all

   --------------------- IP VLANS ----------------------

   Subnet Address              = 1.1.1.0
   Subnet Mask                 = 255.255.255.0
   Bridge Port 1 (Interface 1) = Auto-Detect and Include
   Bridge Port 2 (Interface 2) = Auto-Detect and Include
   Age (expiration in minutes) = 5000
   IP-Cut-Through Status:
      Transmit From This VLAN  = Enabled
      Reception By This VLAN   = Enabled
   VLAN Filter State           = Enabled
   VLAN Name                   = Dept A
   ++++++++++++++++++++++++++++++++++++++++++++++++++++
   Subnet Address              = 2.2.2.0
   Subnet Mask                 = 255.255.255.0
   Bridge Port 1 (Interface 1) = Auto-Detect and Include
   Bridge Port 2 (Interface 2) = Auto-Detect and Include
   Age (expiration in minutes) = 5000
   IP-Cut-Through Status:
      Transmit From This VLAN  = Enabled
      Reception By This VLAN   = Enabled
   VLAN Filter State           = Enabled
   VLAN Name                   = Dept B

   --------------------- IPX VLANS ---------------------

   ------------------- NetBIOS VLAN --------------------

   Bridge Port 1 (Interface 1) = Auto-Detect and Include
   Bridge Port 2 (Interface 2) = Auto-Detect and Include
   Age (expiration in minutes) = 5000
   VLAN Filter State           = Enabled
   VLAN Name                   = Microsoft NT Users
VLAN config>exit
ASRT config>exit
Config (only)>write
```

Notes:

1. Add the physical ATM device
2. Configure the ATM interface
3. Add a locally administered ESI for the ATM interface
4. Begin configuration of the LAN Emulation Service
5. Add the definition for an Ethernet LAN Emulation Server
6. Select locally administered ESI for 'elan1'
7. Add the definition for a second Ethernet LAN Emulation Server
8. Begin configuration of LAN Emulation Clients (LECs)
9. Add an Ethernet LEC on interface 1 for a bridge port on 'elan1'

10 Add an Ethernet LEC on interface 2 for a bridge port on 'elan2'

11 Add an Ethernet LEC on interface 3 for a router port on 'elan1' (a protocol cannot be both bridged and routed over the same LEC)

12 Assign LEC 1 to the LES for 'elan1'

13 Give each LEC a unique locally administered MAC address

14 Assign LEC 2 to the LES for 'elan2'

15 Assign LEC 3 to the LES for 'elan1'

16 Begin IP protocol configuration

17 Add an IP address for LEC 3 on the VLAN for subnet 1.1.1.0

18 Add an IP address for LEC 3 on the VLAN for subnet 2.2.2.0

19 Begin bridge configuration

20 Enable bridging, which by default add ports for all interfaces

21 Disable bridging on port 3, because it will be routing IP

22 Enable SuperELAN bridging

23 Begin configuration of the Bridging Broadcast Manager

24 Enable Bridging Broadcast Manager for IP

25 Enable route caching (not applicable to Ethernet, but shown here because it is valuable for Token Ring bridging)

26 Begin configuration of VLANs (Dynamic Protocol Filtering)

27 Begin definition of a VLAN for IP subnet 1.1.1.0

28 Begin definition of a VLAN for IP subnet 2.2.2.0

29 Begin definition of a VLAN for NetBIOS

Chapter 7. Monitoring Bridging

This chapter describes how to monitor the ASRT (Adaptive Source Routing Transparent) Bridge and how to use the ASRT console commands. Console commands for the bridging router's tunnel and NetBIOS features are also included as part of the general ASRT console command set. The chapter includes the following sections:

- "Accessing the ASRT Console Environment"
- "ASRT Console Commands"

Accessing the ASRT Console Environment

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

```
+protocol asrt
ASRT>
```

ASRT Console Commands

This section summarizes and then explains the ASRT console commands. These commands allow you to view and modify parameters from the active console. Information you modify with the console commands is reset to the SRAM configuration when you restart the bridging router.

You can use these commands to temporarily modify the configuration without losing configuration information in the bridge memory. The `ASRT>` prompt is displayed for all ASRT console commands.

Monitoring commands for NetBIOS are entered at the `NetBIOS>` console prompt. The NetBIOS prompt is a subset of the major ASRT commands and is accessed by entering the ASRT **netbios** command explained later in this chapter.

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

Monitoring and dynamic reconfiguration VLANS commands are entered at the `VLAN>` console prompt. The `VLAN>` command is accessed by entering the **VLANS** command explained later in this chapter.

Note: For commands requiring you to enter MAC Addresses, the addresses can be entered in the following formats:

IEEE 802 canonical bit order 00-00-00-12-34-56
IEEE 802 canonical bit order (shorthand format) 000000123456
IBM Token-Ring native bit order (noncanonical) 00:00:00:12:34:56

Table 7-1 shows the ASRT console commands.

Table 7-1 (Page 1 of 2). ASRT Console Commands Summary

Command	Function
? (Help)	Lists all the ASRT console commands or lists the options associated with specific commands.

Table 7-1 (Page 2 of 2). ASRT Console Commands Summary

Command	Function
Add	Adds permanent (static) address entries to the bridging router's permanent database.
Broadcast	Allows you to access the Bridging Broadcast Manager console prompt for entering specific BBCM console commands
Cache	Displays cache entries for a specified port.
Delete	Deletes MAC addresses entries from the bridging router database.
Flip	Flips MAC address from canonical to 802.5 (noncanonical or IBM) bit order.
List	Displays information about the complete bridge configuration or about selected configuration options.
NetBIOS	Displays the NetBIOS monitoring prompt.
VLANS	Displays the VLAN console prompt.
Exit	Exits the ASRT console process and returns to the GWCON environment.

? (Help)

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

Syntax: ?

Example: **?**

or

list ?

Add

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

Syntax: add destination-address-filter
static-entry . . .

destination-address-filter *mac_address*

Adds a destination address filter to the bridging router's permanent database. Enter the command followed by the MAC address of the entry.

Example: **add destination-address-filter**

```
Destination MAC address [00-00-00-00-00-00]?
```

static-entry *mac_address input_port [output_ports]*

Adds static address entries to the bridging router's permanent database. Enter the command followed by the MAC address of the static entry and the input port number (an optional output port number may also be entered). To create a static entry with multiple port maps (1 per input port), use this command several times.

Example: add static-entry

```
MAC address [00-00-00-00-00-00]?  400000012345
Input port, 0 for all [0]? 2
Output port, 0 for none [0]? 3
Output port, 0 to end [0]?
```

Broadcast

Use the **broadcast** manager command to display the BBCM configuration and to allow you access to the console functions of BBCM.

Syntax: broadcast

Example 1

```
*t 5

CGW Operator Console

+p asrt
ASRT>broadcast
Enter Bridge Broadcast Manager Protocol:  IP or NetBIOS  [IP]? ip

IP Bridge Broadcast Manager User Console
IP B-BCM>
```

Example 2

```
ASRT>broadcast
Enter Bridge Broadcast Manager Protocol:  IP or NetBIOS  [IP]? netbios

NetBIOS Support User Console

NetBIOS>
NetBIOS>exit
ASRT>exit
+
```

Table 7-2 shows the Broadcast console commands.

Table 7-2. Broadcast Console Commands Summary

Command	Function	
?		Displays information about BBCM configuration options
Clear	**cache**	Clears the BBCM cache
	statistics	Clears the BBCM statistics
Disable		Disables BBCM
Enable		Enables BBCM
List	**cache**	Lists the contents of BBCM's cache
	general	Lists general information about BBCM
	statistics	Lists BBCM statistics
Set		Sets the cache aging timeout value for BBCM
Exit		Returns to the ASRT> prompt

? (Help)

Use the **? (HELP)** command to list the commands that are available from the current prompt level.

Help Syntax: ? (help)

Example

```
IP B-BCM>?
CLEAR
DISABLE ip b-bcm
ENABLE ip b-bcm
LIST
SET cache age timeout
EXIT
```

Clear

Use the **clear** command to clear the BBCM cache or clear BBCM statistics.

Clear Syntax: clear cache

clear statistics

Example

```
IP B-BCM>clear ?
CACHE entries
STATISTICS
IP B-BCM>clear cache
Clearing IP B-BCM cache.
IP B-BCM>clear statistics
```

Disable

Use the **disable** command to disable BBCM.

Disable Syntax: disable

Example disable

```
IP B-BCM>disable
IP Bridge Broadcast Manager is   INACTIVE
```

Enable

Use the **enable** command to enable BBCM.

Enable Syntax: enable

Example enable

```
IP B-BCM>enable
IP Bridge Broadcast Manager is   ACTIVE
```

List

Use the **list** command to list general information about BBCM.

List Syntax: list cache

list general

list statistics

Example

```
IP B-BCM>list cache
No entries found
IP B-BCM>list general
IP Bridge Broadcast Manager is   ENABLED
IP B-BCM cache age timeout is    9 minutes

Current Status:
IP Bridge Broadcast Manager is   ACTIVE
# of IP Addresses in cache is    0

IP B-BCM>list ?
CACHE entries
GENERAL information
STATISTICS
```

Set

Use the **set** command to set the cache age timeout for the BBCM cache.

Set Syntax: set

Example

```
IP B-BCM>set
IP B-BCM cache age timeout in minutes (system default=5) [9]? 10
IP B-BCM cache age timeout set to 10 minutes
```

Exit

Use the **exit** command to return to the previous prompt level.

Exit Syntax: exit

Example

```
IP B-BCM>exit
```

Cache

Use the **cache** command to display the contents of a selected bridging-port routing cache. If the port does not possess a cache you will see the message `Port X does not have a cache`.

Syntax: cache *port#*

Example: **`cache`**

```
Port number [1]? 3

MAC Address    MC*   Entry Type       Age  Port(s)

00-00-93-00-C0-D0   PERMANENT          0  3 (TKR/1)
00-00-00-11-22-33   STATIC             0  3 (TKR/1)
```

MAC Address	6-byte MAC address of the entry.
Entry Type	Specifies one of the following address entry types:
	Reserved - entries reserved by the IEEE802.1D Standard.
	Registered - entries consist of unicast addresses belonging to proprietary communications hardware attached to the box or multicast addresses enabled by protocol forwarders.
	Permanent - entries entered by the user in the configuration process which survive power on/offs or system resets.
	Static - entries entered by the user in the console process which do not survive power on/offs or system resets and are not effected by the aging timer.
	Dynamic - entries "learned" by the bridge "dynamically" which do not survive power on/offs or system resets and which have an "age" associated with the entry.
	Free - locations in database that are free to be filled by address entries.
	Unknown - entry types unknown to the bridge. May be possible bugs and/or illegal addresses.
Age	Age in seconds of each dynamic entry. Age is decremented at each resolution intervals.
Port(s)	Specifies the port number associated with that entry and displays the interface name (this will always be that of the interface having the cache).

Delete

Use the **delete** command to delete station (including MAC) address entries from the router's permanent database.

Syntax: delete *MAC-address*

Example: **`delete 00-00-93-10-04-15`**

Flip

Use the **flip** command to view specific MAC addresses in the canonical and noncanonical format by "flipping" the address bit order. This command is useful for translating IEEE 802.5 addresses in their typical noncanonical format to the canonical format universally used by the bridge console and ELS (and vice versa).

Syntax: flip *MAC-address*

Example: **`flip`**

```
MAC address [00-00-00-00-00-00]? 00-00-00-33-44-55
IEEE 802 canonical bit order:    00-00-00-33-44-55
IBM Token-Ring native bit order: 00:00:00:CC:22:AA
```

List

Use the **list** command to display information about the bridging router configuration or to display information about selected configuration or bridging options.

Syntax: list
- adaptive-bridge . . .
- bridge . . .
- conversion . . .
- database . . .
- filtering . . .
- port
- super-elan-bridge cache
- source-routing . . .
- spanning-tree-protocol . . .
- transparent . . .
- tunnel . . .

adaptive-bridge *datagroup-option [sub-option]*

Lists all general information regarding the SR-TB bridge which converts between types of bridging. There are a number of general datagroup options which may be displayed under **list adaptive-bridge**. These include the following:

- Config - Displays general information regarding the SR-TB bridge.
- Counters - Displays all SR-TB bridge counters.
- Database - Displays contents of the SR-TB bridge RIF database.

The following examples illustrate each of the **adaptive-bridge** display options.

Example: **`list adaptive-bridge config`**

```
   Adaptive bridge:            Enabled
   Translation database size:  0
   Aging time:                 320 seconds
   Aging granularity           5 seconds

Port  Segment  Interface   State     MTU
   1  001      TKR/1       Enabled   2052
   -  001      Adaptive    Enabled   1470
```

Conversion bridge	Shows the current state of the SR-TB conversion bridge. This value is displayed as either Enabled or Disabled.
Translation database size	Displays the current size of the SR-TB database, which contains MAC addresses and associated RIFs for the source-routing domain.
Aging time	Displays the aging timer setting in seconds. All SR-TB RIF database entries which exceed this time limit are discarded.
Aging granularity	Displays how often entries are scanned to look for expiration according to the aging timer.
Port	Displays the number of a port associated with conversion bridging.
Segment	Displays the source routing segment number assigned to the port associated with conversion bridging.
Interface	Identifies the device connected to a conversion bridge network segment and the VPI/VCI if an ATM port.
State	Indicates the current state of the conversion bridge port.
MTU	Specifies the maximum frame size (from the end of the RIF to the beginning of the FCS) that the conversion bridge can transmit and receive.

Example: `list adaptive-bridge counters`

```
Hash collision count: 28
Adaptive. database overflow  count: 0
```

Hash Collision Count	Displays number of addresses that were stored (hashed) to the same location in the hash table. This number is accumulative and reflects the total number of hash collision incidents that occurred. Increases in this number may indicate a potential table size problem.
Adaptive Database Overflow	Displays the number of times that an address was overwritten as the conversion database table ran out of table space.

The *database* option of the **list adaptive-bridge** command lets you list select certain portions of the adaptive bridge RIF database to display. This is due to the potential size of the database. The display options include the following:

- Address - Displays the conversion bridge database related to that specific MAC address
- All - Displays the entire database.
- Port - Displays all conversion bridge entries a specific port.
- Segment - Displays all conversion bridge entries associated with the port having the specified segment number.

The following examples illustrate each of the **list adaptive-bridge database** command options.

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

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

Example: **`list adaptive-bridge database all`**

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

Example: **`list adaptive-bridge database segment`** ***segment#***

Each entry is displayed on two lines followed by a blank line. The following information is displayed for each entry:

Canonical address	Lists the MAC address of the node corresponding to this entry. This is displayed in IEEE 802 canonical (hexadecimal) format.
Interface	Displays the name of the network interface that learned this entry.
Port	Displays the port number of the port that learned this address entry.
Seg	Displays the number of the segment that learned this address.
Age	Displays the entry age in seconds.
RIF Type	Displays the RIF type as SRF, STE, or ARE.
RIF Direction	Displays the RIF direction as Forward or Reverse.
RIF Length	Displays the RIF length in bytes.
RIF LF	Displays the largest frame value encoded in the RIF.
IBM MAC Address	Shows the MAC address of the node corresponding to this entry. This is displayed in the "IBM" noncanonical bit order as typically labeled on 802.5 interfaces and used by the IP/ARP, IPX, and NetBIOS protocols.
RIF	Displays the Routing Information Field learned from this node.

bridge

Lists all general information regarding the bridge router configuration.

Example: `list bridge`

```
Bridge ID (prio/add):  32768/10-00-5A-63-01-00
Bridge state:          Enabled
UB-Encapsulation:      Disabled
Bridge type:           STB
Bridge capability:     ASRT
Number of ports:       2
STP Participation:     IEEE802.1d

                                                    Maximum
Port  Interface      State  MAC Address         Modes   MSDU   Segment

   1  Eth/1           Up   10-00-5A-63-01-00       T    1514
   2  AT/0:0:48       Down 00-00-00-00-00-00 SR    0    121   RD


SR bridge number:      7
SR virtual segment:    001
Adaptive segment:      000
```

Bridge ID — Unique ID used by the spanning tree algorithm in determining the spanning tree. Each bridge in the network is assigned a unique bridge identifier. The bridge priority is displayed in decimal followed by the hex address.

Bridge State — Indicates whether bridging is enabled or disabled.

Bridge Type — Displays the configured bridge type. This is displayed as NONE, SRB, TB, SRT, ADAPT, A/SRB, A/TB, or ASRT.

Number of Ports — Displays the number of ports configured for that bridge.

Port — Specifies a user defined number assigned to an interface by the Add Port command.

Interface — Identifies devices connected to a network segment through the bridge.

State — Indicates the current state of the port. This is displayed as UP or DOWN.

MAC address — Displays the MAC address associated with that port in canonical bit order.

Modes — Displays the bridging mode for that port. T indicates transparent bridging. SR indicates source routing. A indicates adaptive bridging.

MSDU — Specifies the maximum frame (data unit) size (including the MAC header but not the FCS field) the source routing bridge can transmit and receive on this interface.

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

SR bridge number — Displays the user assigned source routing bridge number.

SR virtual segment — Displays the source routing bridge virtual segment number show (if any).

Adaptive segment — Displays the number of the segment which is used in the source routing domain to route to the transparent domain (via conversion).

conversion *datagroup-option*

Displays general information about the bridge's rules for converting frame formats based on the frame type. There are a number of general datagroups which may be displayed under the **list conversion** command. These include the following:

- All - Displays all rules.
- Ethertype - Displays rules for all Ethernet types or for a specific Ethernet type.
- SAP - Displays rules for all SAP protocol identifiers or a specific 802.2 SAP type.
- SNAP - Displays rules for all SNAP protocol identifiers or a specific 802.2 SNAP type.

The following examples break down each of the list conversion display options.

Example: `list conversion all`

Example: `list conversion ethertype`

```
Ethernet type (in hexadecimal), 0 for all [0]?
```

Example: `list conversion SAP`

```
SAP (in hexadecimal), 100 for all [100]?
```

Example: `list conversion SNAP`

```
SNAP Protocol ID, return for all [00-00-00-00-00]?
```

database *datagroup-option*

Lists the contents of transparent filtering databases. There are a number of datagroups which can be chosen to be displayed under the list database command. These include the following:

- All - Displays the entire transparent bridging database.
- Dynamic - Displays all dynamic (learned) address database entries.
- Local - Displays all local (reserved) address database entries.
- Permanent - Displays all permanent address database entries.
- Port - Displays address entries for a specific port.
- Range - Displays a range of database entries from the total transparent bridging filtering address database. A starting and ending MAC address is given to define the range. All entries falling within this range will be displayed.
- Static - Displays static entries from the address database.

The following examples break down the list database command options. The first example also shows the related output.

Example: `list database all`

```
MAC Address    MC*  Entry Type       Age  Port(s)
00-00-00-00-AA-AA   Dynamic          295  4 (Eth/2)
00-00-00-12-34-56   Perm/Source filter    2 (TKR/1)  ->  3-4
                                         1-2
00-00-00-22-33-44   Permanent            1-2
                                         1-2
00-00-00-33-44-55   Perm Dest filter     All
00-00-00-55-66-77   Perm/Source filter   1-2,4

00-00-93-10-04-15   Registered           1 (Eth/1)
00-00-93-10-E4-F9   Dynamic          300 1 (Eth/1)
00-00-93-90-04-A6   Dynamic          300 1 (Eth/1)
00-00-A7-10-68-28   Dynamic          270 1 (Eth/1)
01-80-C2-00-00-00*  Registered           1,3
01-80-C2-00-00-01*  Reserved             All
01-80-C2-00-00-02*  Reserved             All
01-80-C2-00-00-03*  Reserved             All
01-80-C2-00-00-0D*  Reserved             All
01-80-C2-00-00-0E*  Reserved             All
01-80-C2-00-00-0F*  Reserved             All
03-00-00-00-80-00*  Reserved             All
08-00-17-00-35-F9   Dynamic          300 1 (Eth/1)
08-00-17-00-4D-DA   Dynamic          300 1 (Eth/1)
```

Note: The following fields are displayed for all of the **list database** command options.

MAC Address — Specifies the address entry in 12-digit hex format (canonical bit order).

*MC** — An asterisk following an address entry indicates that the entry has been flagged as a multicast address.

Entry Type — Specifies one of the following types:

Reserved
Entries reserved by the IEEE802.1D standard.

Registered
Entries consist of unicast addresses belonging to interfaces participating in the bridge or multicast addresses enabled by protocol forwarders

Permanent
Entries entered by the user in the configuration process which survive power on/offs or system resets

Static
Entries entered by the user in the console process which do not survive power on/offs or system resets and are ageless.

Dynamic
Entries “learned” by the bridge “dynamically” which do not survive power on/offs or system resets and which have an “age” associated with the entry

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

Unknown
Unknown entry type. May indicate a software bug. Report the hex entry type to Customer Service.

Age — Refers to the age (in seconds) of each dynamic entry. Age is decremented at each resolution interval.

Port(s) — Specifies the outgoing port number(s) for that entry. Device type is also listed for single port entries. If dynamic entry on IP tunnel, the port will be “5” for IP tunnel.

Example: `list database dynamic`

Example: `list database local`

```
MAC Address    MC*  Entry Type       Age  Port(s)

00-00-93-B8-00-48   Registered            1 (TKR/1)
01-80-C2-00-00-00*  Registered            1
03-00-02-00-00-00*  Registered            1
ASRT>
```

Example: `list database permanent`

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

Example: `list database static`

Example: `list database range`

```
First MAC address [00-00-00-00-00-00]? 00-00-93-00-C0-D0
Last MAC address [FF-FF-FF-FF-FF-FF]? 01-80-C2-00-00-00

MAC Address    MC*  Entry Type       Age  Port(s)
00-00-93-10-04-15   Registered            1 (Eth/2)
01-80-C2-00-00-00   Registered            1,3
```

filtering *datagroup-option*

displays general information about the bridge’s protocol filtering databases. There are a number of general datagroups which may be displayed under the **list filtering** command. These include the following:

- All - Displays all filtering database entries.
- Ethertype - Displays Ethernet protocol type filter database entries.
- SAP - Displays SAP protocol filter database entries.
- SNAP - Displays SNAP protocol identifier filter database entries.

The following examples break down each of the list filtering display options.

Example: `list filtering all`

```
Ethernet type 0800 is routed on ports 1
IEEE 802.2 destination SAP 42 is routed on ports 1
IEEE 802 SNAP PID 00-00-00-08-00 is routed on ports 2-3
```

Descriptors used in explaining how packets are communicated include the following:

- Routed - Describes packets which are passed to routing forwarder to be forwarded
- Filtered- Describes packets which are administratively filtered by the user setting protocol filters
- Bridged and routed - This describes a protocol identifier for which there is a protocol entity within the system which is not a forwarder. An example of this would be a link level echo protocol. Unicast packets from this protocol are bridged or locally processed if being sent to a registered address. Multicast packets are forwarded and locally processed for a registered multicast address.

All of the descriptors just explained also apply to ARP packets with this Ethertype.

Example: list filtering ethertype

```
Ethernet type (in hexadecimal), 0 for all [0]? 0800
Ethernet type 0800 is routed on ports 1
```

Example: list filtering SAP

```
SAP (in hexadecimal), 100 for all [100]? 42
IEEE 802.2 destination SAP 42 is routed on ports 1
```

Example: list filtering SNAP

```
SNAP Protocol ID, return for all [00-00-00-00-00]?
IEEE 802 SNAP PID 00-00-00-08-00 is routed on ports 2-3
```

source-routing

Displays source-routing bridge configuration information. There are a number of general datagroup options which may be displayed under the list source-routing command. These include the following:

- Configuration - Displays general information regarding the SRB bridge.
- Counters - Displays all SRB bridge counters.
- State - Displays contents of all related SR-TB bridge databases.

The following examples illustrate each of the list source-routing display options.

Example: list source-routing configuration

```
Bridge number:              1
Bridge state:               Enabled
Maximum STE hop count       14
Maximum ARE hop count       14
Virtual segment:            003
Port  Segment  Interface   State     MTU    STE Forwarding  LNM
2     001      TKR/1       Enabled   4399   Yes             ENA
3     002      TKR/2       Enabled   4399   Yes
```

Bridge number	The bridge number (in hexadecimal) assigned to this bridge.
Bridge State	Indicates whether bridging is enabled or disabled.
Maximum STE hop count	The maximum hop count for spanning tree explorer frames transmitting from the bridge for a given interface associated with source routing bridging.
Maximum ARE hop count	The maximum hop count for all route explorer frames transmitting from the bridge for a given interface associated with source routing bridging.
Virtual segment	The virtual segment number assigned for 1:N bridging.
Port	The numbers of ports associated with source routing bridging
Segment	The assigned segment numbers for networks associated with source routing bridging.
Interface	The associated interface names. “Adaptive” is listed for interfaces participating in the SR-TB feature and VPI/VCI for ATM.
State	The current port state (Enabled or Disabled).
MTU	The MTU size set for that port.

STE Forwarding	Indicates whether Spanning Tree Explorers received on this port are forwarded (Yes) and whether STEs from other ports go out this port.
LNM	Indicates whether LAN Network Manager (LNM) agents are enabled (ENA) or disabled (DIS) on that specific port.

The counters option has further subgroups of information which may be displayed using the list source-routing command. These include the following:

- All-ports - Displays counters for all ports.
- Port - Displays counters for a specific port.
- Segment - Displays counters for the port corresponding to a specific segment.

The following examples illustrate each of the list source-routing display options.

Example: `list source-routing counters all-ports`

```
ASRT>list source counters all-ports
Counters for port 2, segment 001, interface TKR/1
SRF frames received:          0    sent:          0
STE frames received:          0    sent:          0
ARE frames received:        648    sent:          0
SR frames sent as TB:                             0
TB frames sent as SR:                          2057
Dropped, input queue overflow:                    0
Dropped, source address filtering:                0
Dropped, dest address filtering:
Dropped, invalid RIF length:                      0
Dropped, duplicate segment:                    2594
Dropped, segment mismatch:                        0
Dropped, Duplicate LAN ID or tree error:          0
Dropped, STE hop count exceeded:                  0
Dropped, ARE hop count exceeded:                  0
Dropped, no buffer available to copy:             0
Dropped, MTU exceeded:                            0


Counters for port 3, segment 002, interface TKR/2:
SRF frames received:          0    sent:          0
STE frames received:          0    sent:          0
ARE frames received:        825    sent:          0
SR frames sent as TB:                             0
TB frames sent as SR:                          2041
Dropped, input queue overflow:                    0
Dropped, source address filtering:                0
Dropped, dest address filtering:                  0
Dropped, invalid RIF length:                      0
Dropped, duplicate segment:                    3300
Dropped, segment mismatch:                        0
Dropped, Duplicate LAN ID or tree error:          0
Dropped, STE hop count exceeded:                  0
Dropped, ARE hop count exceeded:                  0
Dropped, no buffer available to copy:             0
Dropped, MTU exceeded:                            0
```

Port	Lists the numbers of ports associated with source routing bridging
Segment	Lists the source-routing segment numbers in hex.
Interface	Lists the name of the network interface. VPI/VCI for ATM.
SRF Frames Received/Sent	Lists the number of Specifically Routed Frames received or sent on this bridge.

STE Frames Received/Sent	Lists the number of Spanning Tree Explorer Frames received or sent on this bridge.
ARE Frames Received/Sent	Lists the number of All Routes Explorer Frames received or sent on this bridge.
SR Frames Sent as TB	Lists the number of source routing frames received on this interface that were sent as Transparent Bridge Frames.
TB Frames Sent as SR	Lists the number of transparent bridge frames received on this interface that were sent as source routing frames.
Dropped, input queue	Lists the number of frames arriving on this interface that were not bridged for flow control reasons. The input queue to the forwarder overflowed
Dropped, source address filtering	Lists the number of frames arriving on this interface that were not bridged because this source address matched a source address filter in the filtering database
Dropped, destination address filtering	Lists the number of frames arriving on this interface that were not bridged because this destination address matched a destination address filter in the filtering database
Dropped, protocol filtering	Lists the number of frames arriving on this interface that were not bridged because their protocol identifier was one that is being administratively filtered.
Dropped, invalid RIF length	Lists the number of frames arriving on this interface that were dropped because the RIF length as less than 2 or over 30.
Dropped, duplicate segment	Lists the number of frames arriving on this interface that were dropped because of a duplicate segment in the RIF. This is normal for ARE frames.
Dropped, segment mismatch	Lists the number of frames arriving on this interface that were dropped because the outgoing segment number does not match any in this bridge.
Dropped, Duplicate LAN ID or tree error:	The number of duplicate LAN IDs or Tree errors. This helps in the detection of problems in networks containing older IBM Source Routing Bridges.
Dropped, STE hop count exceeded:	The number of explorer frames that have been discarded by this port because the Routing Information Field has exceeded the maximum route descriptor length.
Dropped, ARE hop count exceeded:	The number of explorer frames that have been discarded by this port because the Routing Information Field has exceeded the maximum route descriptor length.
Dropped, no buffer available to copy:	Number of times a frame was not forwarded out of an interface, because there were no buffer resources available to copy the frame. (Frame to multicast destinations and to unknown destinations, need to be copied for transmission out on all active ports.)
Dropped, MTU exceeded:	The number of frames that were discarded by this port due to an excessive size. It is incremented by both transparent and source route bridges.

Example: list source-routing counters port 3

```
Counters for port 3, segment 002, interface TKR/1:
SRF frames received:          0    sent:           0
STE frames received:          0    sent:           0
ARE frames received:       1140    sent:           0
SR frames sent as TB:                                0
TB frames sent as SR:                             2931
Dropped, input queue overflow:                       0
Dropped, source address filtering:                   0
Dropped, dest address filtering:                     0

Dropped, invalid RIF length:                         0
Dropped, duplicate segment:                       4560
Dropped, segment mismatch:                           0
Dropped, Duplicate LAN ID or tree error:             0
Dropped, STE hop count exceeded:                     0
Dropped, ARE hop count exceeded:                     0
Dropped, no buffer available to copy:                0
Dropped, MTU exceeded:                               0
Dropped, dest address filtering:                     0
Dropped, protocol filtering:                         0
```

Example: list source-routing counters segment 2

```
Counters for port 3, segment 002, interface TKR/2:
SRF frames received:          0    sent:           0
STE frames received:          0    sent:           0
ARE frames received:       1249    sent:           0
SR frames sent as TB:                                0
TB frames sent as SR:                             3200
Dropped, input queue overflow:                       0
Dropped, source address filtering:                   0
Dropped, dest address filtering:                     0
Dropped, protocol filtering:                         0
Dropped, invalid RI length:                          0
Dropped, duplicate segment:                       4996
Dropped, segment mismatch:                           0
Dropped, Duplicate LAN ID or tree error:             0
Dropped, STE hop count exceeded:                     0
Dropped, ARE hop count exceeded:                     0
Dropped, no buffer available to copy:                0
Dropped, MTU exceeded:                               0
```

spanning-tree protocol

Displays spanning tree protocol information. The spanning tree protocol is used by the transparent bridge to form a loop-free topology. There are a number of general datagroup options which may be displayed under the **list spanning-tree-protocol** command. These include the following:

- Configuration - Displays information concerning the spanning tree protocol.
- Counters - Displays the spanning tree protocol counters.
- State - Displays the current spanning tree protocol state information.
- Tree - Displays the current spanning tree information including port, interface, and cost information.

The following examples illustrate each of the list spanning-tree-protocol display options.

Example: list spanning-tree-protocol configuration

```
Bridge ID (prio/add):  32768/0000-93-00-84-EA
Bridge state:          Enabled
Maximum age:           20 seconds
Hello time:            2 seconds
Forward delay:         15 seconds
Hold time:             1 seconds
Filtering age:         320 seconds
Filtering resolution:  5 seconds

Port  Interface  Priority  Cost   State
  4   Eth/1      128    100     Enabled
128   Tunnel     128    65535   Enabled
```

Example: list spanning-tree-protocol counters

```
  Time since topology change (seconds)         0
  Topology changes:                             1
  BPDUs received:                              0
  BPDUs sent:                                  14170

Port  Interface  BPDUs received  BDPU input overflow  Forward transitions

  1  TKR/1             0              0              1
  2  AT/0:0:48         0              0              0
```

Example: list spanning-tree-protocol state

```
Designated root (prio/add):    32768/00-00-93-00-84-EA
Root cost:                     0
Root port:                     Self
Current (root) maximum age:    20 seconds
Current (root) hello time:     2 seconds
Current (root) Forward delay:  15 seconds
Topology change detected:      FALSE
Topology change:               FALSE

Port  Interface   State
  4  Eth/1       Forwarding
128  Tunnel      Forwarding
```

Example: list spanning-tree-protocol tree

```
Port                 Designated   Desig.         Designated   Des.
No.  Interface            Root     Cost             Bridge   Port

1  TKR/1  32768/12-34-56-78-90-12    0 32768/12-34-56-78-90-12 90-01
2  AT/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 all-ports after the command to display the counters for all ports or enter the specific port number after the command to display counters for one port.
- State - Displays the transparent state information.

  The following examples illustrate each of the list transparent display options.

Example: list transparent configuration

```
Filtering database size:    5141
Aging time:                 300 seconds
Aging granularity           5 seconds
Port  Interface   State     MTU
  4   Eth/1       Enabled   0
128   Tunnel      Enabled   0
```

Example: list transparent counters all-ports

```
Counters for port 4, interface Eth/1:
Total frames received by interface:          25885
Frames submitted to bridging:                13732
Frames submitted to routing:                  6101
Dropped, source address filtering:               0
Dropped, dest address filtering:             12677
Dropped, protocol filtering:                     0
Counters for port 128, interface Tunnel:
Total frames received by interface:              0
Frames submitted to bridging:                    0
Frames submitted to routing:                     0
Dropped, source address filtering:               0
Dropped, dest address filtering:                 0
Dropped, protocol filtering:                     0
Dropped, no buffer available to copy:            0
Dropped, input queue overflow:                   0
Dropped, source port blocked:                    0
Frames sent by bridging:                      5327
Dropped, dest port blocked:                      0
Dropped, transmit error:                         0
Dropped, too big to send on port:                0
```

Example: list transparent counters port 4

```
Counters for port 4, interface Eth/1:
Total frames received by interface:          25885
Frames submitted to bridging:                13732
Frames submitted to routing:                  6101
Dropped, source address filtering:               0
Dropped, dest address filtering:             12677
Dropped, protocol filtering:                     0
Dropped, no buffer available to copy:         6073
Dropped, input queue overflow:                 122
Dropped, source port blocked:                   31
Frames sent by bridging:                       388
Dropped, dest port blocked:                      0
Dropped, transmit error:                         0
Dropped, too big to send on port:                0
```

Example: list transparent state

```
Filtering database size:                      5141
Number of static entries:                        0
Number of dynamic entries:                      10
Hash collision count:                            1
Filtering database overflow count:               0
```

tunnel *bridges* or *config*

Displays tunnel configuration information. There are general datagroup options which may be displayed under the list tunnel command. These include:

- Bridges - Displays tunnel bridge information.
- Config - Displays information concerning the tunnel configuration.

The following examples illustrate each of the list tunnel display options.

Example: list tunnel bridges

Example: list tunnel config

NetBIOS

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

See "NetBIOS Commands" on page 8-15 for the NetBIOS console commands.

Syntax: netbios

Example: **netbios**

```
NetBIOS>
```

Dynamic Protocol Filtering (VLANS)

The VLAN console commands are a superset of the VLAN configuration commands. However, instead of updating the SRAM configuration records immediately, they change the behavior of VLANs in real-time. Changes made through the console can be optionally saved to SRAM. Also, the configuration in SRAM can be loaded and used without requiring a reboot.

Console commands are entered at the VLAN> prompt. This prompt is accessed by entering the **VLANS** command at the ASRT> prompt. The following table shows the VLAN console commands.

Table 7-3. VLAN Console Command Summary

Command	Function
? (Help)	Lists all of the VLAN filtering console commands, or lists the options associated with specific commands.
Add	Adds a VLAN filter for an IP subnet, IPX network, or NETBIOS
Change	Changes VLAN filtering parameters for an IP subnet, IPX network, or NETBIOS
Delete	Deletes the selected VLAN filter(s)
Disable	Disables VLAN filtering on the selected subnet(s)
Enable	Enables VLAN filtering on the selected subnet(s)
List	Displays all information associated with the selected VLAN filter(s)
Load	Loads and uses the VLAN configuration currently in SRAM
Reset-Counters	Resets all counters associated with the selected VLAN filter(s)
Save	Saves the current runtime configuration to SRAM
Exit	Exits the VLAN filtering configuration process and returns to the ASRT environment

? Help See "Dynamic Protocol Filtering (VLANS) Configuration Commands" on page 6-38 for a description of this parameter.

Add See "Dynamic Protocol Filtering (VLANS) Configuration Commands" on page 6-38 for a description of this parameter.

Change See "Dynamic Protocol Filtering (VLANS) Configuration Commands" on page 6-38 for a description of this parameter.

Delete See "Dynamic Protocol Filtering (VLANS) Configuration Commands" on page 6-38 for a description of this parameter.

Disable See “Dynamic Protocol Filtering (VLANS) Configuration Commands” on page 6-38 for a description of this parameter.

Enable See “Dynamic Protocol Filtering (VLANS) Configuration Commands” on page 6-38 for a description of this parameter.

List Use the list command to list the current real-time configuration for a particular VLAN filter, all VLAN filters for a particular protocol, or all defined VLAN filters. If listing a single filter, the VLAN to list can be chosen by specifying the subnet or by selecting the VLAN from a list with the *by-name* option. The resulting output includes both configuration parameters and VLAN counters.

Syntax:

list by-name

ip all

ip subnet *subnet address*

ipx all

ipx network *network number*

netbios

all

Example: `list ip subnet 9.0.0.0`

```
Subnet Address              = 9.0.0.0
Subnet Mask                 = 255.0.0.0
Bridge Port 1 (Interface 0) = Auto-Detect and Include, Forwarding
Bridge Port 2 (Interface 1) = Always Exclude, Not Forwarding
Age (expiration in minutes) = 300
IP-Cut-Through Status:
   Tx From This VLAN    = Enabled  Reception By This VLAN = Disabled
   Packets Transmitted  = 25       Packets Received       = 0
   Tx Packets Discarded = 0        Rx Packets Discarded   = 14
VLAN Status               = Enabled
Packets Processed          = 43
Discards Due To Exclusion  = 13
VLAN Name                   = IP 9.x.x.x
```

A description of the VLAN counters follows:

Packets Transmitted

Total number of IP packets successfully cut-through from this VLAN.

Packets Received

Total number of IP packets successfully cut-through to this VLAN.

Tx Packets Discarded

Number of IP packets that were intended to be cut-through from this VLAN, but were discarded due to IP-Cut-Through transmission being disabled. Packets from ports configured as Always Exclude are not included in this count.

Rx Packets Discarded

Number of IP packets that were intended to be cut-through to this VLAN, but were discarded due to IP-Cut-Through reception being disabled.

Packets Processed

Total number of packets processed by this VLAN's forwarding logic. This includes all packets forwarded and discarded.

Discards Due To Exclusion

Number of packets received with a source subnet matching this VLAN on ports configured as Always Exclude for this VLAN.

Load Use the load command to load and immediately use the VLAN configuration stored in SRAM. This will overwrite any configuration changes that may have been made via the console since the last save. All timers and counters associated with VLANs will be reset.

Syntax: load

Example: load

```
Warning: This process will overwrite your current configuration.
Are you sure you want to load the VLAN configuration from SRAM? [No] y
VLAN configuration loaded
```

Reset-Counters

Use the reset-counters command to set all counters to zero for a particular VLAN filter, all VLAN filters for a particular protocol, or all defined VLAN filters. If resetting the counters in a single filter, the VLAN can be chosen by specifying the subnet or by selecting the VLAN from a list with the by-name option.

Syntax:

reset-counters by-name

ip all

ip subnet *subnet address*

ipx all

ipx network *network number*

netbios

all

Example: reset ipx network 3ff

```
VLAN 'Token Ring B' (IPX Network 0x3FF) counters reset
```

Save Use the save command to store the current runtime VLAN configuration into SRAM. This will overwrite the current SRAM configuration. This command does not affect the runtime behavior of VLANs or reset the timers or counters associated with VLANs.

Syntax: save

Example: save

```
Are you sure you want to save the VLAN configuration to SRAM? [No] y
VLAN configuration saved
```

Exit See “Dynamic Protocol Filtering (VLANS) Configuration Commands” on page 6-38 for a description of this parameter.

Exit

Use the **exit** command to exit the ASRT console process and return to the GWCON environment.

Syntax: exit

Example: `exit`

Chapter 8. Using, Configuring, and Monitoring NetBIOS

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

About NetBIOS

The NetBIOS protocol was designed for use on a LAN (token ring). It is not a routable protocol, but can be bridged.

NetBIOS relies on broadcast frames for most of its functions other than data transfer. While this may not present a problem in LAN environments, if uncontrolled, it may easily present a problem in WAN environments.

The following sections describe NetBIOS names and the different types of NetBIOS broadcast communication.

NetBIOS Names

The key to communication between NetBIOS stations are the NetBIOS names. Each NetBIOS entity is assigned a NetBIOS name. In order to communicate with another NetBIOS entity, its NetBIOS name must be known. The names are used in broadcast NetBIOS frames to indicate the source NetBIOS entity of the frame and the desired target NetBIOS entity to receive the frame.

All names in NetBIOS frames are 16 ASCII characters. There are two types of NetBIOS names:

Individual (or unique)
Represents a single NetBIOS client or server. This name should be unique within the NetBIOS network.

This name is used to communicate with this particular NetBIOS entity.

Group Represents a group of NetBIOS stations (an OS/2 LAN Server domain, for example). This name should not be the same as any individual NetBIOS names in the network.

This name is used to allow communication between a group of NetBIOS entities.

A single NetBIOS station (single MAC address) may have multiple individual and/or group names associated with it. These names are generated by the NetBIOS application based upon a name or names configured at the NetBIOS station by a network administrator.

NetBIOS Name Conflict Resolution

When a NetBIOS entity is preparing to use an individual NetBIOS name as its own, it checks the network to make sure that no other NetBIOS station has already used this name.

It checks the NetBIOS name by repeatedly broadcasting a particular NetBIOS UI frame to all NetBIOS stations. If no stations respond, then the name is assumed to be unique and can be used. If a station does respond, the new station should not attempt to use this name.

NetBIOS Session Setup Procedure

To establish a NetBIOS session in order to do data transfer types of operations, the NetBIOS client first resolves the MAC address of the NetBIOS server and the LLC route to the NetBIOS server.

This is done by repeatedly broadcasting a particular NetBIOS UI frame to all NetBIOS stations. This frame contains the NetBIOS name of the server with which this client is establishing a session. When the server receives this frame with its NetBIOS name in it, it responds with a corresponding broadcast NetBIOS UI frame to the client. When the client receives the response frame, the frame contains the MAC address and the route to the NetBIOS server.

For some NetBIOS applications, finding the NetBIOS server is a multiple step process. For example, the first step may be to find a domain controller which tells the client which domain server to use. Then the client finds this domain server.

Once the MAC address of NetBIOS server and the route to the NetBIOS server is found, the NetBIOS client may take either of the following actions:

- Establish an LLC2 connection with the NetBIOS server to communicate with the server using I-frames.
- Begin communicating with the NetBIOS server using specifically-routed NetBIOS UI frames.

NetBIOS Broadcast Data Flows

For some NetBIOS applications, it is common to periodically broadcast data frames. This may be done if a station has a single frame's worth of data to send to another NetBIOS station. It can do this by broadcasting a particular NetBIOS UI frame (with the target NetBIOS station's name in the frame) to all NetBIOS stations.

Another case is when NetBIOS stations within a group (or domain) need to communicate with each other. This can be done by broadcasting a particular NetBIOS UI frame (with the target NetBIOS group name in the frame) to all NetBIOS stations. This is commonly done.

NetBIOS Status Flows

A less commonly used NetBIOS function is the ability to obtain status from any NetBIOS station. This is done by broadcasting a particular NetBIOS frame (with the target NetBIOS station's name in the frame) to all NetBIOS stations. When the target NetBIOS station receives this frame, it responds with a corresponding broadcast NetBIOS response frame.

NetBIOS All-Stations Broadcast Frames

There are two types of NetBIOS functions that are rarely used. Both of these functions involve broadcasting a NetBIOS frame to all NetBIOS stations. There is no target NetBIOS name in the frames. The two functions are:

- NetBIOS general broadcast function – which sends a data frame to all NetBIOS stations on the network.
- NetBIOS terminate trace function – which allows a network administrator to terminate NetBIOS trace functions in all NetBIOS stations on the network from a single point. A particular NetBIOS frame is broadcast to all NetBIOS stations on the network.

Reducing NetBIOS Traffic

To stabilize a network, the goal is to reduce the amount of ;when 'ncer1' delete broadcast NetBIOS traffic that is forwarded through the bridged or DLSw switched broadcast NetBIOS traffic that is forwarded through the bridged networks. This can be done in two ways:

- Filter as many broadcast NetBIOS frames as possible before bridging them.
- Forward unfiltered NetBIOS UI frames on as few bridge ports as possible.

Table 8-1 lists the filters that IBM provides.

Table 8-1. NetBIOS Filters

Filter Type	Filters
MAC Address	Frames by either the source or destination MAC address.
Byte	Frames by byte offset and field length within a frame.
Name	Frames by NetBIOS source and destination names.
Duplicate Frame	Duplicate frames.
Response	Responses for which the router did not forward a NetBIOS broadcast frame.

Once the router filters frames, NetBIOS name caching and route caching controls how the remaining frames are forwarded. “NetBIOS Byte Filtering” on page 3-5 and “NetBIOS Host Name Filtering” on page 3-5 in this manual describe byte and name filtering. The *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1* describes MAC address filtering.

An introduction to host-name filtering and byte filtering can be found in “NetBIOS Name and Byte Filters” on page 3-4.

The next sections describe frame type, duplicate frame, and response frame filtering, as well as NetBIOS name and route caching.

Frame Type Filtering

Frame type filtering allows certain categories of NetBIOS frames to be filtered entirely for 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.

  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. 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 example:

```
NetBIOS config>set filters bridge

Filter Name Conflict frames? [Yes]:

Name conflict filtering is              ON

Filter General Broadcast frames? [Yes]:

General broadcast filtering is          ON

Filter Trace Control frames? [Yes]:

Trace control filtering is              ON
```

Duplicate Frame Filtering

All of the broadcast NetBIOS frames which could have a response are sent a fixed number of times (default 6), at a fixed interval (default 1/2 second apart) by the origin NetBIOS station. We will call these frames “NetBIOS command frames” and we will call the possible response frames “NetBIOS response frames.”

The NetBIOS command frames are the:

- Name conflict resolution frames – Add-Name-Query and Add-Group-Name-Query
- NetBIOS session setup frames – Name-Query
- NetBIOS status frames – Status-Query

The command frames are sent multiple times to increase the odds of successful delivery (these frames are connectionless frames). Each response frame is sent only once in response to each command frame received.

There is one configurable time period for the bridge 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.

There is one last parameter that controls how long the command frame is saved in order to perform the above bridge 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.

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 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 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 bridge ports as possible. The possible types of NetBIOS names and the information saved for each are:

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, the frame is forwarded on all bridge ports.
- 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.

Learning NetBIOS Names

NetBIOS names are learned and classified from information in the NetBIOS session setup frames (Name-Query and Name-Recognized).

Configuring Name Cache Parameters

To prevent one type of NetBIOS name from filling up the entire name cache, there is a configurable NetBIOS name cache limit:

- 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.

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 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 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.

The last parameter, significant characters in name, controls how many of the 16 characters in a NetBIOS name are needed to consider it a unique NetBIOS name. Some NetBIOS applications use the 16th character of the NetBIOS name to distinguish between certain entities associated with a single NetBIOS name (for example, print server and file server). In these cases, it is best to specify significant characters in name as 15. This causes any frame in which the first 15 characters of the destination NetBIOS name matches the first 15 characters of the router's NetBIOS name cache entry to be forwarded according to the name cache entry information. Thus multiple NetBIOS names can be represented with a single NetBIOS name cache entry.

All of the above NetBIOS name cache related parameters can be configured using the **set cache-parms** command as follows.

```
NetBIOS config>set cache-parms

Significant characters in name [15]?
Best path aging timeout value in seconds [60.0]?
Reduced search timeout value in seconds [1.5]?
Unreferenced entry timeout value in minutes [5000]?
Max nbr local name cache entries [500]?

Cache parameters set
```

See "NetBIOS Commands" on page 8-15 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 8-2 on page 8-8.

Table 8-2. NetBIOS List Cache Configuration Commands	
Command	**Displays . . .**
list cache all	All permanent entries. Does not show static and dynamic entries.
list cache entry-number	A permanent cache entry according to its entry number.
list cache netbios-name	A permanent cache entry for a specific NetBIOS name.

From the NetBIOS monitoring prompt, you can use the list cache commands in Table 8-3.

Table 8-3. NetBIOS List Cache Monitoring Commands	
Command	**Displays . . .**
list cache active	All active entries in the router's name cache, including permanent, static, and dynamic entries.
list cache 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 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 8-15.

To create a host name filter, enter commands at the `NetBIOS Filter config>` prompt.

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

NetBIOS Support User Configuration

NetBIOS config>set filter name
NETBIOS Filtering configuration
NETBIOS Filter config>
```

Creating a Host Name Filter

Use the following procedure as a guideline to creating a host name filter.

1. Create an empty name filter list.

   ```
   NetBIOS Filter config>create name-filter-list
   Handle for Name Filter List []? boston
   ```

2. Add the filter items to the name filter list.

   Enter **update** to get to the prompt for that specific filter list. From this prompt, you can add filter items to the filter list.

```
NetBIOS Filter config>update
Handle for Filter List []? boston
Name Filter List Configuration
NetBIOS Name boston config>
```

3. Add filter items to the filter list with the **add** command. The way filter items are configured determines which NetBIOS packets are bridged or dropped. Host name filter items are configured with the following parameters entered in this order:

   - *Inclusive* (bridged) or *Exclusive* (dropped).
   - *ASCII* or *HEX* - how the hostname is represented.
   - *host name* - the actual host name represented in either an ASCII or hex string (see the command section that follows for syntax). This entry is case sensitive.
   - <*LAST-hex-number*> - an optional parameter for use with ASCII strings containing fewer than 16 characters.

   The following example adds a filter item to the Host Name Filter list **boston**, which allows packets containing the hostname **westboro** (an ASCII string) to be bridged (configured as *inclusive*). No <*LAST-hex-number*> parameter has been configured for this entry.

   ```
   NetBIOS Name boston config>add inclusive ascii
   Hostname []? westboro
   Special 16th character in ASCII hex (<CR> for no special char) []?
   ```

   You can enter all parameters as one string on the command line if you do not want to be prompted. Be sure to use a space between each parameter.

4. Verify the filter item entry.

   Type **list** to verify your entry:

   ```
   NetBIOS Name boston config>list

   NAME Filter List Name: boston
   NAME Filter List Default: Inclusive

    Item #   Type    Inc/Ex    Hostname         Last Char

      1      ASCII    Inc      westboro
   ```

5. Add additional filter items to the filter list.

   Repeat the first four steps to add additional filter items to the filter list. The order in which you enter filter items is important as this determines how the router applies the filter items to a packet. The first match stops the application of filter items and the router either forwards or drops the packet, depending on whether the filter item is Inclusive or Exclusive.

   Entering the most common filter items first makes the filtering process more efficient because the software is more likely to make a match at the beginning of the list.

   If the packet does not match any of the filter items, the router uses the default condition (Inclusive or Exclusive) of the filter list. You can change the default condition of the list by entering **default inclusive** or **default exclusive** at the filter list configuration prompt. For example:

   ```
   NETBIOS Name boston config> default exclusive
   ```

6. When you have finished adding filter items to the filter list, enter **exit** to return to the `NetBIOS Filter config>` prompt.

```
NetBIOS Name boston config>exit
NetBIOS Filter config>
```

7. Add the filter to your configuration.

   The filter list containing the filter items can now be added as a filter to your bridging router configuration. Use the **filter-on** command to do this. Host name filters are configured with the following parameters (entered in this order):

   - *Input* (to filter all NetBIOS packets received on that port) or output (to filter all NetBIOS packets transmitted on that port).
   - *Port#*, which is the desired configured bridge port number on the router.
   - *Filter-list*, which is the name of the filter list (containing filter items) that you want to be included in this filter.
   - An optional operator entered as either AND or OR in all capital letters. If an operator is present, it must be followed by a filter-list name. Filters with more than one filter list are called complex filters.

   The following example adds a host name filter to affect packets input on port #3. It is comprised of the host name filter list **boston**. All packets input on port #3 are evaluated according to the rules provided by the filter items contained in the filter list **boston**. This means that all packets input on port #3 containing the hostname **westboro** are bridged.

   ```
   NetBIOS Filter config>filter-on input
   Port Number [1]? 3
   Filter List []? boston
   ```

8. Verify the newly created filter.

   Enter **list** to verify your entry:

   ```
   NetBIOS Filter config>list

   NetBIOS Filtering: Disabled

   NetBIOS Filter Lists
   --------------------

       Handle           Type
       nlist            Name
       newyork          Name
       HELLO            Byte
       boston           Name

   NetBIOS Filters
   ---------------

       Port #      Direction      Filter List Handle(s)
          3          Output       nlist
          1          Input        newyork OR HELLO
          3          Input        boston
   ```

9. Globally enable NetBIOS filtering.

   Use the **enable** command to globally enable NetBIOS filtering on the router.

   ```
   NetBIOS Filter config>enable Netbios-filtering
   ```

10. Restart the router to activate all NetBIOS filtering configuration changes.

    Enter **exit** followed by **Ctrl** **P** to return to the * prompt. From this prompt, enter **restart** to activate all software changes made during the NetBIOS filtering configuration process.

```
NetBIOS Filter config>exit
ASRT config>exit
Config>  Ctrl  P
* restart
```

Creating a Byte Filter

Use the following procedure as a guideline for creating a byte filter. Enter all commands at the NetBIOS filtering config> prompt.

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

NetBIOS Support User Configuration

NetBIOS config> set filter byte
NetBIOS Filtering configuration
NetBIOS Filter config>
```

1. Create an empty filter list using the **create byte-filter-list** command.

   ```
   NetBIOS Filter config>create byte-filter-list
   Handle for Byte Filter List []? westport
   ```

2. Add the filter items to the byte filter list.

   Enter **update** to get to the prompt for that specific filter list. From this prompt you can add filter items to the filter list.

   ```
   NetBIOS Filter config>update
   Handle for Filter List []? westport
   Byte Filter List Configuration
   NetBIOS Byte westport config>
   ```

   Begin adding filter items to the filter list with the **add** command. The way filter items are configured determines which NetBIOS packets are bridged or dropped. Byte filter items are configured with the following parameters (entered in this order):

   - Inclusive (bridged) or Exclusive (dropped).
   - Byte Offset - the number of bytes (in decimal) to offset into the packet being filtered. This starts at the NetBIOS header of the packet. Zero specifies that the router will examine all bytes in the packet.
   - Hex pattern - a hexadecimal number used to compare with the bytes starting at the byte offset of the NetBIOS header. See “NetBIOS Commands” on page 8-15 for syntax rules.
   - Hex mask - (if present) must be the same length as hex pattern and is logically ANDed with the bytes in the packet starting at byte-offset before the result is compared for equality with hex pattern. If the *hex-mask* argument is omitted, it is considered to be all binary ones.

   The following example adds a filter item to the Byte filter list **westboro** that allows packets with a hex pattern 0x12345678 at byte offset of 0 to be bridged (configured as inclusive). No hex mask is present.

   ```
   NetBIOS Byte westport config>add inclusive
   Byte Offset [0]? 0
   Hex Pattern []? 12345678
   Hex Mask (<CR> for no mask) [[?
   ```

3. Verify the filter item entry with the **list** command.

```
NetBIOS Byte westport config>list

BYTE Filter List Name: westport
BYTE Filter List Default: Inclusive

 Item #   Inc/Ex   Offset  Pattern        Mask

   1       Inc        0    0x12345678     0xFFFFFFFF
```

4. Add additional filter items to the filter list.

   Repeat the first three steps to add additional filter items to the filter list.

5. When you have finished adding filter items to the filter list, type **exit** to return to the `NetBIOS Filter config>` prompt.

   ```
   NetBIOS Byte westport config>exit
   NetBIOS Filter config>
   ```

   The order in which you enter filter items is important, as this determines how the router applies the filter to a packet. The first match stops the application of filter items and the router either forwards or drops the packet, depending on whether the filter item is Inclusive or Exclusive.

   Entering the most common filter items first makes the filtering process more efficient because the software is more likely to make a match at the beginning of the list rather than having to check the whole list before making a match.

   If the packet does not match any of the filter items, the router uses the default condition (Inclusive or Exclusive) of the filter list. You can change the default condition of the list by entering **default inclusive** or **default exclusive** at the filter list configuration prompt. For example:

   ```
   NETBIOS Byte westport config> default exclusive
   ```

6. Add the filter to your configuration.

   The filter list containing the filter items can now be added as a filter to your bridging router configuration. Use the **filter-on** command to do this. Host name filters are configured with the following parameters (entered in this order):

   - *Input* (to filter all packets received on that port) or output (to filter all packets transmitted on that port).
   - *Port#* - the configured bridge port number.
   - *Filter-list* - the name of the filter list (containing filter items) that you want included in this filter,
   - An optional operator entered as either AND or OR. The operator is entered in all capital letters. If an operator is present, it must be followed by a filter-list name. Filters with more than one filter list are called complex filters. These are explained in more detail in “About NetBIOS Configuration and Monitoring Commands” on page 8-13.

   The following example adds a host name filter to affect packets output on port #3. It is comprised of the byte filter list **westboro**. All packets output on port #3 will be evaluated according to the rules provided by the filter items contained in the filter list **westboro**.

   ```
   NetBIOS Filter config>filter-on output
   Port Number [1]? 3
   Filter List []? westboro
   ```

7. Verify the newly created filter.

   Enter **list** to verify your entry:

```
NetBIOS Filter config>list

NetBIOS Filtering: Disabled

NetBIOS Filter Lists
--------------------

    Handle           Type
    nlist            Name
    newyork          Name
    HELLO            Byte
    westboro         Byte

NetBIOS Filters
---------------

    Port #      Direction      Filter List Handle(s)
       3          Output       nlist
       1          Input        newyork OR HELLO
       3          Output       westboro
```

8. Globally enable NetBIOS filtering.

   Enter **enable** to globally enable NetBIOS filtering on the bridging router.

   ```
   NetBIOS Filter config>enable netbios-filtering
   ```

9. Restart the router to activate all NetBIOS filtering configuration changes.

   Enter **exit** followed by **Ctrl** **P** to return to the * prompt. Enter **restart**.

   ```
   NetBIOS Filter config>exit
   ASRT config>exit
   Config> Ctrl P
   * restart
   ```

About NetBIOS Configuration and Monitoring Commands

NetBIOS configuration commands are available at the ASRT/ NetBIOS config> prompt. Changes you make to the router's configuration do not take effect immediately. They become part of the router's configuration memory when you restart it. This chapter refers to configuration changes as permanent.

NetBIOS monitoring commands are available at the ASRT/ NetBIOS> prompt. Monitoring commands take effect immediately, but are not saved in the router's non-volatile configuration memory. Thus, while monitoring commands allow you to make real-time changes to the router's configuration, these changes are temporary. The router's configuration memory overwrites them when the router restarts. This chapter refers to changes you make at the monitoring prompt as static.

Accessing the NetBIOS Configuration Environment

You can display the NetBIOS config> prompt from the ASRT configuration environment.

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>
```

Accessing the NetBIOS Console Environment

You can display the NetBIOS> prompt from the ASRT monitoring environment.

To display the NetBIOS> console prompt from the ASRT monitoring environment:

```
+ protocol asrt
ASRT>netbios

NetBIOS Support User Console

NetBIOS>
```

NetBIOS Commands

Table 8-4 lists the NetBIOS configuration and monitoring commands.

Table 8-4. NetBIOS Configuration and Monitoring Commands

Command	Function
?(Help)	Lists available commands or options.
Disable	Disables duplicate frame filtering and route caching.
Enable	Enables duplicate frame filtering and route caching.
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.
Exit	Returns to the previous prompt.

? (Help)

Lists available commands or options.

Syntax: ?

Example: **`set ?`**

```
CACHE-PARMS
FILTERS
GENERAL
NAME-LIST
```

Disable

Disables duplicate frame filtering, use of NetBIOS name lists, or route caching.

Syntax: disable duplicate-filtering
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
```

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
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
```

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
filters all
filters bridge
general

cache all

Displays all permanent entries in the router's name cache. It does not display static or dynamic entries.

Example: **`list cache all`**

```
Entry  Name                IP Address
-----  ------------------  ---------------

    1  ACCOUNTING     <00>  20.2.1.3
    2  NOTES          <00>  20.2.3.4
```

cache entry-number *record#*

Displays a cache entry according to its entry number. Enter **list cache all** to see a list of entry numbers.

Example: **`list cache entry-number`**

```
Enter name cache record number [1]? 1

Entry  Name                IP Address
-----  ------------------  ---------------
    1  ACCOUNTING     <00>  20.2.1.3
```

cache name *name*

Displays a cache entry for a specific NetBIOS name. You can use the following wildcards to simplify your search:

***** (asterisk) stands for zero or more occurrences of any characters. For example, San* could produce:

- San Francisco
- Santa Fe
- San Juan

? (question mark) stands for any one character.

$ (dollar sign) has an effect only when the number of significant NetBIOS name characters is not 16, and when the search argument does not begin with an asterisk (*).

You can use as many wildcards as you like, up to the maximum number of characters in a NetBIOS name (15 or 16, depending on the configuration).

Note: The NetBIOS name is case sensitive.

Example: `list cache netbios-name`

```
Enter up to 15 characters of NetBIOS name (wild cards ok) []? Acc*

Entry  Name                IP Address
-----  ------------------  ---------------
    1  Accounting     <00>  20.2.1.3
```

filters all

Displays whether or not frame type filtering is on or off for bridging. 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
```

filters bridge

Displays whether or not frame type filtering is on or off for bridging. Use the **set filters bridge** to turn these filters on or off.

Example: `list filters bridge`

```
Bridge name conflict filtering is       OFF
Bridge general bcast filtering is       OFF
Bridge trace control filtering is       OFF
```

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

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
```

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 unknown
- filters all
- filters bridge
- general
- statistics cache
- statistics frames bridge
- statistics general bridge

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 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. 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
```

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
```

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


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
```

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 general bridge

Displays frame counts for bridging.

Example: `list statistics general bridge`

```
Frames received                       1339
Frames discarded                         0
Frames forwarded to bridge            1339
```

Set

Sets name caching parameters, turns frame type filtering on or off for bridging, 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 name
- general

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 the bridges.

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 100 to 100000 minutes.

Max nbr local name cache entries

Maximum number of locally-learned entries the router saves in the name cache.

The default is 500. The range is 1 to 30,000. You can lower this value to save router memory. To optimize memory usage, processor usage, and the amount of broadcast traffic, the number of local name cache entries should be set as close as possible to the total number of NetBIOS stations (servers and clients) that are active on this router's local bridge network.

Max nbr remote name cache entries

Maximum number of remotely-learned entries, group name entries, and unknown entries that the router saves in the name cache.

The default is 100. The range is 1 to 30,000. You can lower this value to save router memory. To optimize memory usage, processor usage, and the amount of broadcast traffic, the number of remote name cache entries should be set to the number of remote NetBIOS servers that are to be accessed by NetBIOS clients on this router's local bridge network, plus about 25%.

filters bridge

Turns frame-type filtering for bridging on or off.

Example: `set filters bridge`

```
Filter Name Conflict frames? [No]: y

Name conflict filtering is             ON

Filter General Broadcast frames? [No]:

General broadcast filtering is         OFF

Filter Trace Control frames? [No]:

Trace control filtering is             OFF
```

filters byte

From the `NetBIOS config>` prompt, displays the NetBIOS filtering configuration prompt (`NETBIOS Filter config>`). Configuring NetBIOS filtering is explained in Chapter 9, "Configuring NetBIOS Filtering."

From the `NetBIOS console>` prompt, displays the NetBIOS filtering monitoring prompt (`NETBIOS Filter>`). Monitoring NetBIOS filtering is explained in Chapter 10, “Monitoring NetBIOS Filtering.”

This parameter allows you to access NetBIOS byte filtering.

Example: `set filters byte`

```
NETBIOS Filtering configuration
NETBIOS Filter config>
```

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 NetBIOS Filtering.”

From the `NetBIOS console>` prompt, displays the NetBIOS filtering monitoring prompt (`NETBIOS Filter>`). Monitoring NetBIOS filtering is explained in Chapter 10, “Monitoring NetBIOS Filtering.”

This parameter allows you to access NetBIOS name filtering.

Example: `set filters name`

```
NETBIOS Filtering configuration
NETBIOS Filter config>
```

general

Sets the duplicate frame timeout, duplicate frame-detect timeout, and the command frame retry count and timeout. See “Duplicate Frame Filtering” on page 8-5 for more information on how duplicate frame filters work.

Example: `set general`

```
ATTENTION! Setting Duplicate Frame Filter Timeout to zero...
disables duplicate frame checking!

Duplicate frame filter timeout value in seconds [1.5]?

Duplicate frame detect timeout value in seconds [5.0]?

General parameters set
```

Duplicate frame filter timeout

Applies only to bridged traffic if duplicate-filtering is enabled. During this timeout period, the router filters all duplicate frames it receives.

The range is 0.0 to 100.0 seconds. Zero disables duplicate frame checking. The default is 1.5 seconds.

Duplicate frame-detect timeout

Applies to both bridged and DLSw traffic. Amount of time the router saves entries in its duplicate frame filter database. When this timer expires, the router creates new entries for new frames that it receives.

The range is 0.0 to 100.0 seconds. The default is 5 seconds.

Command frame retry count

Applies only to DLSw traffic.

Number of duplicate NetBIOS UI frames the target DLSw router sends to its locally-attached LAN. These frames are sent at intervals specified by the command frame retry timeout.

The range is 0 to 10. The default is 5.

Command frame retry timeout

Applies only to DLSw traffic. This is the interval at which a neighbor DLSw router retries sending duplicate NetBIOS UI frames to its local bridge network.

The range is 0.0 to 10.0 seconds. The default is 0.5 seconds.

Exit

Returns to the previous prompt.

Syntax: exit

Example: `exit`

Chapter 9. Configuring NetBIOS Filtering

This chapter summarizes and then explains all of the NetBIOS filtering configuration commands. These commands let you configure NetBIOS filtering as an added feature to ASRT bridging. Configuration commands are accessed from the `NetBIOS config>` prompt.

"NetBIOS Filtering Configuration Commands" is included.

Accessing the ASRT 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>
```

Table 9-1 shows the NetBIOS filtering configuration commands.

NetBIOS Filtering Configuration Commands

Table 9-1. NetBIOS Filtering Configuration Commands

Command	Function
? (Help)	Lists all of the NetBIOS filtering configuration commands, or lists the options associated with specific commands.
Create	Creates byte filter and host-name filter lists for NetBIOS filtering.
Delete	Deletes byte filter and host-name filter lists for NetBIOS filtering.
Disable	Disables NetBIOS filtering on the bridging router.
Enable	Enables NetBIOS filtering on the bridging router.
Filter-on	Assigns a created filter to a specific port. This filter can then be applied to all NetBIOS packets input OR output on the specified port.
List	Displays all information concerning created filters.
Update	Adds information to or deletes information from a host-name or byte filter list.
Exit	Exits the NetBIOS filtering configuration process and returns you to the previous prompt.

? (Help)

Use the **? (Help)** command to obtain a list of the commands available from that prompt level. You can also enter this command after specific command names to obtain a listing of the command options available for that command.

Syntax: ?

Example: `create ?`
```
byte-filter-list
name-filter-list
```

Create

Use the **create** command to create a byte filter or host-name filter list.

Syntax: create byte-filter-list *filter-list*
name-filter-list *filter-list*

byte-filter-list *filter-list*

Creates a byte filter list name for NetBIOS filtering. You can use up to 16 characters to identify the list being built. *Filter-list* must be a unique name that has not been used previously with the **create byte-filter-list** or **create name-filter-list** command.

Example: `create byte-filter-list newyork`

name-filter-list *filter-list*

Creates a host-name filter list name for NetBIOS filtering. You can use up to 16 characters to identify the name filter list being built. *Filter-list* must be a unique name that has not been used previously with the **create byte-filter-list** or **create name-filter-list** command.

Example: `create name-filter-list atlanta`

Delete

Use the **delete** command to delete byte filter lists, host-name filter lists, and filters created using the **filter-on input** or **filter-on output** command. The command removes all information associated with byte and host-name filter lists. It also frees the user-defined string as a name for a new filter list.

Syntax: delete byte-filter-list *filter-list*
name-filter-list *filter-list*
filter input *port#*
filter output *port#*

byte-filter-list *filter-list*

Deletes a byte filter list created for NetBIOS filtering. *Filter-list* is the user-defined string being used to identify the byte filter list being deleted.

Example: `delete byte-filter-list newyork`

name-filter-list *filter-list*

Deletes a host-name filter list created for NetBIOS filtering. *Filter-list* is the user-defined string that is used to identify the name-filter-list being deleted.

Example: `delete name-filter-list atlanta`

filter input *port#*

Deletes a filter that was created using the **filter-on input** command. The command removes all information associated with the filter and fills any resulting gap in filter numbers.

Example: `delete filter input 2`

filter output *port#*

Deletes a filter that was created using the **filter-on output** command. The command removes all information associated with the filter and fills any resulting gap in filter numbers.

Example: **`delete filter output 2`**

Disable

Use the **disable** command to globally disable NetBIOS name and byte filtering on the router.

Syntax: disable netbios-filtering

Example: **`disable netbios-filtering`**

Enable

Use the **enable** command to globally enable NetBIOS name and byte filtering on the router.

Syntax: enable netbios-filtering

Example: **`enable netbios-filtering`**

Filter-on

This command assigns one or more previously configured filter lists to the input or output of a specific port.

Syntax: filter-on input *port# filter-list <operator filter-list ...>*
output *port# filter-list <operator filter-list ...>*

input *port# filter-list <operator filter-list . . . >*

This command assigns one or more filter lists to incoming packets on a specific port. The resulting filter is then applied to all NetBIOS packets input on the specified port.

Port# is a configured bridge port number on the router. The port number identifies this filter. Enter **list** to see a list of port numbers Filter-list is a string previously entered via the **create** command. To add additional filter lists to this port, enter AND or OR in all capital letters followed by the filter list name.

Note: Multiple operators can be used to create a complex filter. If you enter multiple operators, they must all be entered at the same time on the same command line.

The filter created by this command is applied to all incoming NetBIOS packets on the specified port. Each filter list on the command line is evaluated left to right along with any operators that are present. An Inclusive evaluation of a filter list is equivalent to a TRUE condition and an Exclusive evaluation is equivalent to a FALSE condition. If the result of the evaluation of the filter-list(s) is TRUE, the packet is bridged. Otherwise, the packet is filtered (dropped).

If the packet is not one of the types supported by NetBIOS filtering then all host-name filter lists for this filter are designated “Inclusive” (TRUE). If an input filter already exists for specified port number, an error message is displayed.

Example: **`filter-on input 2 newyork AND boston`**

output *port# filter-list <operator filter-list . . . >*

This command assigns one or more filters to outgoing packets on a port. This filter is then applied to all NetBIOS packets output on that port.

Port# is a configured bridge port number on the router. The port number identifies this filter. Enter **list** to see a list of port numbers. Filter-list is a string previously entered via the create command. An optional operator is entered as either “AND” or “OR.” The operator is entered in all capital letters. If an operator is present, it must be followed by a filter-list name. The port number is used to identify this filter.

Note: Multiple operators can be used. This creates a complex filter. If one or more operators are present, they must all be entered at the same time on the same command line.

The filter created by this command is applied to all NetBIOS packets output on the specified port number. Each filter list on the command line is evaluated left to right along with any operators that are present. An Inclusive evaluation of a filter list is equivalent to a TRUE condition and an Exclusive evaluation is equivalent to a FALSE condition. If the result of the evaluation of the filter-list(s) is TRUE, the packet is bridged. Otherwise, the packet is filtered (dropped).

If the packet is not one of the types supported by NetBIOS filtering then all host-name filter lists for this filter are designated “Inclusive” (TRUE). If an output filter already exists for specified port number, an error message is displayed.

Example: **`filter-on output 2 newyork OR boston`**

List

Use the **list** NetBIOS Filtering command to display all information concerning created filters.

Syntax: list

Example: **`list`**

```
NetBIOS Filtering: Disabled

NetBIOS Filter Lists
--------------------

Handle           Type

nlist            Name
newyork          Byte

NetBIOS Filters
---------------

Port #      Direction      Filter List Handle(s)

3           Output         nlist
```

NetBIOS Filtering: Displays whether NetBIOS filtering is enabled or disabled.

NetBIOS Filter Lists	Displays the user-defined name (handle) of the configured filter lists. For type, “Name” indicates a host-name filter list and “Byte” indicates a byte filter list.
NetBIOS Filters	Displays the assigned port number and direction (input or output) of each filter. Filter List Handles displays the names of the filter lists making up the filter.

Update

Use the **update** command to add or delete information from host-name or byte filter lists. The filter-list is a string previously entered via the create byte (or name) filter-list prompt. This command brings you to the `NetBIOS Byte (or Name) filter-list Config>` prompt, which lets you perform update tasks to the specified filter list. At this prompt you can add, delete, list, or move filter-items from byte and host-name filter lists. At this prompt you can also set the default value of each filter list to Inclusive or Exclusive.

Using the add subcommand creates a filter item within the filter list. The first filter item created is assigned number 1, the next one is assigned number 2, and so on. After you enter a successful add subcommand, the router displays the number of the filter item just added.

Note: Adding more filter items to filter lists adds to processing time (due to the time it takes to evaluate each filter item in the list) and can affect performance in heavy NetBIOS traffic.

The order in which filter items are specified for a given filter list is important as this determines the way in which the filter items are applied to a packet. The first match that occurs stops the application of filter items, and the filter list is evaluated as either Inclusive or Exclusive (depending on the Inclusive or Exclusive designation of the matched filter item). If none of the filter items of a filter list produces a match, then the default condition (Inclusive or Exclusive) of the filter list is returned.

The delete subcommand specifies the number of a filter item to be deleted from the filter list. When a delete subcommand is given, any hole created in the list is filled in. For example, if filter items 1, 2, 3, and 4 exist and filter item 3 is deleted, then filter item 4 will be renumbered to 3.

The default subcommand lets you change the default setting of the filter list to either Inclusive or Exclusive. If a filter list evaluates as Inclusive, then the packet is bridged. Otherwise, the packet is filtered.

The move subcommand is available to renumber filter items within a filter list. The first argument to the move subcommand is the number of the filter list to be moved. The second argument to the move subcommand is the number of the filter list after which the first filter list should be moved.

Syntax: update byte-filter-list . . .
name-filter-list . . .

byte-filter-list *filter-list*

Updates information belonging to a byte filter-list. The filter-list parameter is a string previously entered via the **create byte-filter-list** command. This command brings you to the next `NetBIOS BYTE filter-list Config>` command

level (see example). At this level you can perform update tasks to the specified filter-list.

Example: **`update byte-filter-list newyork`**

```
NetBIOS Byte newyork Config>
```

At this prompt level you can execute several commands. Each available command is listed in the "**Update Byte-Filter** Command Options" section which follows. The correct syntax is listed followed by a description of that command and its required parameters.

name-filter-list *filter-list*

Updates information belonging to a name-filter list. This command is identical to the byte-filter-list command, except that it specifies a name-filter list rather than a byte-filter list. The filter-list parameter is a string previously entered via the create name-filter-list prompt. This command brings you to the next `NetBIOS Name filter-list Config>` command level (see example). At this level you can perform update tasks to the specified filter-list.

Example: **`update name-filter-list accounting`**

```
NetBIOS Name accounting Config>
```

At this prompt level you can execute several commands. Each available command is listed in the "**Update Name-Filter** Command Options" section which follows. The correct syntax is listed followed by a description of that command and its required parameters.

Update BYTE-Filter-List (Command Options)

This section lists the command options available for the **update byte-filter-list** command:

add inclusive *byte-offset hex-pattern <hex mask>*

Adds a filter item to the byte filter list. If the byte filter item that is added produces a match with a NetBIOS packet, the filter list it belongs to will evaluate to Inclusive (True).

- Byte-offset specifies the number of bytes (in decimal) to offset into the packet being filtered. This starts at the NetBIOS header of the packet.
- Hex-pattern is a hexadecimal number used to compare with the bytes starting at the byte-offset offset of the NetBIOS header. Syntax rules for hex-pattern include no 0x in front, a maximum of 32 numbers, and an even number of hex numbers.
- Hex-mask, if present, must be the same length as hex-pattern and is logically ANDed with the bytes in the packet starting at byte-offset before the result is compared for equality with hex-pattern. If the hex-mask argument is omitted, it is considered to be all binary 1s.

If the offset and pattern of a byte filter item represent bytes that do not exist in a NetBIOS packet (that is, if the packet is shorter than was intended when setting up a byte-filter list), then the filter item will not be applied to the packet and the packet will not be filtered. If a series of byte filter items is used to set up a single NetBIOS filter list, then a packet will not be tested for filtering if any of the byte filter items within the NetBIOS filter list represent bytes that do not exist in the NetBIOS packet.

Example: **`add inclusive`**

```
Byte Offset  [0] ?
Hex Pattern  [] ?
Hex Mask (<CR> for no mask)  [] ?
```

add exclusive *byte-offset hex-pattern <hex mask>*

Adds a filter item to the byte filter list. This command is identical to the add inclusive command, except that if the result of the comparison between the filter item and a NetBIOS packet results in a match, then the filter list evaluates to Exclusive (False). Datagram Broadcast Packets can be specified to be discarded by using this command with a byte offset of 4 and a byte pattern of 09.

- Byte-offset specifies the number of bytes (in decimal) to offset into the packet being filtered. This starts at the NetBIOS header of the packet.
- Hex-pattern is a hexadecimal number that is compared with the bytes starting at the byte-offset offset of the NetBIOS header. Syntax rules for hex-pattern include no 0x in front, a maximum of 32 numbers, and an even number of hex numbers.
- Hex-mask, if present, must be the same length as hex-pattern and is logically ANDed with the bytes in the packet starting at byte-offset before the result is compared for equality with hex-pattern. If the hex-mask argument is omitted, it is considered to be all binary 1's.

If the offset and pattern of a byte filter item represent bytes that do not exist in a NetBIOS packet (that is, if the packet is shorter than was intended when setting up a byte-filter list), then the filter item will not be applied to the packet and the packet will not be filtered. If a series of byte filter items is used to set up a single NetBIOS filter list, then a packet will not be tested for filtering if any of the byte filter items within the NetBIOS filter list represent bytes that do not exist in the NetBIOS packet.

Example: `add exclusive`

```
Byte Offset  [0] ?
Hex Pattern  [] ?
Hex Mask  (<CR> for no mask)  [] ?
```

default include

Changes the default setting of the filter list to "inclusive." This command indicates that if no filter items of the filter list match the contents of the packet being considered for filtering, the filter list will be evaluated as Inclusive. This is the default setting.

default exclude

Changes the default setting of the filter list to "exclusive." This command indicates that, if no filter items of the filter list match the contents of the packet being considered for filtering, the filter list will be evaluated as Exclusive.

delete *filter-item*

Deletes a filter item from the filter list.

- Filter-item is a decimal number representing a filter item that was previously created by the add command.

list

Displays information related to filter items in the specified filter list.

```
BYTE Filter List Name:     Engineering
BYTE Filter List Default:  Exclusive
Filter Item # Inc/Ex       Byte Offset    Pattern          Mask
1          Inclusive        14         0x123456         0xFFFF00
2          Exclusive         0         0x9876           0xFFFF
3          Exclusive        28         0x1000000        0xFF00FF00
```

move *filter-item1 filter-item2*

Reorders filter items within the filter list. The filter item whose number is specified by filter-item1 is moved and renumbered to be just after filter item2.

exit

Exits to the previous command prompt level.

Update NAME-Filter-List (Command Options)

The following section lists the command options available for the update name-filter-list command:

add inclusive *ASCII host-name <LAST-hex number>*

Adds a filter item to the host-name filter list. With this command, the host name fields of the NetBIOS packets are compared with the host-name given in this command. The following list shows how these comparisons are made:

- ADD_GROUP_NAME_QUERY: Source NetBIOS name field is examined
- ADD_NAME_QUERY: Source NetBIOS name field is examined
- DATAGRAM: Destination NetBIOS name field is examined
- NAME_QUERY: Destination NetBIOS name field is examined

If there is a match (taking into account wildcard designations in this command), then the filter list evaluates to Inclusive. If not, the next filter item of the filter list (if any) of the filter is applied to the packet. If the packet is not one of the four types supported by NetBIOS Name filtering, then the packet is bridged.

- Host-name is an ASCII string up to 16 characters long. A question mark (?) can be used in host-name to indicate a single character wildcard. An asterisk (*) can be used as the final character of host-name to indicate a wildcard for the remainder of the host-name. If host-name contains fewer than 15 characters, it is padded to the 15th character with ASCII spaces. Host-name can contain any character but the following:

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

- LAST-hex-number can be used if host-name contains fewer than 16 characters. It is a hexadecimal number (with no 0x in front of it) which indicates the value to be used for the last character. If the LAST argument is not specified on a hostname less than 16 characters, then a “?” wildcard is supplied for the 16th character.

add inclusive HEX *hexstring*

Adds a filter item to the host-name filter list. This command is functionally the same as add inclusive ASCII command. However, the representation of hostname is different. This command supplies the hostname as a series of hexadecimal numbers (with no 0x in front).

- Hexstring must consist of an even number of hexadecimal numbers. If you do not supply a full 32 hexadecimal numbers, ASCII blanks are padded to the 29th and 30th numbers and a wildcard is supplied as the 31st and 32nd (16th byte) numbers. A wildcard for a single byte can be specified by "??."

add exclusive ASCII *host-name <LAST-hex-number>*

Adds a filter item to the host-name filter list. This command is identical to the add inclusive ASCII command, except that packets that are matched against this filter item produce an Exclusive result for the filter list.

- Host-name is an ASCII string up to 16 characters long. A question mark (?) can be used in host-name to indicate a single character wildcard. An asterisk (*) can be used as the final character of host-name to indicate a wildcard for the remainder of the host-name. If host-name contains fewer than 15 characters, it is padded to the 15th character with ASCII spaces. Host-name can contain any character but the following:

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

- LAST-hex-number can be used if host-name contains fewer than 16 characters. It is a hexadecimal number (with no 0x in front of it) that indicates the value to be used for the last character. If the LAST argument is not specified on a host-name less than 16 characters, then a "?" wildcard is supplied for the 16th character.

add exclusive HEX *hexstring*

Adds a filter item to the name filter list. This command is functionally the same as the add inclusive hex command, except that packets that are matched against this filter item produce an Exclusive result for the filter list.

- Hexstring must consist of an even number of hexadecimal numbers. If you do not supply a full 32 hexadecimal numbers, ASCII blanks are padded to the 29th and 30th numbers and a wildcard is supplied as the 31st and 32nd (16th byte) numbers. A wildcard for a single byte can be specified by "??."

default include

Changes the default setting of the filter list to "inclusive." This command indicates that, if no filter items of the filter list match the contents of the packet being considered for filtering, the filter list will evaluate to Inclusive. This is the default setting.

default exclude

Changes the default setting of the filter list to "exclusive." This command indicates that, if no filter items of the filter list match the contents of the packet being considered for filtering, the filter list is evaluated as Exclusive.

delete *filter-item*

Deletes a filter item from the filter list.

- Filter-item is a decimal number representing a filter item that was previously created by the add command.

list

Displays information related to filter items in the specified filter-list.

```
NAME Filter List Name: nlist
NAME Filter List Default: Exclusive

Filter Item #   Type    Inc/Ex        Hostname       Last Char

      1         ASCII   Inclusive     EROS
      2         ASCII   Inclusive     ATHENA
      3         ASCII   Exclusive     FOOBAR
```

move *filter-item1 filter-item2*
Reorders filter items within the filter list. The filter item whose number is specified by filter-item1 is moved and renumbered to be just after filter-item2.

exit
Exits to the previous command prompt level.

Exit

Use the **exit** command to return to the previous prompt.

Syntax: exit

Example: **`exit`**

Chapter 10. Monitoring NetBIOS Filtering

This chapter summarizes and then explains the NetBIOS Filtering console commands. These commands let you monitor and display NetBIOS Filter information as an added feature to ASRT bridging. Console commands are entered at the `NetBIOS console>` prompt.

Changes you make at the `NetBIOS>` console prompt affects bridging.

Included in this chapter are the following sections:

- "Accessing the ASRT NetBIOS Filtering Console Environment"
- "NetBIOS Filtering Monitoring Commands"

Accessing the ASRT NetBIOS Filtering Console Environment

To display the `NetBIOS>` console prompt from the ASRT monitoring environment:

```
+ protocol asrt

ASRT> netbios
NetBIOS Support User Console

NetBIOS console> list filter name or byte

NetBIOS filter>
```

NetBIOS Filtering Monitoring Commands

Table 10-1 lists the NetBIOS filtering commands.

Table 10-1. NetBIOS Filtering Monitoring Commands Summary

Command	Function
List	Displays all information concerning created filters.
Exit	Exits the NetBIOS Filtering console process and returns to the previous ASRT prompt level.

? (Help)

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

Syntax: ?

Example: **?**

or

list ?

List

Use the **list** NetBIOS Filtering command to display all information concerning created filters.

Syntax: list byte-filter-lists
filters
name-filter-lists

byte-filter-lists

Displays information related to filter items in the specified byte-filter-list.

Example: list byte-filter-lists

```
BYTE Filter-List Name:  Engineering
BYTE Filter-List Default:  Exclusive

Filter Item # Inc/Ex     Byte Offset  Pattern      Mask

1             Inclusive     14        0x123456    0xFFFF00
2             Exclusive      0        0x9876      0xFFFF
3             Exclusive     28        0x1000000   0xFF00FF00
```

Filter Item#	Specifies the filter item number of the filter item. Filter items are evaluated in numerical order when determining the Inclusive/Exclusive status of the filter list.
Inc/Ex	Specifies the default status of the filter item.
Byte-offset	Specifies the number of bytes (in decimal) to offset into the packet being filtered. This starts at the NetBIOS header of the packet.
Pattern	The hexadecimal number used to compare with the bytes starting at the byte-offset of the NetBIOS header. Syntax rules for hex-pattern include no 0x in front, a maximum of 32 numbers, and an even number of hex numbers.
Mask	If present, must be the same length as hex-pattern and is logically ANDed with the bytes in the packet, starting at byte-offset, before the result is compared for equality with hex_pattern. If the hex-mask argument is omitted, it is considered to be all binary 1s.

filters

Displays information related to all configured filters.

Example: list filters

```
NetBIOS Filtering: Enabled

    Port #      Direction      Filter List Handle(s)   Pkts Filtered

       1          Input        valencia                      0
       2          Output       raleigh                       0
```

name-filter-lists

Displays information related to filter items in the specified name-filter-list.

Example: list name-filter-lists

```
NAME Filter List Name: nlist
NAME Filter List Default: Exclusive


Filter Item #   Type    Inc/Ex     Hostname   Last Char

 1              ASCII   Inclusive  EROS       <0x03>
 2              ASCII   Inclusive  ATHENA
 3              ASCII   Exclusive  FOOBAR
```

Filter Item#	Specifies the filter item number of the filter item. Filter items are evaluated in numerical order when determining the Inclusive/Exclusive status of the filter list.
Inc/Ex	Specifies the default status of the filter item.
Type	"ASCII" indicates a host-name filter item added as ASCII characters. "Hex" indicates a host name filter item added as hexadecimal numbers
Host-name	ASCII string up to 16 characters long. A question mark (?) can be used in hostname to indicate a single-character wildcard. An asterisk (*) can be used as the final character of hostname to indicate a wildcard for the remainder of the hostname. If hostname contains fewer than 15 characters, it is padded to the 15th character with ASCII spaces. Hostname can contain any character but the following: . / \ [ ] : | < > + = ; , <space>
Last char	Used if host-name contains fewer than 16 characters. It is a hexadecimal number (with no 0x in front of it) which indicates the value to be used for the last character. If the LAST argument is not specified on a hostname less than 16 characters, then a "?" wildcard is supplied for the 16th character.

Exit

Use the **exit** command to return to the previous prompt.

Syntax: exit

Example: `exit`

Chapter 11. Configuring TCP/IP Host Services

This chapter describes how to configure the TCP/IP Host Services (TCP/IP Host) protocol and how to use the TCP/IP Host configuration commands. The chapter includes the following sections:

- "Basic Configuration Procedures"
- "Accessing the TCP/IP Host Configuration Environment"
- "TCP/IP Host Configuration Commands" on page 11-2

See "TCP/IP Host Services (Bridge-Only Management)" on page 3-3 if you want to know more about why you would use TCP/IP host services.

Do not use this chapter if you are configuring the router for IP routing; instead, refer to Chapter 14, "Using and Configuring IP."

Basic Configuration Procedures

The following sections describe the basic configuration procedures for enabling TCP/IP Host Services on your IBM 8210.

Setting the IP Address

To minimally configure TCP/IP Host services, assign the IBM 8210 an IP address by using the **set ip-host** command. This IP address is associated with the IBM 8210 as a whole, instead of being associated with a single interface.

Adding a Default Gateway

The IBM 8210 uses its default gateway to communicate with hosts and gateways that are not on the bridged network to which the IBM 8210 is directly connected. The IBM 8210 can dynamically learn its default gateway using either ICMP Router Discovery (see the **enable router-discovery** command in this chapter) or RIP (see the **enable rip-listening** command in this chapter). You also can statically specify one or more default gateways by using the **add default gateway** command. The IBM 8210 uses only one default gateway at a time; any additional default gateways are used for backup.

To save the assigned IP address and default gateway information, exit from the `TCP/IP-Host config>` prompt to the `Config>` and use the **restart** command. After restarting the IBM 8210, return to the `TCP/IP-Host config>` prompt.

Enabling TCP/IP Host Services

After assigning and saving the IBM 8210 IP address and default gateway information, use the **enable services** command to enable TCP/IP Host Services.

Accessing the TCP/IP Host Configuration Environment

To access the TCP/IP Host configuration environment, enter the following command at the `Config>` prompt:

```
Config> protocol iphost
TCP/IP-Host Services user configuration
TCP/IP-Host config>
```

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

TCP/IP Host Configuration Commands

This section summarizes and explains all the TCP/IP Host configuration commands. The TCP/IP Host configuration commands allow you to specify network parameters for the TCP/IP Host bridge. Restart the router to activate the configuration commands. Enter the TCP/IP Host configuration commands at the `TCP/IP-Host config>` prompt. Table 11-1 shows the commands.

Table 11-1. TCP/IP Host Configuration Commands Summary

Command	Function
? (Help)	Lists all of the TCP/IP Host configuration commands, or lists the options associated with specific commands.
Add	Adds a default-gateway.
Delete	Deletes a default-gateway.
Disable	Disables TCP/IP Host Services, router-discovery processes, and RIP listening.
Enable	Enables TCP/IP Host Services, router-discovery processes, and RIP listening.
List	Lists the current TCP/IP Host configuration.
Set	Sets the IBM 8210's IP address.
Exit	Exits the TCP/IP Host configuration process and returns to the CONFIG environment.

? (Help)

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

Syntax: ?

Example: **?**

```
LIST
SET
ADD
DELETE
ENABLE
DISABLE
EXIT
TCP/IP-Host config>
```

Add

Use the **add** command to add default gateways (that is, routers) to your configuration.

Default gateways are used when trying to send packets to IP destinations that are off the local connection. The routing table is then built up through redirect processing. An attempt is made to detect routers that disappear. If the IBM 8210

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 IBM 8210 configuration. Enter the IP address of the default gateway you want to remove after the **delete** command.

Syntax: delete default-gateway *def-gateway-IP-address*

Example: **delete default-gateway**

```
Enter address to be deleted [0.0.0.0]?  123.45.67.89
```

Disable

Use the **disable** command to disable the following TCP/IP functions:

- TCP/IP Host Services
- Router-discovery processes
- RIP listening

Syntax: disable rip-listening
router-discovery
services

rip-listening
: Disables the building of routing table entries that have been gathered by listening to the RIP protocol. By default, RIP-listening is disabled.

  Example: **disable rip-listening**

router-discovery
: Disables the ability to learn default gateways by receiving ICMP Router Discovery messages. By default, router discovery is enabled.

  Example: **disable router-discovery**

services
: Disables the TCP/IP Host Services protocol entirely. If IP routing is not enabled, TCP/IP Host Services is enabled by default.

  Example: **disable services**

Enable

Use the **enable** command to enable the following TCP/IP functions:

- TCP/IP Host Services
- Router discovery processes
- RIP listening

Syntax: enable rip-listening
router-discovery
services

rip-listening
: Enables the building of routing table entries that have been gathered by the bridge “listening” to the RIP protocol. RIP-listening is disabled by default.

  Example: `enable rip-listening`

router-discovery
: Enables the learning of default gateways through reception of ICMP Router Discovery messages. By default, router discovery is enabled.

  Example: `enable router-discovery`

services
: Enables the TCP/IP Host Services protocol. If IP routing is not enabled, TCP/IP Host Services is enabled by default.

  Example: `enable services`

List

Use the **list** command to display information about the current TCP/IP Host configuration.

Syntax: list all

Example: `list all`

```
IP-Host IP address : 128.185.142.1
Address mask : 255.255.255.0

Default Gateway IP-address(es)
128.185.142.47

TCP/IP-Host Services Enabled.

RIP-LISTENING Disabled.

Router Discovery Enabled.
```

IP-Host IP address	Displays the current IP-Host IP address.
Address mask	Displays the current IP-Host IP subnet address mask.
Default Gateway IP-address(es)	Displays the current default gateway IP address.
TCP/IP Host Services	Displays whether TCP/IP Host Services is enabled or disabled.
RIP-LISTENING	Displays whether RIP-LISTENING is enabled or disabled.
Router Discovery	Displays whether Router Discovery is enabled or disabled.

Set

Use the **set** command to set the IBM 8210's IP address. You must assign the IBM 8210 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 IBM 8210 is a network host operating as an IP host.

Syntax: set IP-Host address *IP-host-address*

Example: **`set ip 123.45.67.89`**

```
Address mask [255.255.0.0]?
IP-Host Address set.
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Chapter 12. Monitoring TCP/IP Host Services

This chapter describes how to monitor the TCP/IP Host Services on the IBM 8210. The chapter includes the following sections:

- "Accessing the TCP/IP Host Console Environment"
- "TCP/IP Host Console Commands"

Accessing the TCP/IP Host Console Environment

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

```
+ protocol iphost
TCP/IP-Host Services user configuration
TCP/IP-Host>
```

TCP/IP Host Console Commands

This section summarizes and then explains the TCP/IP Host console commands. These commands allow you to view parameters and enter information requests from the active console. Enter these commands at the `TCP/IP-Host>` prompt. Table 12-1 shows the commands.

Table 12-1. TCP/IP Host Console Commands Summary

Command	Function
? (Help)	Lists all of the TCP/IP Host console commands, or lists the options associated with specific commands.
Dump	Displays the current IP routing table. One line is printed for each destination.
Interface	Displays the IBM 8210'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 IBM 8210.
Exit	Exits the TCP/IP Host console process and returns to the GWCON environment.

? (Help)

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

Syntax: ?

Example: ?

```
DUMP routing address
INTERFACE address
PING address
TRACEROUTE address
ROUTERS
EXIT
```

Dump

Use the **dump** command to display the current IP routing table. One line is printed for each destination. Many of the entries that are displayed are the result of ICMP redirects.

Syntax: dump

Example: dump

```
Type   Dest net         Mask      Cost  Age  Next hop(s)

Stat   0.0.0.0          00000000   0     0   128.185.142.47
Dir*   128.185.142.0    FFFFFF00   1     0   TKR/0


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

Routing table size: 768 nets (43008 bytes), 2 nets known
```

Type (route type)	Indicates how the route was derived: RIP - the route was learned through the RIP protocol. Stat - a statically configured route.
Dest net	Displays the IP address of the destination network/subnet.
Mask	Displays the IP address mask.
Cost	Displays the Route Cost.
Age	Displays the time that has elapsed since the routing table entry was last refreshed for RIP and BGP routes.
Next Hop	Displays the IP address of the next router on the path toward the destination host. Also displayed is the interface type used by the sending router to forward the packet.
Default gateway	Displays the IP address of the default gateway along with the route type, cost, age, and next-hop information associated with that entry.
Routing table size	Displays the current size (in networks and bytes) of the current table. Also identifies the number of networks (nets) known to the host.

Interface

Use the **interface** command to display the IBM 8210's IP address. When TCP/IP Host Services are running over the bridge, a single address is displayed on the console as Bridge/0.

Syntax: interface

Example: `interface`

```
Interface   IP Address(es)        Mask

  TKR/0     128.185.142.16        255.255.255.0

              or

  BDG/0     128.185.142.16        255.255.255.0
```

Interface	Displays a single address as BDG/0 when TCP/IP Host Services are running over the bridge. When services are disabled, interfaces with their corresponding numbers are displayed.
IP Address	Displays the IP address of the TCP/IP Host Services interface.
Mask	Displays the IP address subnet mask.

Ping

Use the **ping** command to make the router send ICMP Echo Requests to a given destination once a second ("pinging") and watch for a response. This command can be used to isolate trouble in an internetwork environment.

This process is done continuously, incrementing the ICMP sequence number with each additional packet. Matching received ICMP Echo responses are reported with their sequence number and the round trip time. The granularity (time resolution) of the round trip time calculation is platform specific, and usually is around 20 milliseconds.

To stop the pinging process, type any character at the console. At that time, a summary of packet loss, round trip time, and number of unreachable ICMP destinations will be displayed.

When a multicast address is given as destination, there may be multiple responses printed for each packet sent, one for each group member. Each returned response is displayed with the source address of the responder.

Note: The size of the ping (number of data bytes in the ICMP message, excluding the ICMP header) is 56 bytes, and the TTL used is 60. The size of the ping (number of data bytes in the ICMP message, excluding the ICMP header), TTL value, and frequency of pinging are all user configurable. The default values are a size of 56 bytes, a TTL of 64 seconds, and a frequency of 1 ping per second.

Syntax: ping *destination source size ttl frequency*

Example: `ping 128.185.142.11  128.185.142.06 56  60  1`

```
PING 128.185.142.11: 56 data bytes
56 bytes from 128.185.142.11: icmp_seq=0. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=1. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=2. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=3. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=4. time=0. ms
56 bytes from 128.185.142.11: icmp_seq=5. time=0. ms

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

Traceroute

Use the **traceroute** command to display the entire path to a given destination, hop by hop. For each successive hop, the traceroute command sends out three probes and prints the IP address of the responder along with the round trip time associated with the response. If a particular probe receives no response, an asterisk (*) is printed. Each line in the display relates to this set of three probes, with the leftmost number indicating the distance from the router executing the command (in router hops).

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

Syntax: traceroute *interface-address*

Example: `traceroute 128.185.142.239`

```
TRACEROUTE 128.185.142.239: 56 data bytes
   1 128.185.142.7 16 ms 0 ms 0 ms
   2 128.185.123.22 16 ms 0 ms 16 ms
   3 * * *
   4 * * *
   5 128.185.124.110 16 ms ! 0 ms ! 0 ms !
```

TRACEROUTE	Displays the destination area address and the size of the packet being sent to that address.
1	The first trace showing the destination's NSAP and the round trip time it took the packet to reach the destination and return. The packet is traced three times.
Destination unreachable	Indicates that no route to the destination is available.
*1 * * ** *2 * * *	Indicates that the router is expecting some form of response from the destination, but the destination is not responding.

When a probe receives an unexpected result (see the previous output example), several indicators can be printed. These indicators are explained in the following table.

!N	Indicates that an ICMP Destination Unreachable (net unreachable) has been received.
!H	Indicates that an ICMP Destination Unreachable (host unreachable) has been received.
!P	Indicates that an ICMP Destination Unreachable (protocol unreachable) has been received.

!	Indicates that the destination has been reached, but the reply sent by the destination has been received with a TTL of 1. This usually indicates an error in the destination, prevalent in some versions of UNIX, whereby the destination is inserting the probe's TTL in its replies. This unfortunately leads to a number of lines consisting solely of asterisks before the destination is finally reached.

Routers

Use the **routers** command to display the list of all IP routers that are known to the IBM 8210. Routers can be learned through:

- Static configuration (using the **add default-gateway** command explained on page 11-2).
- Received ICMP redirects
- ICMP Router Discovery messages (if configured)
- RIP updates (if configured)

Each router is listed with its origin, its priority (used when selecting the default route), and its lifetime (the number of seconds before the router will be declared invalid unless it is heard from again).

Syntax: routers

Example: `routers`

Exit

Use the **exit** command to exit the TCP/IP Host console process and return to the GWCON environment.

Syntax: exit

Example: `exit`

Part 2. Configuring and Monitoring Router Protocols

Chapter 13. Overview of Classical IP Over ATM

Note: See the glossary for definitions of the acronyms and terms used in this chapter. Classical IP over ATM is simply an extension of the current IP paradigm. That is, IP is independent of the medium over which it travels. An IP subnet is a group of Class A, B, or C hosts that share a common network and subnetwork portion of their IP address. The host portion of the IP address is unique for each station on the subnet. These are the characteristics of the subnet:

- Traffic from a host can be sent to any other host on the subnet.
- Traffic destined for a host outside of the subnet must pass through a gateway or router that attaches to more than one subnet.
- All members of the subnet use the same Maximum Transmission Unit (MTU) size. For classical IP, this value defaults to 9180 bytes.

Benefits of Classical IP

The Internet Engineering Task Force (IETF) has standardized a solution for sending IP traffic over an ATM interface.[1] The design described in this standard strives to keep the ATM infrastructure transparent to IP. Most applications that run today in a LAN or WAN environment will see no difference in functionality; however, their performance and throughput gains can be substantial.

In addition to the high link speeds that ATM provides, Classical IP (CIP) requires fewer framing bytes than, for example, LANs, which contain source and destination MAC addresses. Therefore, less bandwidth is used for overhead bytes, and more is used for data. In addition, no broadcast traffic is required for the resolution of ARP frames. In a broadcast environment, ARP traffic can adversely affect all stations in the subnet.

In CIP, the ARP traffic affects only the ARP Server and the client requesting the information. Other stations on the subnet are unaffected by this traffic. Even non-broadcast traffic on a shared medium such as Token-Ring or Ethernet precludes other stations from using that medium for discrete amounts of time. In CIP, independent channels are established between hosts having the conversation. These channels can be established with traffic parameters that protect the conversation from being impacted by other conversations.

The same benefits from simplifying moves, adds, and deletes that was described for ELANs apply to the CIP logical IP subnet (LIS). Membership is not based on physical location. Logically related stations are grouped in the same LIS. The ease with which a client can register to the ARP Server makes additions and changes trivial. Deletion from a subnet occurs naturally as the ARP Server ages its entries.

While all members of a LIS must support the Classical IP model, the server can route between subnets that are CIP-based and subnets that are based on LAN emulation (LE). Some equipment can be more adept at CIP, while other equipment

[1] The standard is “Classical IP and ARP over ATM,” RFC 1577, Hewlett-Packard Laboratories, Jan. 1994.

can be more adept at LANE. The flexibility of the server allows equipment to be utilized in the most effective manner.

Finally, investments in a CIP solution are protected. Enhancements to the IP over ATM work in the IETF will provide continual growth in function and performance. Distributed ARP Servers, Next Hop Routing Protocol (NHRP), Multicast Address Resolution Service (MARS), resource ReSerVation Protocol (RSVP), and other work that is being defined in the IETF will provide continual growth in functionality and performance.

Components of Classical IP

The LIS contains all of the properties of a normal IP subnet whether it is Ethernet, Token-Ring, or Frame Relay. However, because ATM is a Non-Broadcast Multiple Access (NBMA) network, the existing broadcast method for resolving addresses cannot be performed. ARP Servers and ARP Clients were developed to solve this problem.

Within the CIP model, there are two forms of requests and replies: ATMARP request and replies, which are also referred to simply as ARPs, and InATMARP request and replies. InATMARP is used determine the IP and ATM addresses of the entity at the other end of a VCC. ATMARP is used to request the ATM address associated with a particular IP address.

One ARP Server is defined per LIS. The server maintains the translation of IP addresses to ATM addresses. The server allows clients to register by accepting incoming VCCs and querying the client (with an InATMARP request) for the appropriate mapping information, which consists of the IP and ATM addresses of the client. The ARP Server also responds to ATMARP requests for ATM addresses corresponding to IP addresses specified by the client. Finally, the ARP server updates its tables by aging its ARP entries and managing incoming VCCs.

The ARP Client is the entity that always places calls. As a client is initialized, it places a call to the ARP Server, and, through the exchange of InATMARP requests and replies, registers with the ARP Server. When a client has traffic to transmit to another client on the LIS, it sends an ARP request, which contains the target IP address, to the ARP Server. If the server finds the target ATM address in its table, it sends that address back in its reply; if the server cannot find the ATM address, it sends back a negative acknowledgement (NAK) reply. When the client receives the ATM address, it then uses this address to place a call to the target client. IP datagrams then traverse this VCC.

Timeouts and Refresh

Both clients and servers age their ARP table entries. Once the timer expires, these ARP entries are deleted. If traffic is flowing when an ARP entry gets aged out, that traffic will cease until a new ARP entry is created.

To avoid an interruption in service, the server provides an automatic refresh option. This option allows the server client to transmit either an ARP to the ARP Server or an InATMARP to the target client some time before the ARP entry expires. If the target replies, then the timer of the ARP entry is reset. If the target does not reply, then the entry is deleted. The ARP Server automatically sends out an InATMARP

message before aging an entry out of its table. The server CIP clients and ARP Servers have default aging periods of 5 minutes and 20 minutes respectively. These times are configurable for each LIS.

IP Addresses and CIP components

IP addresses are key to IP routing.

Helpful Tip: When you configure the server, the act of adding an IP address to an ATM interface automatically creates a CIP client.

You must then specify whether the server is also to act as the ARP Server for the LIS. An MSS ARP Server never exists without a paired client and each LIS has one client/server pair. The server supports up to 32 LISs per ATM interface.

Creation of an IP address on the server implies packet forwarding behavior; the server forwards packets between subnets even when no routing protocol, such as Open Shortest Path First (OSPF), is configured. Furthermore, if a packet is sent to the server and the destination of the packet is not the server, but the destination is on the same subnet as the source, the server sends an Internet Control Message Protocol (ICMP) redirect message to the originator, and forwards the packet to the correct host.

ATM Addresses of CIP components

In general, ATM addresses must be unique among CIP components; however, on the server, client/server pairs share an ATM address, so that a single connection can used for both control and data traffic. The ESI and selector portions of the ATM address of a CIP component can be explicitly configured or generated automatically at run-time. The ESI defaults to the MAC address burned into the ATM interface hardware. As in LAN Emulation, you can override the default by explicitly selecting one of the locally-administered ESIs defined for the ATM interface.

Important: If only a client is being created, then explicitly configuring the ESI or selector is not recommended; however, if a client/server pair is being created, then at least the selector should be specified in order to provide the server with a fixed address that can be configured at all the clients on the LIS.

The client/server pair is created when you designate the ATM client as an ATM ARP server during configuration. When there is a client/server pair, the ARP server in that pair is local, that is, it is located in the server rather than remotely on the ATM network.

Virtual Channel Connections

The server implementation of Classical IP supports both switched virtual circuits (SVCs) and permanent virtual circuits (PVCs). SVCs require a signaling protocol to establish connections. PVCs do not require a signaling protocol, but do require configuration in both the ATM network and end systems.

SVCs can be generated automatically through the address resolution and call setup procedures of Classical IP or an SVC can be explicitly configured. Automatic SVCs

are brought up and torn down by the ARP subsystem as required for sending IP traffic. A configured SVC is brought up during initialization and kept up indefinitely.

PVCs and configured SVCs do not require an ARP Server. That is, a LIS could consist of hosts that were interconnected only by configured information. While these techniques can prove useful in small networks, the amount of manual configuration can quickly become prohibitive in larger networks.

Control channels are connections from a client to a server; data channels are connections from one client to another. The attributes of both control channels and data channels can be tailored to meet specific user needs. For example, *Quality of Service* characteristics can be specified for each LIS by configuring VCC traffic parameters such as Peak and Sustained Rates.

Key Configuration Parameters for Classical IP

Due to the simplicity of CIP, very few configuration parameters are required. The information required for a client-only configuration is:

1. IP address and Subnet mask
2. ATM address of the ARP Server

Configuration of a client/server pair requires:

1. IP address and Subnet mask
2. Answering *yes* to the question asking whether this client is also a server
3. Specifying an explicit selector for the ATM address of the server

The Maximum AAL-5 Service Data Unit (SDU) Size for CIP components does not generally need to be configured because the default of 9188 bytes is usually appropriate. However, if you need to change the SDU size, you will need to understand the relationships between the Max SDU Size for CIP components, the Max AAL-5 SDU Size for the ATM interface, and the CIP Maximum Transmission Unit (MTU) Size.

The Max AAL-5 SDU Size for CIP components can be configured for a client, but the value set for the client cannot be greater than the Max AAL-5 SDU Size for the ATM interface, which defaults to 9234 bytes. Although the Max CIP SDU Size can be configured for a client, the value set for one client can affect the MTU size for all the clients on the ATM interface. All CIP clients on the same ATM interface share a common MTU Size that is dependent on the Max SDU Sizes: the CIP MTU Size is set to “the smallest CIP Max SDU Size – 8” (CIP frames have an 8 byte header). Consequently, all LISs associated with a given ATM interface must have the same MTU. Therefore, care should be excercised when altering the Max CIP SDU Size.

Chapter 14. Using and Configuring IP

This chapter describes how to configure the Internetwork Protocol (IP) and how to use the IP configuration commands. Included are the following sections:

Basic Configuration Procedures

This section outlines the initial steps required to get the IP protocol up and running. Details about making further configuration changes are covered in other sections of this chapter. Details on individual configuration commands are covered in the command section of this chapter. The following list outlines the initial configuration tasks to bring up IP on the router. After completing these tasks, you must restart the router for the new configuration to take effect.

1. Access the IP configuration environment. (See "Accessing the IP Configuration Environment" on page 14-14.)
2. Assign IP addresses to network interfaces. (See "Assigning IP Addresses to Network Interfaces.")
3. Enable dynamic routing. (See "Enabling Dynamic Routing" on page 14-2.)
4. Add static routing information (if necessary). (See "Adding Static Routing Information" on page 14-4.)
5. Enable ARP subnet routing (if necessary). (See "Enabling ARP Subnet Routing" on page 14-6.)
6. Set up ARP parameters (if necessary). (See "Setting Up ARP Configuration" on page 14-6.)

   If RFC 1577 (Classical IP and ARP over ATM) is being used, additional ARP Server and ARP Client configuration may be required for each IP address added to this interface. This configuration is described in "ARP Over ATM Configuration Commands" on page 22-18.
7. Exit the IP configuration process.
8. Restart the router to activate the configuration changes.

The following sections discuss each configuration task in more detail.

Assigning IP Addresses to Network Interfaces

Use the IP configuration **add address** command to assign IP addresses to the network interfaces. The arguments for this command include the interface number (obtained from the `Config>` **list devices** command) and the IP address and its associated address mask.

In the following example, network interface 2 has been assigned the address 128.185.123.22 with the associated address mask 255.255.255.0 (using the third byte for subnetting).

```
IP Config> add address 2 128.185.123.22 255.255.255.0
```

The IBM 8210 allows multiple IP addresses to be configured for an interface as long as each address is for a different network/subnet. It is not valid to configure an interface with multiple IP addresses where only the host portion of the address is different.

IP allows you to use a serial line interface for IP traffic without assigning a real IP address to the line. However, you must still assign each serial line a pseudo IP address; this address is used by the router to refer to the interface but is never used externally. Use the **add address** command to assign the serial line an address of the form 0.0.0.n, where n is the interface number (again obtained from the `Config>` **list devices** command). This address format tells the router that the interface in question is an *unnumbered serial line*.

To enable IP on serial-line interface number 2 without assigning the interface an IP address, use the following command:

```
IP Config> add address 2 0.0.0.2
```

Enabling Dynamic Routing

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

OSPF, RIP, and BGP may be used over ATM if LAN emulation is selected. If RFC 1577, Classical IP and ARP over ATM (sometimes referred to native IP over ATM) is selected, only OSPF and BGP may be used. In the latter case, the ATM network is treated as a Non-Broadcast Multiple Access (NBMA) network for configuration purposes.

All routing protocols can run simultaneously. However, most routers will probably run only a single routing protocol (one of the IGPs). The OSPF protocol is recommended because of its robustness and the additional IP features (such as equal-cost multipath and variable-length subnets) that it supports.

Setting the Routing Table Size

The routing table size determines the number of entries in the routing table from all sources, including dynamic routing protocols and static routes. The default size is 768 entries.

To change the size of the routing table, use the **set routing table-size** configuration command. Setting the routing table size too small results in routes being discarded. Setting it too large results in inefficient use of memory resources. After operation, use the console **dump** command to view the contents of the table and then adjust the size as necessary, allowing some room for expansion.

Enabling the OSPF Protocol

OSPF configuration is done via its own configuration console (entered via the Config> **protocol ospf** command). To enable OSPF, use the following command:

```
OSPF Config> enable OSPF
```

After enabling the OSPF protocol, you are prompted for size estimates for the OSPF link state database. This gives the router some idea how much memory must be reserved for OSPF. You must supply the following two values that will be used to estimate the size of the OSPF link state database:

- Total number of external routes imported into the OSPF routing domain.
- Total number of OSPF routers in the routing domain.

Enter these values at the following prompts (sample values have been provided):

```
OSPF Config> enable ospf
Estimated # external routes[0]? 200
Estimated # OSPF routers [50]? 60
```

Next, configure each IP interface that is to participate in OSPF routing. To configure an IP interface for OSPF, use the following command:

```
OSPF Config> set interface
```

You are prompted to enter a series of operating parameters. Each interface is assigned a cost as well as other OSPF operating parameters.

When running other IP routing protocols besides OSPF, you may want to enable the exchange of routes between OSPF and the other protocols. To do this, use the following command:

```
OSPF Config> enable AS-boundary-routing
```

For more information on the OSPF configuration process, see Chapter 16, “Using and Configuring OSPF” on page 16-1.

Enabling the RIP Protocol

This section describes how to initially configure the RIP protocol. When configuring the RIP protocol, you can specify which set of routes the router will advertise and/or accept on each IP interface.

| With an ATM network, RIP will work properly only if LAN Emulation is configured.
| For 1577 clients, use OSPF instead of RIP for an IGP.

First, enable the RIP protocol with the following command:

```
IP Config> enable RIP
```

When RIP is enabled, the following default behavior is established:

- The router includes all network and subnet routes in RIP updates sent out on each of its configured IP interfaces. It does not include default and static routes.
- The router processes all RIP updates received on each of its configured IP interfaces.
- RIP will not override default and static routes.

To change any of the default sending/receiving behaviors, use the following IP configuration commands, which are defined on a per-IP-interface basis.

```
IP Config> enable/disable sending net-routes
IP Config> enable/disable sending subnet-routes
IP Config> enable/disable sending static-routes
IP Config> enable/disable sending host-routes
IP Config> enable/disable sending default-routes
IP Config> enable/disable receiving rip
IP Config> enable/disable receiving dynamic nets
IP Config> enable/disable receiving dynamic subnets
IP Config> enable/disable receiving host-routes
IP Config> enable/disable override default
IP Config> enable/disable override static-routes
```

Enabling the BGP Protocol

The BGP protocol is enabled from its own configuration prompt, `BGP Config>` For more information about configuring BGP, refer to the discussion on using and configuring BGP4 in *Protocol Configuration and Monitoring Reference Volume 2*.

Adding Static Routing Information

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

Static routing information consists of any of the following items:

- **Default Gateway.** Packets are routed to default (authoritative) gateways when the packet destination cannot be found in the routing table.
- **Default Subnet Gateways.** If you are using subnetted networks, you can define a separate default gateway for each subnetted network.
- **Static Network/Subnet/Host Routes.** For each destination that is to have a fixed route, configure the next hop and distance to the destination.

Default Gateway

Routers send packets having unknown destinations (that is, destinations not present in the routing table) toward the default gateway. A default gateway is configured in the router by specifying the next hop to use to get to the default gateway and the cost of sending packets to the default gateway.

In the following example, the next hop toward the default gateway is 192.9.1.4 and the cost of sending a packet to the default gateway is 5.

```
IP Config> set default network-gateway
Default gateway [0.0.0.0]? 192.9.1.4
gateway's cost [0]? 5
```

Default gateways can be learned and advertised by both the OSPF and RIP protocol. For the OSPF protocol, a router can be configured to advertise itself as the default gateway with the following OSPF command:

```
OSPF Config> enable/disable AS-boundary-routing
```

The RIP protocol can be configured so that it will advertise knowledge of the default gateway (if it has any) to its neighbors. RIP can also be configured so that a learned default gateway will (or will not) override a statically configured default gateway. These configuration tasks are accomplished with the following two commands:

```
IP Config> enable/disable sending default-routes
IP Config> enable/disable override default
```

Finally, a router that runs BGP can be configured to advertise itself (via the OSPF and RIP protocol) as the default gateway whenever it has BGP-learned routes in its routing tables. For OSPF, this is accomplished through the OSPF **enable/disable AS-boundary-routing** command. For RIP, the following commands are used:

```
IP Config> set originate-RIP-default
```

Default Subnet Gateways

There can be a default subnet gateway configured for each subnetted network that the router knows about. When the router attempts to forward a packet to a destination belonging to the subnetted network, but that destination cannot be found in the routing table, the packet is forwarded instead to the default subnet gateway.

Configuring default subnet gateways is the same as configuring the above default network gateway. The only difference is that you must specify the subnetted network on the command line. For example, to create a default subnet gateway for the subnetted network 18.0.0.0, you could use the following command:

```
IP Config> set default subnet-gateway
For which subnetted network [0.0.0.0]? 18.0.0.0
Default gateway [0.0.0.0]? 128.185.123.22
gateway's cost [0]? 2
```

This example specifies that the next hop to the subnet default gateway is 128.185.123.22, and that the cost of routing a packet to the default subnet gateway is 2.

Static Network/Subnet/Host Routes

Configure static routes for those destinations that cannot be discovered by the dynamic routing protocols, or to establish permanent or temporary main or backup routes. The destination is described by an IP network/subnet/host number (**dest-addr**) and the destination's address mask (**mask**). For host routes, the mask is always 255.255.255.255. The route to the destination is described by the IP address of the first hop router to use (**1st-hop**) and the cost of routing a packet to the destination (**cost**). To create, modify, or delete a static route, use the commands:

```
IP Config> add route dest-addr mask 1st-hop cost
IP Config> change route dest-addr mask new-mask 1st-hop cost
IP Config> delete route dest-addr mask
```

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

Routes dynamically learned through the OSPF and RIP protocols can override static routes. For the RIP protocol, you can disable this override behavior. See the RIP section of this chapter concerning the **enable/disable override static-routes** commands.

You can configure both OSPF and RIP to advertise configured static routes over interfaces where these dynamic protocols are enabled.

To configure RIP to advertise static routes, enter the following command at the IP Config> prompt:

```
IP Config> enable sending static-routes ip-interface-address
```

To configure OSPF to advertise static routes, enter the following command at the OSPF Config> prompt:

```
OSPF Config> enable as boundary
Import static routes? yes
```

Setting Up ARP Configuration

The Address Resolution Protocol (ARP) is used to map protocol addresses to hardware addresses before a packet is forwarded by the router. ARP is always active on the router, so you do not need to do any additional configuration to enable it with its default characteristics. However, if you need to alter any ARP configuration parameters (such as **enable auto-refresh** or **set refresh-timer** which changes the default refresh timer), or if you need to add, change, or delete permanent address mappings, see Chapter 22, “Using and Configuring ARP.”

If LAN Emulation is configured on an interface, the defaults apply. You can effectively use the ARP protocol without any changes. If RFC 1577 (Classical IP and ARP over ATM) is used, additional configuration for ARP Clients and ARP Servers is required for each IP address configured on that ATM interface (as described in “ARP Over ATM Configuration Commands” on page 22-18).

Enabling ARP Subnet Routing

If there are hosts on attached subnetted networks that do not support IP subnetting, use Address Resolution Protocol (ARP) subnet routing (described in RFC 1027). When the router is configured for ARP subnet routing, it will reply by proxy to ARP requests for destination (that is, off the LAN if the router is itself the best route to the destination, and the destination is in the same natural network as the source). For proper operation, all routers attached to a LAN containing subnetting-ignorant hosts should be configured for ARP subnet routing.

To enable ARP subnet routing, use the following command:

```
IP Config> enable ARP-subnet-routing
```

Enabling ARP Network Routing

Some IP hosts ARP for all destinations, whether or not the destination is in the same natural network as the source. For these hosts, ARP subnet routing is not enough, and the router can be configured to reply by proxy to any ARP request as long as the destination is reachable through the router and the destination is not on the same local network segment as the source.

To enable ARP network routing, use the following command:

```
IP Config> enable arp-network-routing
```

IP Filtering

Filtering is a process by which the user specifies certain criteria that the router uses to control packet forwarding. The following two main types of filtering are provided to help users achieve their security and administrative goals:

- Access control
- Route filtering

Access Control

Access control allows the IP router to control the processing of individual packets based on source and destination IP addresses, IP protocol number, and by destination port number for the TCP and UDP protocols. This can control access to particular sets of IP hosts and services.

You can define access controls by configuring access control lists. One global list and two lists per interface can be specified. The global list applies to the router as a whole. Interface lists, also known as packet-filters, are assigned names and only apply to the designated interface. For each interface, one list applies to incoming packets, and the other applies to outgoing packets. The lists are applied independently of each other. A packet might *pass* an incoming interface list, and be *dropped* by the global list.

Figure 14-1 illustrates the series of searches a packet must pass before being forwarded (routed):

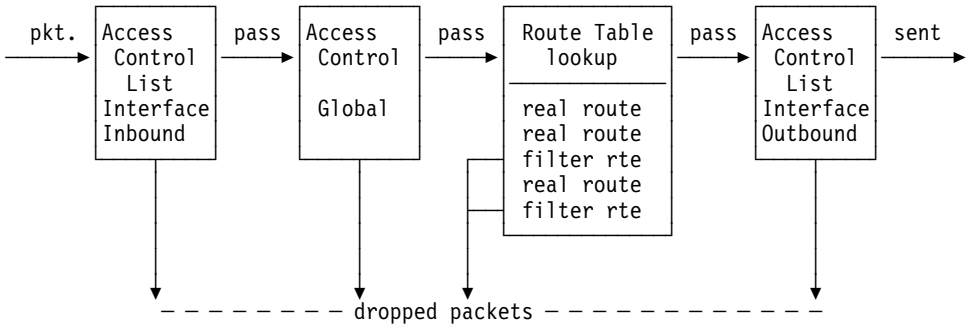


Figure 14-1. Access Control - Searching a Packet for Forwarding

Each access control list consists of one or more access control records that set the filtering criteria.

Access Control Records

Each record in a list may be inclusive or exclusive. Source and destination IP addresses and masks are required for every record and IP protocol number ranges and destination port ranges can also be specified. As IP packets flow through the router, IP headers are compared to access control list records. A packet matches a record if every specified field in the record matches a corresponding field in the packet's IP header. If a packet matches a record, and the record is inclusive, the packet *passes*. If the record is exclusive, the packet is *dropped* and is not processed any further by the router. If no records match after going through the entire list, the packet is also dropped. When defining records in access control lists, it is important to remember the following:

The order of records in a list is important. Configuration commands are provided to change the order of records in a list.

- For every list that includes at least one access control record, an inclusion record must exist for any packets to pass the list. One method of allowing all packets that do not match any of the specified records in a list to pass is to include the following wildcard record as the last record in the list:

```
add access-control inclusive 0.0.0.0  0.0.0.0
```

Source and Destination IP Addresses

Each record has an IP address and mask pair for both the source and destination IP addresses. When an IP packet is compared against an access control record, the IP address in the packet is “and-ed” with the mask in the record, and the result compared with the address in the record. For example, a source address of 26.0.0.0 with a mask of 255.0.0.0 in an access control record will match any IP packet source address with 26 in the first byte. A destination address of 192.67.67.20 and a mask of 255.255.255.255 will only match IP packet destination host address 192.67.67.20. An address of 0.0.0.0 with mask 0.0.0.0 is a wildcard, and matches any IP address.

Protocol Number

Each record can also have an IP protocol number range. This range is compared to the protocol byte in the IP header; a protocol value within the specified range (inclusive) will match. If you specify a range of 0 to 255, any protocol will match. Commonly used protocol numbers are 1 (ICMP), 6 (TCP), 17 (UDP), and 89 (OSPF).

Port Number

TCP/UDP port number ranges can also be specified in an access control record. This range is compared to the port number field in the TCP or UDP header of the IP packet; a port number value within the specified range (inclusive) will match. This field is ignored for IP packets that are not TCP or UDP packets. If you specify a range of 0 to 65535, any port number will match. Commonly used port numbers are 21 (FTP), 23 (Telnet), 25 (SMTP), 513 (rlogin) and 520 (RIP). See RFC 1700 (Assigned Numbers) for a list of IP protocol and port numbers.

Examples

The following example allows any host to send packets to the SMTP TCP socket on 192.67.67.20.

```
add access-control  inclusive 0.0.0.0  192.67.67.20  255.255.255.255 6 6 25 25
```

The next example prevents any host on subnet 1 of Class B network 150.150.0.0 from sending packets to hosts on subnet 2 of Class B network 150.150.0.0 (assuming a 1-byte subnet mask).

```
add access-control exclusive 150.150.1.0 255.255.255.0 150.150.2.0 255.255.255.0 0 255 0 65535
```

This command allows the router to send and receive all RIP packets.

```
add access-control inclusive 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 17 17 520 520
```

Enabling Access Control

IP Access Control (including global and interface access control) is enabled with the **set access-control on** command, and disabled with the **set access-control off** command.

If IP access control is enabled, you must be careful with packets that the router originates and receives. Be sure not to filter out the RIP or OSPF packets being sent or received by the router. The easiest way to do this is to add a wildcard inclusive entry as the last in the access control list. Alternately, you can add specific entries for RIP and OSPF, perhaps with restrictive addresses and masks. Note that some OSPF packets are sent to the Class D multicast addresses 224.0.0.5 and 224.0.0.6, which is important if address checking is being done for routing protocols. See the **add** command for more information on access control.

Defining the Global Access Control List

The global access control list is defined when records are added at the `IP Config>` prompt:

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

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

Defining Interface Specific Access Control Lists (Packet Filters)

To define interface specific access control lists, use the **add packet-filter** command at the `IP Config>` prompt. The router prompts you for the filter name, direction (input or output), and the interface number to which it applies.

```
Packet-filter name [ ]? test
Filter incoming or outgoing traffic? [IN]? in
Which interface is this filter for [0]? 1
```

You can use the **list packet-filter** command to list all interface specific access control lists configured in the router.

Setting Up Access Control Records for Interface Specific Access Control Lists

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

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

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

Route Filtering

Route filtering impacts packet forwarding by influencing the content of the routing table. In general, route filtering is more efficient but less flexible than access control. Filtering based on source IP address, IP protocol, and TCP/UDP port number can only be done using access control, described above. Route filtering is *not* recommended when OSPF is used in your network; OSPF-learned internal routes will override filtered routes in the routing table.

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

- Filter routes
- RIP input filters

Defining a Filter Route

You can designate an IP destination to be inserted in the routing table as a *filter route*. IP packets will not be forwarded to these destinations, and routing information concerning them will not be advertised.

To configure a filter route, enter the following command at the `IP Config>` prompt:

```
IP Config>  add filter  dest-IP-address  address-mask
```

Filter routes will be listed as an entry with the type *fltr* when the **dump** command is used to view the IP routing table.

Note: If a more specific route is available, packets will be forwarded. For example, if a filter route is defined for network 9.0.0.0 (mask 255.0.0.0), but a route is learned for a subnet of the network (for example 9.1.0.0, mask 255.255.0.0), then packets will be forwarded to subnet 9.1.0.0 but not to other subnets of that network.

Defining RIP Input Filters

When RIP is used as the dynamic routing protocol, you can configure certain interfaces to ignore routes in RIP updates.

The following command results in ignoring all RIP updates received on an interface:

```
IP Config>  disable receiving rip  ip-interface-address
```

The following commands result in ignoring certain types of routes received on an interface:

```
IP Config> disable receiving dynamic nets ip-interface-address
IP Config> disable receiving dynamic subnets ip-interface-address
IP Config> disable receiving dynamic host ip-interface-address
```

When the latter group of commands are used, you can allow specific routes to be accepted using the following command:

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

Configuring the BOOTP/DHCP Forwarding Process

BOOTP (documented in RFC 951 and RFC 1542) is a bootstrap protocol used by a diskless workstation to learn its IP address, the location of its boot file, and the boot server name. Dynamic Host Configuration Protocol (DHCP), documented in RFC 1541, is used to allocate reusable network addresses and host-specific configuration parameters from a server.

The following terms are useful when discussing the BOOTP/DHCP forwarding process:

- *Client* - the workstation requiring BOOTP/DHCP services.
- *Servers* - the boot host (with UNIX daemon bootpd, DOS version available from FTP software, or OS/2) or other BOOTP/DHCP server that is providing these services. This router does not provide server support.
- *BOOTP relay agent* or *BOOTP forwarder* - a device which forwards requests/replies exchanged by the Client and Server. This router supports the relay agent function.

The following steps outline an example of the BOOTP forwarding process. (DHCP exchanges proceed in a similar way):

1. The Client copies its Ethernet address (or appropriate MAC address) into a BOOTP packet and broadcasts it onto the local LAN. BOOTP is running on top of UDP.
2. The local BOOTP relay agent receives the packet and checks to see if the packet is well formatted and that the maximum number of application hops has not expired. It also checks to see if the client has been trying long enough.

   Note: If multiple hops are required before reaching the BOOTP agent, the packet is routed normally via IP. All other routers would not examine the packet to determine whether it is a BOOTP packet.
3. The Local BOOTP agent forwards a separate BOOTP request to each of its configured servers. The BOOTP request is the same as the one that was initially sent by the client except that it has a new IP header with the relay agent's IP address copied into the body of the BOOTP request.
4. The server receives the request and looks up the client's hardware (for example, Ethernet) address in its database. If found, it formats a BOOTP reply containing the client's IP address, the location of its boot file, and the boot server name. The reply is then sent to the BOOTP relay agent.
5. The BOOTP relay agent receives the reply and makes an entry in its ARP table for the client and then forwards the reply to the client.
6. The client then continues to boot using TFTP, using the information in the BOOTP reply packet.

Enabling/Disabling BOOTP Forwarding

To enable or disable BOOTP forwarding on the router, enter the following command at the IP configuration prompt. (Enable BOOTP Forwarding to allow the router to forward BOOTP and/or DHCP requests and replies between Clients and Servers on different segments of your network.)

```
IP Config> enable/disable bootp
```

When enabling BOOTP, you are prompted for the following values:

- Maximum number of application hops you want the BOOTP request to go. This is the maximum number of BOOTP relay agents that can forward the packet. This is NOT the maximum number of IP hops to the Server. A typical value for this parameter is 1.

- Number of seconds you want the Client to retry before the BOOTP request is forwarded. *This parameter is not commonly used.* A typical value for this parameter is 0.

After accepting a BOOTP request, the router forwards the BOOTP request to each BOOTP server. If there are multiple servers configured for BOOTP, the router replicates the packet.

Configuring a BOOTP/DHCP Server

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

```
IP Config> add BOOTP-SERVER [IP address of server]
```

Multiple servers can be configured. In addition, if only the network number of the server is known or if multiple servers reside on the same network segment, a broadcast address can be configured for the server.

IP Multicast Support

IP multicast is an extension of LAN multicasting to a TCP/IP Internet. It is the ability of an IP host to send a single datagram (called IP multicast datagram) that will be delivered to multiple destinations. IP multicast datagrams are identified as those packets whose destinations are class D IP addresses (that is, whose first byte lies in the range 224-239 inclusive). Each class D address defines a multicast group.

The extensions required of an IP host to participate in IP multicasting are specified in RFC 1112 (Host Extensions for IP Multicasting.) That document defines a protocol, the Internet Group Management Protocol (IGMP), that enables hosts to dynamically join and leave multicast groups. This router implements the IGMP protocol functions that enable it to keep track of IP group membership on its local physical and on its emulated LANs by sending IGMP Host Membership Queries and receiving IGMP Host Membership Reports.

A router must also be able to route IP multicast datagrams between the source and (multiple) destination hosts. This router supports the Multicast Open Shortest Path First (MOSPF) protocol as defined by RFC 1584 (Multicast Extensions to OSPF), and the Distance Vector Multicast Routing Protocol (DVMRP).

A MOSPF router distributes group location information throughout the routing domain by flooding a new type of link state advertisement, the group-membership-LSA (type 6). This in turn enables the MOSPF routers to most efficiently forward a multicast datagram to its multiple destinations: each router calculates the path of the multicast datagram as a tree whose root is the datagram source, and whose terminal branches are LANs containing group members. For more information, see “Multicast OSPF” on page 16-3.

DVMRP is a multicast routing protocol derived from the Routing Informatin Protocol (RIP). This router provides support for DVMRP so that you can exchange multicast routing information with other routing entities that do not support MOSPF. This router's DVMRP implementation also allows tunneling of DVMRP information over an MOSPF-capable network and over a non-multicast-capable IP network.

This router also allows you to “enroll” the router itself as a member of one or more multicast groups. As a member of a multicast group, the router will respond to “pings” and SNMP queries addressed to the group address (one command could be used to query multiple routers).

Configuring the router for IP Multicast

To enable the router to track IP multicast group memberships and forward multicast datagrams, you must enable MOSPF.

Enabling MOSPF: To enable MOSPF, you must first enable OSPF (see “Enabling the OSPF Protocol” on page 14-3) and then do the following:

1. Enable multicast forwarding by entering the following command at the `OSPF Config>` prompt:

   ```
   OSPF Config> enable multicast-routing
   ```

2. Set the following parameters when configuring each OSPF interface through the `OSPF Config>` **set interface** command:

   ```
   Forward multicast datagrams (Yes or No)?
   Forward as datalink unicasts (Yes or No)?
   IGMP polling interval (in seconds) [60]?
   IGMP timeout (in seconds) [180]?
   ```

For more details on these configuration commands, see “Multicast OSPF” on page 16-3.

The following OSPF monitoring commands are used to obtain information about IP multicast groups and MOSPF routing:

```
OSPF> advertisement ls-type link-state-id
      (by entering type 6 in the ls-type field, this
       command displays the contents of the group-membership
       link-state-advertisement sent out by the router)

OSPF> interface interface-ip-address
       (a number of output fields are specific to multicasting
        and provide statistics on IGMP and multicast packets)

OSPF> Mcache
       (displays list of currently active multicast cache
        entries

OSPF> Mgroups
      (displays group membership of the router's attached
       interfaces as reported via IGMP)

OSPF> Mstat
      (displays various multicast packet statistics)
```

For more information on these commands, refer to “Multicast OSPF” on page 16-3.

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.

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.

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 an MSS Server 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 *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.
3. Select an MSS to have the primary gateway. This MSS must have a LEC interface on the ELAN of the end station.
4. Select an MSS or set of MSSs to have the backup gateway. This MSS or set of MSSs must have a LEC interface on the ELAN of the end station.
5. Config a redundant gateway on each MSS using the “Add” option for IP.

Accessing the IP Configuration Environment

To access the IP configuration environment, enter the following command at the `Config>` prompt:

```
Config> Protocol IP
Internet protocol user configuration
IP Config>
```

IP Configuration Commands

This section summarizes and then explains all IP configuration commands. These commands allow you to modify the IP protocol behavior to meet your specific requirements. Some amount of configuration is necessary to produce a fully functional IP router. Enter IP configuration commands at the `IP config>` prompt.

Table 14-1. IP Configuration Commands Summary

Command	Function
? (Help)	Lists the configuration commands or lists the actions associated with specific commands.
Add	Adds to the IP configuration information. Interface addresses can be added, along with access controls, filters, and packet-filters.
Change	Modifies information that was originally entered with the **add** command.
Delete	Deletes IP configuration information that had been entered with the **add** command.
Disable	Disables certain IP features that have been turned on by the **enable** command.
Enable	Enables IP features such as ARP subnet routing, UDP Forwarding, originate default, directed broadcasts, BOOTP, and the various RIP flags controlling the sending and receiving of RIP information.
List	Displays IP configuration items.
Move	Changes the order of access control records.
Set	Establishes IP configuration modes such as the use of access control and the format of broadcast addresses. Also sets IP parameters such as default routers, TTL (time-to-live) of packets originated by the router, and the size of the IP routing table.
Update	Used to assign access control entries.
Exit	Exits the IP configuration process.

? (Help)

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

Syntax: ?

Example: ?

```
ADD
CHANGE
DELETE
DISABLE
ENABLE
LIST
MOVE
SET
UPDATE
EXIT
```

Example: add ?

```
accept-rip-route
access-control
address
bootp-server
filter
packet-filter
REDUNDANT Default Gateway
route
udp-destination
```

Add

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

Syntax: add
- accept-rip-route . . .
- access-control . . .
- address . . .
- bootp-server
- filter . . .
- packet-filter
- REDUNDANT Default Gateway
- route . . .

accept-rip-route *IP-network/subnet*

Allows an interface to accept a RIP route when input RIP filtering is enabled for an interface. You can print the list of networks/subnets that have already been entered using the **list rip-routes-accept** command. You can enable the input filtering of RIP routes on a per-IP-interface basis. This is done separately for network-level routes (for example, a route to 10.0.0.0) for subnet-level routes (for example, a route to 128.185.0.0), and for host-level routes (for example 128.185.123.28). To enable input filtering of routes on an IP interface, use the **disable dynamic nets/subnets/host** commands.

IP network/subnet

Valid Values: any valid IP address

Default Value: none

Example: **`add accept-rip-route 10.0.0.1`**

or

Example: **`add accept-rip-route`**

```
Network number [0.0.0.0]? 10.0.0.0
```

access-control *type IP-source source-mask IP-dest dest-mask*

[first-protocol last-protocol] [first-port last-port]

Adds an access control record to the end of the global access control list. This allows you to describe a class of packets to forward or drop, depending on the type of the record. The length and order of the IP access control list can affect the performance of the IP forwarder. Each record must be assigned the following: type, IP source, source-mask, IP destination, and destination-mask fields. The type must either be inclusive or exclusive. The *IP-source* and *IP-dest* fields are in the form of IP addresses in dotted decimal notation. Optionally, you may specify an IP protocol number range with the *first-protocol* and *last-protocol* fields, which are an inclusive range of IP protocols that match this entry. You also may specify a TCP or UDP port number or port number range that matches an entry, where “port number range” is an inclusive range of TCP and UDP ports that matches this entry. Specify TCP or UDP in the protocol fields, then specify the port number range in the first-port and last-port fields.

type Indicates whether packets are sent or dropped for a specific address or set of addresses.

Specify *Include* to cause the router to receive a packet and to forward it if it matches criteria in the remaining arguments.

Specify *Exclude* to cause the router to discard the packets.

IP-source

Valid Values: any valid IP address

Default Value: none

source-mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: none

IP-dest

Valid Values: any valid IP address

Default Value: none

dest-mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: none

first-protocol

The lower boundary of a range of IP protocol numbers.

Some commonly used protocol numbers are:

- 1 for ICMP
- 6 for TCP
- 17 for UDP
- 89 for OSPF

Valid Values: 0 to 255

Default Value: 0

last-protocol

The upper boundary of a range of IP protocol numbers.

Some commonly used protocol numbers are:

- 1 for ICMP
- 6 for TCP
- 17 for UDP
- 89 for OSPF

Valid Values: 0 to 255

Default Value: 0

first-port

The lower boundary of an IP TCP/UDP port number range.

Some commonly used port numbers are:

- 21 for FTP
- 23 for Telnet
- 25 for SMTP
- 513 for rlogin

- 520 for RIP

Valid Values: a port number in the range of 0 - 65535

Default Value: 0

last-port

The upper boundary of an IP TCP/UDP port number range.

Some commonly used port numbers are:

- 21 for FTP
- 23 for Telnet
- 25 for SMTP
- 513 for rlogin
- 520 for RIP

Valid Values: a port number in the range of 0 - 65535

Default Value: 0

Example: **`add access-control inclusive`**

```
Internet source [0.0.0.0]?
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]?
Destination mask [255.255.255.255]?
Enter starting protocol number ([CR] for all) [-1]?
IP config>
```

address *interface-number IP-address address-mask*

Assigns an IP address to one of the router's hardware network interfaces. A hardware network interface will not receive or transmit IP packets until it has at least one IP address. You must specify an IP address together with its subnet mask. For example, if the address is on a class B network, using the third byte for subnetting, the mask would be 255.255.255.0. Use the **list devices** command to obtain the appropriate command interface-number. Serial lines do not need addresses. Such lines are called unnumbered. However, you must still enable them for IP traffic using the **add address** command. The address then used is 0.0.0.n, where n is the *interface-number*.

Valid Values: For non-serial line interfaces:

- The class A range is 1.0.0.1 through 126.255.255.254
- The class B range is 128.0.0.1 through 191.255.255.254
- The class C range is 192.0.0.1 through 223.255.255.254

For serial line interfaces:

- 0.0.0.n, where n is the hardware interface number.

You must specify an IP address together with its subnet mask. For example, if the address is on a class B network, using the third byte for subnetting, the mask would be 255.255.255.0. Use the **List Devices** option to obtain the appropriate option interface-number.

interface-number

Valid Values: a number in the range of 1 - 7

Default Value: none

IP-address

Valid Values: any valid IP address

Default Value: none

address mask

Valid Values: 0.0.0.0 - 255.255.255.255

Default Value: none

Example: `add address 0 128.185.123.22 255.255.255.0`

bootp-server *server-IP-address*

Adds a BOOTP/DHCP server to a network configuration. Acting as a bootp relay agent, your router accepts and forwards BOOTP/DHCP requests to the BOOTP/DHCP server. BOOTP is a bootstrap protocol used by a router or a diskless workstation to learn its IP address, the location of its boot file, and the boot server name. DHCP is Dynamic Host Configuration Protocol, used to configure a host over a network connection.

server-IP-address

Valid Values: any valid Bootp server IP address

Default Value: none

Example: `add bootp-server 128.185.123.22`

filter *dest-IP-address address-mask*

Designates an IP destination to be filtered. IP packets will not be forwarded to filtered destinations, nor will routing information be disseminated concerning such destinations. Packets to filtered destinations are simply discarded. You must specify a filtered destination as an IP address with its subnet mask. For example, to filter a subnet of a class B network, using the third byte for subnetting, the mask would be 255.255.255.0. Using the filter mechanism is more efficient than IP access controls, although not as flexible. Filters also affect the operation of the IP routing protocols, unlike access controls. Filtered networks/subnets are overridden if learned using the OSPF routing protocol.

The effect of this command is immediate; you do not have to reboot the router for it to take effect.

dest-IP-address

Valid Values: any valid IP address

Default Value: none

address mask.

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: 0.0.0.0

Example: `add filter 127.0.0.0 255.0.0.0`

packet-filter *filter-name type intf#*

Defines a packet filter record within the router configuration.

filter-name

Valid Values: any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: none

type *IN* filters incoming traffic.

OUT filters outgoing traffic.

intf# **Valid Values**: any defined interface

Default Value: none

Example: `add packet-filter`

```
Packet-filter name [ ]? filt-1-0
Filter incoming or outgoing traffic? [ IN]?
Which interface is this filter for [0]? 1
```

redundant *redundant-IP-gateway IP-address Address-mask MAC-address Primary/backup*

Adds a Redundant Default Gateway IP address to your configuration.

Redundant IP Gateway

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

Gateway IP Address Mask

Specifies the mask of the IP address.

Valid Values: Any valid IP net mask.

Default Value: 0.0.0.0

MAC Address

Valid Values: Any valid MAC addresses not used by other interfaces on the ELAN.

Default Value: 00.00.00.00.00.00

Gateway Type

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

This query asks whether the gateway on this MSS 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 *IP-network/subnet/host IP-mask next-hop cost*

Adds a static network/subnet/host route to the router's IP configuration. When dynamic routing information is not available for a particular destination, static routes are used.

The destination is specified by an IP address (*IP-network/subnet/host*) together with an address mask (*IP-mask*). If the destination IP address is a network address, then the IP mask must be a network mask. If the destination IP address is a subnet address, then the IP mask must be a subnet mask. Finally, if the destination IP address is a host address, then the IP mask must be a host mask (which means that the only valid value is 255.255.255.255.) The IP-mask must be accurate; if it is not, the static route will not be accepted.

The route to the destination is specified by the IP address of the next hop (*next-hop*), and the cost (*cost*) of routing the packet to the destination. The next hop must be on the same (sub)net as one of the router's directly connected interfaces. Static routes are always overridden by routes learned through OSPF. By default, static routes are also overridden by routes learned through RIP; however, you can change that with the **enable/disable override static-routes** command.

The effect of this command is immediate; you do not have to reboot the router for it to take effect.

IP-network/subnet/host

Valid Values: any valid IP address

Default Value: none

IP-mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: none

next-hop

Valid Values: any valid IP address

Default Value: none

cost **Valid Values**: an integer in the range of 1 - 16

Default Value: 1

Examples: `add route 17.0.0.0 255.0.0.0 128.185.123.22 6`

Class A (network example):
`add route 9.0.0.0 255.0.0.0 17.102.23.1 4`

Class A (subnet example):
`add route 9.67.0.0 255.255.0.0 17.102.23.1 4`

Class B (subnet example):
`add route 192.3.2.32 255.255.255.224 17.102.23.1 4`

Class B (host example):
`add route 167.59.34.67 255.255.255.255 17.102.23.1 4`

Change

Use the **change** command to change an IP configuration item previously installed by the **add** command. In general, you must specify the item you want to change, just as you specified the item with the **add** command.

Syntax: change access-control . . .
address . . .
route . . .

access-control *record-number type IP-source source-mask IP-dest dest-mask [first-protocol last-protocol] [first-port last-port]*

Modifies an existing global access-control record. Use the **list access-control** command to view all existing records and obtain the record number.

Example: **`change access-control 2`**

```
Enter type [E]? i
Internet source [1.1.1.1]?
Source mask [255.255.255.255]?
Internet destination [2.2.2.2]?
Destination mask [255.255.255.255]?
Enter starting protocol number [6]?
Enter ending protocol number [6]?
Enter starting port number [23]?
Enter ending port number [23]?
```

address *old-address new-address new-mask*

Modifies one of the router's IP interface addresses. You must specify each new address together with the new address' subnet mask. This command can also be used to change an existing address' subnet mask.

For non-serial line interfaces:

- The class A range is 1.0.0.1 through 126.255.255.254
- The class B range is 128.0.0.1 through 191.255.255.254
- The class C range is 192.0.0.1 through 223.255.255.254

For serial line interfaces:

- 0.0.0.n, where n is the hardware interface number.

Use these address guidelines to:

1. Enter the *old-address*.
2. Enter the *new-address*.
3. Enter the new *new-mask*.

Example: **`change address 192.9.1.1 128.185.123.22 255.255.255.0`**

route *destination mask new-1st-hop new-cost*

Modifies either the next hop or the cost associated with a configured static network/subnet route. The effect of this command is immediate; you do not have to reboot the router for it to take effect.

destination

Valid Values: any valid IP address

Default Value: none

mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: none

new-1st-hop

Valid Values: any valid IP address

Default Value: none

new-cost

Valid Values: an integer in the range of 1 - 16

Default Value: 1

Example: `change route 10.0.0.0 255.0.0.0 128.185.123.18 6`

Delete

Use the **delete** command to delete an IP configuration item previously installed by the **add** command. In general, you must specify the item you want to delete, just as you specified the item with the **add** command.

Syntax: delete
- accept-rip-route . . .
- access-control . . .
- address . . .
- bootp-server
- default network/subnet-gateway . . .
- filter . . .
- packet-filter
- REDUNDANT Default Gateway
- route . . .
- udp-destination . . .

accept-rip-route *net-number*

Removes a route from the list of networks that the RIP protocol always accepts.

Valid Values: Any IP address contained in the list of accepted networks.

Default Value: none

Example: `delete accept-rip-route 10.0.0.0`

access-control *record-number*

Deletes one of the access control records from the global access control list.

Example: `delete access-control 2`

address *ip-interface-address*

Deletes one of the router's IP interface addresses.

Valid Values: any valid IP address

Default Value: none

Example: `delete address 128.185.123.22`

bootp-server *server-IP-address*

Removes a BOOTP server from an IP configuration.

Valid Values: any valid Bootp-Server IP address

Default Value: 0.0.0.0

Example: **`delete bootp-server 128.185.123.22`**

default *network/subnet-gateway [subnetted network]*

Deletes either the default gateway or the default subnet gateway for the specified subnetted network.

Valid Values: any valid IP address

Default Value: 0.0.0.0

Example: **`delete default subnet-gateway 128.185.0.0`**

filter *destination address destination mask*

Deletes one of the router's filtered networks. The effect of this command is immediate; you do not have to reboot the router for it to take effect.

destination address

Valid Values: any valid IP address

Default Value: 0.0.0.0

destination mask

Valid Values: 0.0.0.0 - 255.255.255.255

Default Value: none

Example: **`delete filter 127.0.0.0`**

```
Address mask [0.0.0.0]? 255.0.0.0
```

packet-filter *filter-name*

Deletes a specified packet-filter from the router's configuration.

Valid Values: any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: none

Example: **`delete packet-filter pf-in-0`**

```
IP config> delete packet-filter pf-in-0
All access controls defined for 'pf-in-0' will also be deleted.
Are you sure you want to delete(Yes or [No]): y
Deleted
IP config>
```

redundant

Deletes the redundant IP gateway from a LEC interface.

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 *destination address destination mask*

Deletes one of the router's configured static routes. The effect of this command is immediate; you do not have to reboot the router for it to take effect.

destination address

Valid Values: any valid IP address

Default Value: none

destination mask

Valid Values: any valid IP mask

Default Value: none

Example: **`delete route 10.0.0.0`**

```
Address mask [0.0.0.0]? 255.0.0.0
```

Disable

Use the **disable** command to disable IP features previously enabled by the **add** command.

Syntax: disable
- arp-net-routing
- arp-subnet-routing
- bootp-forwarding
- directed-broadcast
- echo-reply
- override default/static-routes . . .
- packet-filter
- per-packet-multipath
- receiving rip . . .
- receiving dynamic nets/subnets/host . . .
- rip
- sending default/net/subnet/poisoned/host/static . . .
- source-routing
- udp-forwarding . . .

arp-net-routing

Turns off ARP network routing. When this is enabled, the router replies by proxy to all ARP requests for remote destinations that are best reached through the router. This is the default and the generally recommended setting.

Example: **`disable arp-net-routing`**

arp-subnet-routing

Turns off the IP feature called ARP subnet routing or proxy ARP, which, when enabled, deals with hosts that have no IP subnetting support. This is the default and the generally recommended setting.

Example: **`disable arp-subnet-routing`**

bootp-forwarding

Turns off the BOOTP/DHCP relay function.

Example: **`disable bootp-forwarding`**

directed-broadcast

Disables the forwarding of IP packets whose destination is a nonlocal (for example, remote LAN) broadcast address. The source host originates the packet as a unicast where it is then forwarded as a unicast to a destination subnet and "exploded" into a broadcast. You can use these packets to locate network servers.

Note: Forwarding and exploding cannot be disabled separately.

Example: `disable directed-broadcast`

echo-reply

Disables the router's ICMP Echo Reply function. Thus a ping sent to any of the router's interfaces will not generate a reply. The router defaults to echo-reply enabled.

Example: `disable echo-reply`

override default/static-routes *ip-interface-address*

Prevents an RIP default route received on interface *ip-interface-address* from being installed as the router's default route. The **disable override static-routes** command prevents RIP routes received on interface *ip-interface-address* from overriding any of the router's static routes.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `disable override default 128.185.123.22`

packet-filter *filter-name*

Disables specified interface-specific access control list (packet-filters).

filter-name

Valid Values: Any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: None

Example: `disable packet-filter pf-in-0`

per-packet-multipath

If per-packet-multipath is disabled, the router will choose the first available path to a destination. The default for this feature is disabled.

Example: `disable per-packet-multipath`

receiving rip *ip-interface-address*

Prevents any RIP packets from being received on interface *ip-interface-address.*

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `disable receiving rip 128.185.123.22`

receiving dynamic nets/subnets/host *ip-interface-address*

The **disable receiving dynamic nets** command ensures that for RIP updates received on the interface *ip-interface-address*, the router accept only those network level routes entered by the **add accept-rip-route** command. The **disable receiving dynamic subnets** command produces the analogous

behavior for subnet routes. The **disable receiving dynamic host** produces the analogous behavior for host routes.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `disable receiving dynamic nets 128.185.123.22`

sending default/net/subnet/static *ip-interface-address ip-interface-address*

Prevents the router from advertising a default route in RIP updates sent out the interface ip-interface-address. The other flags that control the RIP routes sent out an interface are **host-routes. static-routes. net-routes**, and **subnet-routes**. You can turn these off individually. A route is advertised if it is specified by any of the enabled flags.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `disable sending net-routes 128.185.123.22`

rip

Turns off the RIP protocol.

Example: `disable rip`

source-routing

Prevents the router from forwarding source-routed packets (that is, IP datagrams that include a source-route option, thus avoiding normal IP routing tables). This option defaults to source-routing enabled.

Example: `disable source-routing`

udp-forwarding *port-number*

Disables UDP forwarding for packets received by the router with the specified UDP destination port number.

Default: UDP forwarding is disabled for all port numbers.

port-number

Valid Values: an integer in the range of 0 - 65535

Default Value: 0

Example: `disable udp-forwarding 36`

Enable

Use the **enable** command to activate IP features, capabilities, and information added to your IP configuration.

Syntax: enable
- arp-net-routing
- arp-subnet-routing
- bootp-forwarding
- directed-broadcast
- echo-reply
- override default ...
- override static-routes ...
- packet-filter
- per-packet-multipath

receiving rip ...
receiving dynamic nets ...
receiving dynamic subnets ...
rip
sending default-routes ...
sending net-routes ...
sending poisoned-reverse-routes
sending subnet-routes ...
sending static-routes ...
sending host-routes
source-routing
udp-forwarding ...

arp-net-routing

Turns on ARP network routing. When enabled, the router replies by proxy to all ARP requests for remote destinations that are best reached through the router. Use this command when there are hosts on the LAN that ARP for all destinations, instead of (as is proper) only local destinations.

Example: **`enable arp-net-routing`**

arp-subnet-routing

Turns on the router's ARP subnet routing (sometimes also called Proxy ARP) function. This function is used when there are subnet-incapable hosts attached to directly-connected IP subnets. The directly connected subnet having subnet-incapable hosts must use ARP for this feature to be useful.

The way ARP subnet routing works is as follows. When a subnet-incapable host wants to send an IP packet to a destination on a remote subnet, it does not realize that it should send the packet to a router. The subnet-incapable host therefore simply broadcasts an ARP request. This ARP request is received by the router. The router responds as the destination (hence the name proxy) if both arp-subnet-routing is enabled and if the next hop to the destination is over a different interface than the interface receiving the ARP request.

If there are no hosts on your LAN that are "subnet-incapable," do not enable ARP-subnet routing. If ARP subnet routing is needed on a LAN, it should be enabled on all routers on that LAN.

Example: **`enable arp-subnet-routing`**

bootp-forwarding

Turns on BOOTP/DHCP packet forwarding. In order to use BOOTP forwarding, you must also add one or more BOOTP servers with the **add bootp-server** command.

Example: **`enable bootp-forwarding`**

```
Maximum number of forwarding hops [4]?
Minimum seconds before forwarding [0]?
```

Maximum number of forwarding hops	Maximum number of allowable BOOTP agents that can forward a BOOTP request from the client to the Server (this is not the maximum number of IP hops to the server). **Default: 4**

Minimum seconds before forwarding
: This parameter is generally not used. Use this parameter when there is a redundant path between the client and the server, and you want to use the secondary path(s) as a standby.

Default Value: 0

directed-broadcast
: Enables the forwarding of IP packets whose destination is a network-directed or subnet-directed broadcast address. The packet is originated by the source host as a unicast where it is then forwarded as a unicast to a destination subnet and "exploded" into a broadcast. These packets can be used to locate network servers. This command enables both the forwarding and exploding of directed broadcasts. The IP packet forwarder never forwards link level broadcasts/multicasts, unless they correspond to Class D IP addresses. (See the OSPF **enable multicast-routing** command.) The default setting for this feature is enabled.

Note: Forwarding and exploding cannot be implemented separately. Also, the router will not forward all-subnets IP broadcasts.

Example: `enable directed-broadcast`

echo-reply
: Enables the building and sending of an ICMP Echo Reply in response to an ICMP Echo Request.

Example: `enable echo-reply`

override default *ip-interface-address*
: Enables received RIP information to override the router's default gateway. This command is invoked on a per-IP-interface basis. When the **enable override default** command is invoked, default RIP routes received on interface ip-interface-address overwrite the router's current default gateway, providing the cost of the new default is cheaper.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable override default 128.185.123.22`

override static-routes *ip-interface-address*
: Enables received RIP information to override some of the router's statically configured routing information. This command is invoked on a per-IP-interface basis. When the **enable override static-routes** command is invoked, RIP routing information received on interface ip-interface-address overwrite statically configured network/subnet routes providing the cost of the RIP information is cheaper.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable override static-routes 128.185.123.22`

packet-filter *filter-name*
: Enables specified interface-specific access control list (packet-filters).

filter-name

Valid Values: any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: none

Example: `enable packet-filter pf-in-0`

per-packet-multipath

If per-packet-multipath is enabled, and there are multiple equal-cost paths to a destination, then the router chooses the path for forwarding each packet in a round-robin fashion. The default for this feature is disabled.

Example: `enable per-packet-multipath`

receiving rip *ip-interface-address*

Enables the processing of RIP updates that are received on a particular interface. This command has an analogous disable command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving rip** command, no RIP updates will be accepted on interface *ip-interface-address* address.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable receiving rip 128.185.123.22`

receiving dynamic nets *ip-interface-address*

Modifies the processing of RIP updates that are received on a particular interface. This command has an analogous disable command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving dynamic nets** command, for RIP updates received on interface *ip-interface-address*, the router will not accept any network-level routes unless they have been specified in an **add accept-rip-route** command.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable receiving dynamic nets 128.185.123.22`

receiving dynamic subnets *ip-interface-address*

Modifies the processing of RIP updates that are received on a particular interface. This command has an analogous disable command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving dynamic subnets** command, for RIP updates received on interface *ip-interface-address*, the router will not accept any subnet-level routes unless they have been specified in an **add accept-rip-route** command.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable receiving dynamic subnets 128.185.123.22`

receiving dynamic host *ip-interface-address*

Modifies the processing of RIP updates that are received on a particular interface. This command has an analogous disable command. (See the **disable receiving** command.) This command is enabled by default. The opposite command is easier to describe, and that is done in the following paragraph.

If you invoke the **disable receiving dynamic host** command, for RIP updates received on interface *ip-interface-address*, the router will not accept any host routes unless they have been specified in an **add accept-rip-route** command.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable receiving dynamic host 128.185.  123.22`

rip

Enables the router's RIP protocol processing.

When RIP is enabled, the following default behavior is established:

- The router includes all network and subnet routes in RIP updates sent out on each of its configured IP interfaces.
- The router processes all RIP updates received on each of its configured IP interfaces.

To change any of the default sending/receiving behaviors, use the IP configuration commands which are defined on a per-IP-interface basis.

Example: `enable rip`

sending default-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous disable command. (See the **disable sending** command.) The effect of the **enable sending** command is additive. Each separate enable sending command specifies that a certain set of routes should be advertised from a particular interface. A route is included in a RIP update only if it has been included by at least one of the enable sending commands. The **enable sending default-routes** command specifies that the default route (if one exists) should be included in RIP updates sent out interface ip-interface-address.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable sending default-routes 128.185.123.22`

Note: Some settings of the **enable sending ...** commands are redundant. For example, if you invoke **enable sending net-routes**, **enable sending subnet-routes**, and **enable sending host-routes** for a particular interface, there is no need to also specify **enable sending static-routes** (because each static route is a network-level, subnet, or host route). By default, when you

first enable RIP, sending net-routes, sending subnet-routes, and sending host-routes are enabled for each interface, while sending static-routes and sending default are disabled.

sending net-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous disable command. (See the **disable sending** command.)

The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes should be advertised from a particular interface. A route is included in an RIP update only if it has been included by at least one of the **enable sending** commands. The **enable sending network-routes** command specifies that all network-level routes should be included in RIP updates sent out interface ip-interface-address. A network-level route is a route to a single class A, B, or C IP network.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: **`enable sending net-routes 128.185.123.22`**

sending poisoned-reverse-routes *ip-interface-address*

A technique used by RIP to improve convergence time when routes change (for complete details on the technique, refer to rfc 1058). Use of this technique increases the size of RIP update messages. You may find it more acceptable to minimize routing overhead by accepting somewhat slower convergence. The **disable sending poisoned-reverse-routes** command specifies that poisoned reverse routes should not be included in RIP updates sent out on an interface specified by the **enable ip-interface-address** command.

Default: Enabled

ip-interface-address

Valid Values: any valid IP address

Default Value: none

sending subnet-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous disable command. (See the **disable sending** command.) The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes should be advertised out a particular interface. A route is included in an RIP update only if it has been included by at least one of the enable sending commands. The **enable sending subnet-routes** command specifies that all subnet routes should be included in RIP updates sent out interface ip-interface-address. However, a subnet route is included only if ip-interface-address connects directly to a subnet of the same IP subnetted network.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: **`enable sending subnet-routes 128.185.123.22`**

sending static-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous disable command. (See the **disable sending** command.) The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes should be advertised out a particular interface. A route is included in an RIP update only if it has been included by at least one of the enable sending commands. The **enable sending static-routes** command specifies that all statically configured and directly connected routes should be included in RIP updates sent out interface ip-interface-address.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable sending static-routes 128.185.123.22`

sending host-routes *ip-interface-address*

Determines the contents of RIP updates that are sent out a particular interface. This command has an analogous **disable ...** command. (See the **disable sending** command.) The effect of the **enable sending** command is additive. Each separate **enable sending** command specifies that a certain set of routes should be advertised out a particular interface. A route is included in an RIP update only if it has been included by at least one of the **enable sending** commands. The **enable sending host-routes** command specifies that all host routes should be included in RIP updates sent out interface ip-interface-address.

source-routing

Allows the router to forward IP packets containing an IP source route option.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

Example: `enable source-routing`

udp-forwarding *port-number*

Enables UDP forwarding for packets received by the router with the specified UDP destination port number.

Default: UDP forwarding is disabled for all port numbers.

port-number

Valid Values: an integer in the range of 0 - 65535

Default Value: 0

Example: `enable udp-forwarding 36`

List

Use the **list** command to display various pieces of the IP configuration data, depending on the particular subcommand invoked.

Syntax: list
- all
- access-controls
- addresses
- bootp
- filters

packet-filter
protocols
REDUNDANT Default Gateway
rip-routes-accept
routes
sizes
tags

all
Displays the entire IP configuration.

Example: `list all`

access-controls
Displays the configured access control mode (enabled or disabled) and the list of configured global access control records. Each record is listed with its record number. This record number can be used to reorder the list with the IP **move access-control** command.

Example: `list access control`

addresses
Displays the IP interface addresses that have been assigned to the router, along with their configured broadcast formats.

Example: `list addresses`

bootp
Indicates whether BOOTP forwarding is enabled or disabled as well as the configured list of BOOTP servers.

Example: `list bootp`

packet-filter *[filter-name]*
Lists information on packet filters. If you specify a name, the command lists access control information configured for the filter. If you do not specify a filter name, the command lists configured packet-filters.

Example: `list packet-filter pf-in-0`

```
Name              Direction     Interface
pf-in-0           In            0

Access Control is: enabled
List of access control records:

                                                  Beg  End Beg End
    Ty Source      Mask      Destination  Mask    Pro  Pro Prt Prt
1   E  128.185.0.0 FFFF0000  0.0.0.0      00000000 0   255  0  65535
2   I  0.0.0.0     00000000  0.0.0.0      00000000 0   255  0  65535
```

protocols
Displays the configured state of the IP routing protocols (OSPF, RIP, BGP) along with other general configuration settings.

Example: `list protocols`

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-routes-accept

Displays the set of routes that the RIP routing protocol always accepts. See the IP configuration commands **enable/disable receiving dynamic nets/subnets/hosts** for more information.

Example: `list rip-routes-accept`

routes

Displays the list of static routes that have been configured.

Example: `list routes`

sizes

Displays the routing table size, reassembly buffer size, and the route cache size.

Example: `list sizes`

tags

Displays the per-interface tags that will be associated with received RIP information. These tags can be used to group routes together for later readvertisement via BGP where a tag will be treated as if it were a route's source autonomous system (AS). (Refer to the section titled "Originate, Send, and Receive Policies" in *Protocol Configuration and Monitoring Reference Volume 2*.) Tags are also propagated by the OSPF routing protocol.

Example: `list tags`

Move

Use the **move** command to change the order of records in the global access control list. This command places record number from# immediately after record number to#. After you move the records, they are immediately renumbered to reflect the new order.

The router applies the access control records in a list in the order that they were created. For each packet received on an interface, the router applies each access control record in order until it finds a match. The first record that matches the packet determines whether it will be discarded, or forwarded to its destination.

This makes the order of the access control records very important. If they are in the wrong order, certain packets may slip through, or be blocked, in a manner contrary to your intentions.

Let us say, for example, that access control record 1 enforces the rule: *all packets from network 10.0.0.0 shall be blocked on this interface.* Contrary to this, access control record 2 states: *Packets from subnet 10.5.5.0 in network 10.0.0.0, which are destined for address 1.2.3.4, shall be allowed to pass.* Assigned in this order, these records will block all traffic from 10.0.0.0, even though record 2 explicitly allows certain types of packets to pass.

In this example, record 1 makes record 2 moot. Record 1 guarantees that the router discards all packets from 10.0.0.0, despite the intent of record 2, which is that certain packets be forwarded. The key to fixing this type of problem is in the order of the access control records. This way, packets in subnet 10.5.5.0 and destined for address 1.2.3.4 will pass through the interface; the router discards all other packets from 10.0.0.0 as intended.

Syntax: move access-control *from# to#*

Example: `move 5 2`

Set

Use the **set** command to set certain values, routes, and formats within your IP configuration.

Syntax: set
- access-control . . .
- broadcast-address . . .
- cache-size
- default network-gateway . . .
- default subnet-gateway . . .
- internal-ip-address
- originate-rip-default
- reassembly-size
- router-id . . .
- routing table-size . . .
- tag . . .
- ttl

access-control *on* or *off*

Allows you to configure the router to enable or disable IP access control. Setting access-control *on* enables the global access control list as well as the interface specific lists. Setting it *off* disables all lists; but does not delete them

Example: `set access-control on`

broadcast-address *ip-interface-address style fill-pattern*

Specifies the IP broadcast format that the router uses when broadcasting packets out a particular interface. IP broadcasts are most commonly used by the router when sending RIP update packets.

The style parameter can take either the value local wire or the value network. Local-wire broadcast addresses are either all ones (255.255.255.255) or all zeros (0.0.0.0). Network style broadcasts begin with the network and subnet portion of the ip-interface-address.

You can set the fill-pattern parameter to either 1 or 0. This indicates whether the rest of the broadcast address (that is, other than the network and subnet portions, if any) should be set to all ones or all zeros.

When receiving the router recognizes all forms of the IP broadcast address.

ip-interface-address

Valid Values: any valid IP address

Default Value: none

style

Valid Values: *local-wire* or *network*

Default Value: none

fill-pattern

Valid Values: *0* or *1*

Default Value: none

The example below configures a broadcast address of 255.255.255.255. The second example produces a broadcast address of 192.9.1.0, assuming that the network 192.9.1.0 is not subnetted.

Example:
```
set broadcast-address 192.9.1.11 local-wire 1
set broadcast-address 192.9.1.11 network 0
```

cache-size *entries*

Configures the maximum number of entries for the IP routing cache. This cache stores information about the specific IP addresses to which the router has recently forwarded packets. The cache reduces the processing time needed to forward multiple packets to the same destination.

In contrast with this cache, the IP routing *table* stores information about all accessible networks but does not contain specific IP destination addresses. Use the **set routing table-size** command to configure the size of the IP routing table.

Valid Values: 64 to 10000

Default Value: 64

Example:
```
set cache-size 64
```

default network-gateway *next-hop cost*

Configures a route to the authoritative router (default gateway). You should assume that the router's default gateway has more complete routing information than the router itself.

The route is specified by the IP address of the next hop (next-hop) and the distance (cost) to the default gateway.

All packets having unknown destinations are forwarded to the authoritative router (default gateway).

next-hop **Valid Values**: any valid IP address

Default Value: 0.0.0.0 with a gateway cost of 1.

cost **Valid Values**: an integer in the range of 1 - 16

Default Value: 1

Example:
```
set default network-gateway 192.9.1.10 10
```

default subnet-gateway *subnetted-network next-hop cost*

Configures a route to a subnetted network's authoritative router (default subnet gateway). You can configure a separate default subnet gateway for each subnetted network.

The IP address of the next hop (next-hop) and the distance (cost) to the default subnet gateway specify the route.

All packets destined for unknown subnets of a known subnetted network are forwarded to the subnetted network's authoritative router (default subnet gateway).

subnetted network

Valid Values: any valid IP address

Default Value: 0.0.0.0

next-hop

Valid Values: any valid IP address

Default Value: 0.0.0.0

cost

Valid Values: an integer in the range of 1 - 16

Default Value: 1

Example: **`set default subnet-gateway 128.185.0.0`** `128.185.123.22 6`

internal-IP-address *ip-address*

Configures an IP address that is independent of the state of any interface. The internal address is always considered active. The primary reason for defining an internal address is to provide an address for a TCP connection that will not become inactive when an interface becomes inactive. This address is used for data link switching (DLSw), allowing alternate paths to be used to avoid disrupting DLSw connections when an interface becomes inactive. Because the internal address remains active and because OSPF maintains active IP routes to this destination, IP routing can switch DLSw traffic onto the alternate path without bringing down the TCP connection or disrupting the SNA sessions that are running on top of DLSw.

The internal IP address also provides some value when unnumbered interfaces are used. It is the first choice as a source address for packets originated by this router and transmitted over an unnumbered interface. The stability of this address makes it easier to keep track of such packets. The chance for confusion is further reduced when the same IP address is used for both the router ID and the internal address. Therefore the router ID will default to the internal address.

When an internal address is defined it will be advertised by OSPF as a host route into all areas directly attached to the router.

Valid Values: any valid IP address.

Default Value: none

Example: **`set internal-ip-address 142.82.10.1`**

originate-rip-default

Causes RIP to advertise this router as the default gateway. Use this command in the following environment:

- The IP routes in this router's routing table are determined by a number of protocols.
- RIP is one of those protocols.
- At most partial routing information is imported from the other protocols and advertised by RIP.

Traffic in the RIP network for destinations that are not known by RIP can follow the default path to this router. The more complete routing information in this node's route table can then be used to forward the traffic along an appropriate path towards its destination. You can configure the router to only originate the default when routes are known to this router that will not be advertised in the RIP network.

When you issue this command, you will be prompted as to whether to originate a rip-default for the other routing protocols your router is running.

This default route will direct traffic bound for a non-RIP network to a boundary router. Originating a single default route means that the boundary router does not have to distribute the other network's routing information to the other nodes in its network.

from AS number

Valid Values: an integer in the range of 0 - 65535

Default Value: 0

to network number

Valid Values: any valid IP address

Default Value: none

default cost

Valid Values: an integer in the range of 1 - 16

Default Value: 1

Example: `set originate-rip-default`

```
IP Config> set originate rip-default
Always originate default route? [No]:?
Originate default if BGP routes available? [No] yes
  From AS number [6]?
  To network number [0.0.0.0]?
Originate default if OSPF routes available? [No]
Originate default cost [1]?
```

means a default route is always originated

- Answering "yes" to the "BGP" question originates a default whenever there are BGP routes in the routing table.
- Answering "yes" to the "if OSPF routes available" question causes the RIP default to be advertised when OSPF routes are in the routing table.
- When the router does decide to originate a RIP default, it uses the "original default cost" number.

reassembly-size *bytes*

Configures the size of the buffers that are used for the reassembly of fragmented IP packets.

Default: 12000

Example: `set reassembly-size 12000`

router-id *ip-address*

Sets the default IP address used by the router when sourcing various IP packets. This address is of particular importance in multicasting and OSPF.

The router ID must match one of the configured IP interface addresses of the router or the configured internal IP address. If not, it is ignored. When ignored, or just not configured, the default IP address of the router (and its OSPF router ID) is set to the internal IP address (if configured) or to the first IP address in the router's configuration.

Valid Values: any valid IP address

Default Value: none

Example: `set router-id 128.185.120.209`

routing table-size *number-of-entries*

Sets the size of the router's IP routing table. The default size is 768 entries. Setting the routing table size too small causes dynamic routing information to be discarded. Setting the routing table size too large wastes router memory resources. See "Sizes" on page 15-7 for additional information about table sizes.

Valid Values: an integer number of entries in the range of 1 - 65535

Default Value: 768 entries

Example: **`set routing table-size 1000`**

tag

Configures the per-interface tags associated with received RIP information. These tags can be used to group routes together for later readvertisement via BGP where a tag will be treated as if it were a route's source autonomous system (AS) number. (Refer to the section titled "Originate, Send, and Receive Policies" in the chapter "Using and Configuring BGP" in *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.) Tags are propagated also by the OSPF routing protocol.

Valid Values: an integer in the range of 0 - 65535

Default Value: 0

Example: **`set tag`**

```
Interface address [0.0.0.0]? 1.1.1.1
Interface tag (AS number) [0]? 1
```

ttl

Specifies the time-to-live for packets originated by the router.

Valid Values: a numeric in the range of 1 - 255

Default Value: 64

Example: **`set ttl 255`**

Update

Use the **update packet-filter** command at the `IP config>` prompt to assign access control entries. The router prompts you for the name of the filter you want to update. The `IP config>` prompt changes to incorporate the packet filter name you provide.

Valid Values: any 16-character name.

You can include dashes (-) and underscores (_) in the name.

Default Value: none

```
IP Config> update packet-filter
Packet-filter name [ ]? pf-1-in
Packet-filter 'pf-1-in' Config>
```

You can access a list of sub-commands by typing **?** at the `Packet-filter 'name' Config>` prompt.

```
Packet-filter 'test' Config> ?
LIST
CHANGE
DELETE
ADD
MOVE
EXIT
```

Adding and Changing Access Controls to a Packet Filter

Use the **add access-control** command to add access controls to the specified packet filter. The router prompts you for the access control type (either Exclusive or Inclusive), and the source and destination addresses and masks of packets to which the filter will apply.

type **Valid Values**:

- *Exclusive*— Specifies that any packets matching one or more of the filters in the access control list for this interface will be dropped.
- *Inclusive*— Specifies that only packets matching one or more of the filters in the access control list for this interface will be forwarded.

Default Value: Exclusive

source address

Valid Values: A valid IP address in dotted decimal notation.

Default Value: 0.0.0.0

source mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: 255.255.255.255

destination address

Valid Values: A valid IP address in dotted decimal notation.

Default Value: 0.0.0.0

destination mask

Valid Values: 0.0.0.0 to 255.255.255.255

Default Value: 255.255.255.255

first protocol

The lower boundary of a protocol number range.

The commonly used protocol numbers are:

- **1** for ICMP
- **6** for TCP
- **17** for UDP
- **89** for OSPF.

See RFC 1340, "Assigned Numbers" for details on IP protocol numbers.

Valid Values: 0 to 255

Default Value: 0

last protocol

The upper boundary of a protocol number range.

The commonly used protocol numbers are:

- **1** for ICMP
- **6** for TCP

- **17** for UDP
- **89** for OSPF.

See RFC 1340, "Assigned Numbers" for details on IP protocol numbers.

Valid Values: 0 to 255

Default Value: 0

first port

The lower boundary of a IP TCP/UDP port range.

Valid Values: a port number in the range of 0 - 65535

Address Default Value: 0

Some commonly used port numbers are:

- 21 for FTP
- 23 for Telnet
- 25 for SMTP
- 513 for rlogin
- 520 for RIP

last port

The upper boundary of a IP TCP/UDP port range.

Valid Values: a port number in the range of 0 - 65535

Address Default Value: 0

Some commonly used port numbers are:

- 21 for FTP
- 23 for Telnet
- 25 for SMTP
- 513 for rlogin
- 520 for RIP

This example shows how to exclude all incoming packets originating from network 128.185.0.0 and received on interface 0.

```
Packet-filter 'pf-in-0' Config> add access-control
Enter type [E]?
Internet source [0.0.0.0]? 128.185.0.0
Source mask [255.255.255.255]? 255.255.0.0
Internet destination [0.0.0.0]?
Destination mask [255.255.255.255]? 0.0.0.0
Enter starting protocol number ([CR] for all) [-1]?
```

Use the **change access-control** command to change existing access controls using the index number of the access control that you want to change.

You can use the **list access-control** command to view the access controls configured for each packet filter.

```
Packet-filter 'pf-in-0' Config> list access-control
Access Control is: enabled
List of access control records:

                                                Beg  End Beg End
    Ty Source      Mask      Destination  Mask    Pro  Pro Prt Prt
1   E  128.185.0.0 FFFF0000  0.0.0.0      00000000 0   255  0  65535
2   I  0.0.0.0     00000000  0.0.0.0      00000000 0   255  0  65535
```

You can change the order of a packet filter's access control records with the **move access-control** command as shown.

```
Packet-filter 'test' Config> move access-control
Enter index of control to move [1]?
Move record AFTER record number [0]? 2
About to move:

                                                Beg  End Beg End
    Ty Source      Mask      Destination  Mask    Pro  Pro Prt Prt
1   E  10.0.0.0    FFFF0000  0.0.0.0      00000000  0  255  0  65535
to be after:
2   I  10.5.5.0    FFFF0000  1.2.3.4      FF0000FF 0   255  0  65535
Are you sure this is what you want to do (Yes or [No]): y
```

Deleting Access Controls for a Filter

Use the **delete access-control** command to delete a record from a packet filter's access-control list.

```
Packet-filter 'test' Config> delete access-control
Enter index of access control to be deleted [1]? 4
```

The router responds by displaying the access-control record you have specified.

```
                                                Beg  End Beg End
    Ty Source      Mask      Destination  Mask    Pro  Pro Prt Prt
4   I  1.2.9.9     FF0000FF  0.0.0.0      00000000 0   255  0  65535
Are you sure this is the record you want to delete (Yes or [No]): y
Deleted
Packet-filter 'test' Config>
```

Exiting the Access Controls Process

Exit the access controls process by typing **exit** at the prompt. This returns you to the `IP config>` prompt.

```
Packet-filter 'test' Config> exit
IP config>
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Chapter 15. Monitoring IP

This chapter describes the IP console commands and includes the following sections:

- "Accessing the IP Console Environment"
- "IP Console Commands"

Accessing the IP Console Environment

For information on how to access the IP console environment, refer to "Getting Started (Introduction to the User Interface)" in the *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

IP Console Commands

This section summarizes and then explains all the IP console commands. Table 15-1 lists the IP console commands. The commands allow you to monitor the router's IP forwarding process.

Table 15-1. IP Console Command Summary

Command	Function
? (Help)	Lists the console commands or lists the actions associated with specific commands.
Access controls	List the current IP access control mode, together with the configured access control records.
Cache	Displays a table of all recent routed destinations.
Counters	Lists various IP statistics, including counts of routing errors and packets dropped.
Dump routing tables	Lists the contents of the IP routing table.
Interface addresses	Lists the router's IP interface addresses.
Packet-filter	Displays the access-control information defined for the specified packet-filter, or all filters.
Ping	Sends ICMP Echo Requests to another host and watches for a response. This command can be used to isolate trouble in an internetwork environment.
Redundant Default Gateway	Lists whether a redundant default gateway exists and if it is active or inactive.
Route	Lists whether a route exists for a specific IP destination, and if so, the routing table entry that corresponds to the route.
Sizes	Displays the size of specific IP parameters.
Static routes	Displays the static routes that have been configured. This includes the default gateway.
Traceroute	Displays the complete path (hop-by-hop) to a particular destination.
Exit	Exits the IP console environment.

? (Help)

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

Syntax: ?

Example: ?

```
ACCESS controls
CACHE
COUNTERS
DUMP routing tables
INTERFACE addresses
PACKET-FILTER summary
PING dest-addr [src-addr size ttl rate]
REDUNDANT default gateway
ROUTE given address
SIZES
STATIC routes
TRACEROUTE address
UDP-FORWARDING
EXIT
```

Access Controls

Use the **access controls** command to print the global access control mode in use together with a list of the configured access control records.

The access control mode is either disabled (meaning that no access control is being done and the access control records are being ignored) or enabled (meaning that access control is being done and the access control records are being recognized). When access control is enabled, access control records are scanned in order looking for the first match.

Exclusive (E) means that packets matching the access control record are being discarded. Inclusive (I) means that packets matching the access control record are being forwarded. When access control is enabled, packets failing to match any access control record are discarded. Pro (protocol) indicates the IP protocol number, and Port indicates the UDP or TCP destination port number.

Syntax: access

Example: access

```
Access Control currently enabled
Access Control run 13 times, 14 cache hits

List of access control records:
                                                          Beg End Beg  End
    Ty Source        Mask      Destination   Mask      Pro Pro Port Port   Use
1   E  10.5.22.0     FFFFFF00  0.0.0.0       00000000  0   255 0    65535  0
2   E  0.0.0.0       00000000  10.5.22.0     FFFFFF00  0   255 0    65535  0
3   I  0.0.0.0       00000000  0.0.0.0       00000000  0   255 0    65535  27
```

The IP access control system is based on a global list of *inclusive* and *exclusive* access control records. If access control is enabled, each IP packet being originated, forwarded, or received, is subject to the access control list.

The Use field (far right) specifies the number of times the access control system matched a particular record to an incoming packet, for example, the number of

times that a particular record in the IP access controls system was invoked by the characteristics of an incoming or outgoing packet.

Cache

Use the **cache** command to display the IP routing cache which contains recently routed destinations. If a destination is not in the cache, the router looks up the destination in the routing information table in order to make a forwarding decision.

Syntax: cache

Example: **`cache`**

```
Destination     Usage           Next hop
128.185.128.225 1               128.185.138.180
```

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
```

Routing table overflow	Lists the number of routes that have been discarded due to the routing table being full.
Net unreachable	Indicates the number of packets that could not be forwarded due to unknown destinations. This does not count the number of packets that have been forwarded to the authoritative router (default gateway).
Bad subnet number	Counts the number of packets or routes that have been received for illegal subnets (all ones or all zeroes).
Bad net number	Counts the number of packets or routes that have been received for illegal IP destinations (for example, class E addresses).

Unhandled broadcasts	Counts the number of (non-local) IP broadcasts received (these are not forwarded).
Unhandled multicasts	Counts the number of IP multicasts that have been received, but whose addresses were not recognized by the router (these are discarded).
Unhandled directed broadcasts	Counts the number of directed (non-local) IP broadcasts received when forwarding of these packets is disabled.
Attempted forward of LL broadcast	Counts the number of packets that are received having non-local IP addresses but were sent to a link level broadcast address. These are discarded.
Packets discarded through filter	Counts the number of received packets that have been addressed to filtered networks/subnets. These are discarded silently.
IP multicasts accepted	Counts the number of IP multicasts that have been received and successfully processed by the router.
IP packet overflows	Counts the number of packets that have been discarded due to congestion at the forwarder's input queue. These counts are sorted by the receiving interface.

Dump Routing Table

Use the **dump** command to display the IP routing table. A separate entry is printed for each reachable IP network/subnet. The IP default gateway in use (if any) is listed at the end of the display.

Syntax: dump

Example: **dump**

```
Type   Dest net         Mask     Cost Age   Next hop(s)
SPE1   0.0.0.0          00000000  4    3    128.185.138.39  (2)
Sbnt   128.185.0.0      FFFF0000  1    0    None
SPF    128.185.123.0    FFFFFF00  3    3    128.185.138.39  (2)
SPF    128.185.124.0    FFFFFF00  3    3    128.185.138.39  (2)
SPF    192.26.100.0     FFFFFF00  3    3    128.185.131.10  (2)
RIP    197.3.2.0        FFFFFF00  10   30   128.185.131.10
RIP    192.9.3.0        FFFFFF00  4    30   128.185.138.21
Del    128.185.195.0    FFFFFF00  16   270  None


Default gateway in use.

Type Cost Age  Next hop
SPE1 4    3    128.185.138.39

Routing table size: 768 nets (36864 bytes), 36 nets known
```

Type (route type) Indicates how the route was derived.

Sbnt - Indicates that the network is subnetted; such an entry is a place-holder only.

Dir - Indicates a directly connected network or subnet.

RIP - Indicates the route was learned through the RIP protocol.

Del - Indicates the route has been deleted.

Stat - Indicates a statically configured route.

	BGP - Indicates routes learned through the BGP protocol.
	BGPR - Indicates routes learned through the BGP protocol that are readvertised by OSPF and RIP.
	Fltr - Indicates a routing filter.
	SPF - Indicates that the route is an OSPF intra-area route.
	SPIA - Indicates that it is an OSPF inter-area routes.
	SPE1, SPE2 - Indicates OSPF external routes (type 1 and 2 respectively).
	Rnge - Indicates a route type that is an active OSPF area address range and is not used in forwarding packets.
Dest net	IP destination network/subnet.
Mask	IP address mask.
Cost	Route Cost.
Age	For RIP and BGP routes, the time that has elapsed since the routing table entry was last refreshed.
Next Hop	IP address of the next router on the path toward the destination host. Also displayed is the interface type used by the sending router to forward the packet.

An asterisk (*) after the route type indicates the route has a static or directly connected backup. A percent sign (%) after the route type indicates that RIP updates will always be accepted for this network/subnet.

A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination. The first hops belonging to these routes can be displayed with the IP **route** command.

Interface Addresses

Use the **interface addresses** command to display the router's IP interface addresses. Each address is listed together with its corresponding hardware interface and IP address mask.

Hardware interfaces having no configured IP interface addresses will not be used by the IP forwarding process; they are listed as Not an IN net. There is one exception. Serial lines need not be assigned IP interface addresses in order to forward IP traffic. Such serial lines are called unnumbered. They show up as having address 0.0.0.0.

Syntax: interface

Interface	Indicates the hardware type of the interface.
IP addresses	Indicates the IP address of the interface.
Mask	Indicates the subnet mask of the interface.

Packet-filter

Use the **packet-filter** command to display information defined for a specific packet filter, or for all filters. Packet-filters are interface-specific lists of access control records.

Syntax: packet-filter [*name*]

Example: **packet-filter pf-in-0**

```
Name              Direction     Interface   #Access-Controls
pf-in-0           In            0           2

Access Control currently enabled
Access Control run 8 times, 7 cache hits

List of access control records:

                                                    Beg End  Beg  End
  Ty  Source       Mask     Destination  Mask      PPP PPP  Port Port  Use
0 I   0.0.0.0      00000000 192.67.67.20 00000000  6   6    25   25    0
1 E   150.150.1.0  FFFFFF00 150.150.2.0  00000000  0   255  0    655   0
2 I   0.0.0.0      00000000 0.0.0.0      00000000  89  89   0    655   27
```

Ping

Use the **ping** command to have the router send ICMP Echo Requests to a given destination (that is, “pinging”) and watch for a response. This command can be used to isolate trouble in an internetwork environment.

This process is done continuously, incrementing the ICMP sequence number with each additional packet. Each matching received ICMP Echo response is reported with its sequence number and the round-trip time. The granularity (time resolution) of the round trip time calculation is usually around 20 milliseconds, depending on the platform.

To stop the pinging process, type any character at the console. At that time, a summary of packet loss, round-trip time, and number of unreachable ICMP destinations will be displayed.

When a multicast address is given as destination, there may be multiple responses printed for each packet sent, one for each group member. Each returned response is displayed with the source address of the responder.

The size of the ping (number of data bytes in the ICMP message, excluding the ICMP header), TTL value, and frequency of pinging are all user configurable. The source IP address is also configurable. If not specified, the router uses its local address on the outgoing interface to the specified destination. If you are validating connectivity from any of the router’s other interfaces to the destination, enter the IP address for that interface as the source address.

The default values are a size of 56 bytes, a TTL of 64 seconds, and a frequency of 1 ping per second. Only the destination value is required, all other values are optional.

Syntax: ping *dest-addr [src-addr size ttl rate]*

Example: **ping 128.185.142.06 128.185.142.11 56 60 1**

```
PING 128.185.142.11 -> 128.185.142.06: 56 data bytes, ttl = 60 every 1 sec.
56 bytes from 128.185.142.06: icmp_seq=0 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=1 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=2 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=3 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=4 time=0 ms
56 bytes from 128.185.142.06: icmp_seq=5 time=0 ms

----128.185.142.06 PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0 ms.
```

Redundant Default Gateway

Use the **redundant default gateway** command to display the redundant Default IP Gateways configured for each interface.

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.”

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`**

Example: **`route 128.185.230.0`**

Example: **`route 128.185.232.0`**

Sizes

Use the **sizes** command to display the configured sizes of specific IP parameters.

Syntax: sizes

Example: **`sizes`**

```
Routing table size:         768
Table entries used:         3
Reassembly size:            12000
Largest reassembled pkt:    0
Size of routing cache:      64
# of cache entries in use:  0
```

Routing table size — The configured number of entries that the routing table will maintain.

Table entries used	The number of entries used from the routing table. This number includes both active and inactive entries. The value displayed using the “dump” command as “xx nets known” is the number of active routing table entries. The configured routing table size should be large enough to maintain current active entries as well as other anticipated routing entries.
Reassembly buffer size	The configured size of the reassembly buffer that is used to reassemble fragmented IP packets.
Largest reassembled pkt	The largest IP packet that this router has had to reassemble.
Size of routing cache	The configured size of the routing cache.
# of cache entries in use	The number of entries currently being used from the cache.

Static Routes

Use the **static routes** command to display the list of configured static routes. Configured default gateways and default subnet gateways are also listed.

Each static route’s destination is specified by an address-mask pair. Default gateways appear as static routes to destination 0.0.0.0 with mask 0.0.0.0. Default subnet gateways also appear as static routes to the entire IP subnetted network.

The example below shows a configured default gateway, a configured default subnet gateway (assuming 128.185.0.0 is subnetted), and a static route to network 192.9.10.0.

Syntax: static

Example: **`static`**

```
Net             Mask            Cost  Next hop
0.0.0.0         0.0.0.0         1     128.185.123.18
128.185.0.0     255.255.0.0     1     128.185.123.22
192.9.10.0      255.255.255.0   10    128.185.123.22
```

Net	The network address of the route.
Mask	The subnet mask of the IP address.
Cost	The cost of using this route.
Next Hop	The next router a packet would pass through using this route.

Traceroute

Use the **traceroute** command to display the entire path to a given destination, hop by hop. For each successive hop, **traceroute** sends out three probes, and prints the IP address of the responder, together with the round trip time associated with the response. If a particular probe receives no response, an asterisk is displayed. Each line in the display relates to this set of three probes, with the leftmost number indicating the distance from the router executing the command (in router hops).

The traceroute is done whenever the destination is reached, an ICMP Destination Unreachable is received, or the path length reaches 32 router hops.

When a probe receives an unexpected result, several indications can be displayed. “!N” indicates that an ICMP Destination Unreachable (net unreachable) has been received. “!H” indicates that an ICMP Destination Unreachable (host unreachable)

has been received. "!P" indicates that an ICMP Destination Unreachable (protocol unreachable) has been received; because the probe is a UDP packet sent to a strange port, a port unreachable is expected "!" indicates that the destination has been reached, but the reply sent by the destination has been received with a TTL of 1. This usually indicates an error in the destination, prevalent in some versions of UNIX, whereby the destination is inserting the probe's TTL in its replies. This unfortunately leads to a number of lines consisting solely of asterisks before the destination is finally reached.

Syntax: traceroute *interface-address*

Example: **`traceroute 128.185.142.239`**

```
TRACEROUTE 128.185.124.110: 56 data bytes
 1 128.185.142.7 16 ms 0 ms 0 ms
 2 128.185.123.22 16 ms 0 ms 16 ms
 3 * * *
 4 * * *
 5 128.185.124.110 16 ms ! 0 ms ! 0 ms !
```

TRACEROUTE	Displays the destination area address and the size of the packet being sent to that address.
1	The first trace showing the destination's NSAP and the amount of time it took the packet to arrive at the destination. The packet is traced three times.
Destination unreachable	Indicates that no route to destination is available.
*1 * * ** *2 * * **	Indicates that the router is expecting some form of response from the destination, but the destination is not responding.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Chapter 16. Using and Configuring OSPF

This chapter describes how to use the Open Shortest Path First (OSPF) Protocol, which is an Interior Gateway Protocol (IGP). The router supports the following IGPs for building the IP routing table, Open Shortest Path First (OSPF) Protocol and RIP Protocol. OSPF is based on link-state technology or the shortest-path first (SPF) algorithm. RIP is based on the Bellman-Ford or the distance-vector algorithm.

Included in this chapter are the following sections:

Routers that use a common routing protocol form an *autonomous system* (AS). This common routing protocol is called an Interior Gateway Protocol (IGP). IGPs dynamically detect network reachability and routing information within an AS and use this information to build the IP routing table. IGPs can also import external routing information into the AS. The router can simultaneously run OSPF and RIP. When it does, OSPF routes are preferred. In general, use of the OSPF protocol is recommended due to its robustness, responsiveness, and decreased bandwidth requirements.

The OSPF Routing Protocol

The router supports a complete implementation of the OSPF routing protocol, as specified in RFC 1583 (Version 2). This version is incompatible with bridging routers running OSPF Version 1. OSPF information will not be exchanged between routers running Version 1 and Version 2.

OSPF is a link-state dynamic routing protocol that detects and learns the best routes to (reachable) destinations. OSPF can quickly perceive changes in the topology of an AS, and after a short convergence period, calculate new routes. The OSPF protocol does not encapsulate IP packets, but forwards them based on destination address only.

OSPF Routing Summary

When a router is initialized, it uses the Hello Protocol to send hello packets to its neighbors, and they in turn send their packets to the router. On broadcast and point-to-point networks, the router dynamically detects its neighboring routers by sending the Hello packets to the multicast address *ALLSPFRouters*; on non-broadcast networks you must configure information to help the router discover its *neighbors*. On all multi-access networks (broadcast and non-broadcast), the Hello Protocol also elects a *designated router* for the network.

Note: For ATM networks, RFC 1577 will allow IP to use the network as a Non-Broadcast Multiple 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 not available with RIP. OSPF includes the following features:

- *Least-Cost Routing.* Allows you to configure path costs based on any combination of network parameters. For example, bandwidth, delay, and dollar cost.
- *No limitations to the routing metric.* While RIP restricts the routing metric to 16 hops, OSPF has no restriction.
- *Multipath Routing.* Allows you to use multiple paths of equal cost that connect the same points. You can then use these paths for load distribution that results in more efficient use of network bandwidth.
- *Area Routing.* Decreases the resources (memory and network bandwidth) consumed by the protocol and provides an additional level of routing protection.
- *Variable-Length Subnet Masks.* Allow you to break an IP address into variable-size subnets, conserving IP address space.
- *Routing Authentication.* Provides additional routing security.

OSPF supports the following physical network types:

- *Point-to-Point.* Networks that use a communication line to join a single pair of routers. A 56-Kbps serial line that connects two routers is an example of a point-to-point network.
- *Broadcast.* Networks that support more than two attached routers and are capable of addressing a single physical message to all attached routers. A token-ring network is an example of a broadcast network. Emulated LANs over ATM treat the ATM network as a broadcast network.
- *Non-Broadcast.* Networks that support more than two attached routers but have no broadcast capabilities. An X.25 Public Data Network is an example of a non-broadcast network. For OSPF to function properly, this network requires extra configuration information about other OSPF routers attached to the non-broadcast network. Classical IP over ATM (RFC 1577) treats the ATM interface as a Non-Broadcast Multiple Access (NBMA) interface.

Designated Router

Every multi-access network has a designated router that performs two main functions for the routing protocol: it originates network link advertisements and it becomes adjacent to all other routers on the network.

When a designated router originates network link advertisements, it lists all the routers, including itself, currently attached to the network. The link ID for this advertisement is the IP interface address of the designated router. By using the subnet/network mask, the designated router obtains the IP network number.

The designated router becomes adjacent to all other routers and is tasked with synchronizing the link state databases on the broadcast network.

The Hello Protocol elects the designated router after determining the router's priority from the *Rtr Pri* field of the hello packet. When a router's interface first becomes functional, it checks to see if the network currently has a designated router. If it does, it accepts that designated router regardless of that router's priority, otherwise, it declares itself the designated router. If the router declares itself the designated router at the same time that another router does, the router with higher router priority (*Rtr Pri*) becomes the designated router. If both *Rtr Pris* are equal, the one with the higher router ID is elected.

Once the designated router is elected, it becomes the endpoint for many adjacencies. On a broadcast network, this optimizes the flooding procedure by allowing the designated route to multicast its Link State Update packets to the address ALLSPFRouters rather than sending separate packets over each adjacency.

Multicast OSPF

Multicasting is a LAN technique that allows copies of a single packet to pass to a selected subset of all possible destinations. Some hardware (Ethernet, for example) supports multicast by allowing a network interface to belong to one or more multicast groups. Refer to "IP Multicast Support" on page 14-12 for details about the router's support of IP multicasting.

The OSPF protocol supports IP multicast routing through multicast extensions to OSPF (MOSPF).

An MOSPF router distributes group location information throughout the routing domain by flooding a new type (type 6) of link state advertisement, the group-membership-LSA. This enables the MOSPF routers to efficiently forward a multicast datagram to its multiple destinations. Each router does this by calculating the path of the multicast datagram as a tree whose root is the datagram source and whose terminal branches are LANs containing group members.

While running MOSPF, multicast datagram forwarding works in the following ways:

- Although forwarding IP multicasts is not reliable, IP multicast datagrams are delivered with the same best effort as with the delivery of IP unicasts.
- Multicast datagrams travel the shortest path between the datagram source and any particular destination (OSPF link state cost). This occurs because a separate tree is built for each datagram source and destination group pair.

- A multicast datagram is forwarded as a datalink multicast at each hop. The ARP protocol is not used. For some network technologies, mapping between Class D addresses and datalink multicast occurs while for others, Class D addresses are mapped to the datalink broadcast address.
- When paths from the datagram source to two separate group members share an initial common segment, only a single datagram is forwarded until the paths go in separate directions. The path can split at either a router or at a network. If the path splits at a router, the router replicates the packet before it is sent. If the path splits at a network, it replicates through a datalink multicast.
- A network configuration could include both MOSPF routers and routers without multicast extensions. In this configuration, all routers interoperate in the routing of unicasts. This allows you to slowly introduce multicast capability into an internetwork.

  Some configurations of MOSPF and non-MOSPF routers may produce unexpected failures in multicast routing.
- Separate multicast paths are constructed in MOSPF for each TOS. However, some routers do not support TOS-based routing. You can mix non-TOS routers with TOS-based routers but this causes TOS to be ignored in the forwarding of multicasts.
- The router can be configured to send SNMP traps to a multicast group address by adding a group address to a particular SNMP community name.

Configuring OSPF over ATM

The options for configuring OSPF over an ATM subnetwork depend on whether LAN Emulation or Classical IP over ATM is being used for the IP layer. In the case of LAN Emulation, OSPF is configured in the same way as for a real LAN. For Classical IP over ATM the OSPF configuration options are the same as for Wide Area Subnetworks. See “Configuring Wide Area Subnetworks” on page 16-12. Both NBMA and P-2-MP configurations are supported.

Configuring OSPF

The following sections present information on how to initially configure the OSPF protocol. This information outlines the tasks required to get the OSPF protocol up and running. Information on how to make further configuration changes is explained under “OSPF Configuration Commands” on page 16-18.

The following steps outline the tasks required to get the OSPF protocol up and running. The sections that follow explain each step in detail, including examples.

Before your router can run the OSPF protocol, you must do the following:

1. Enable the OSPF protocol. In doing so, you must estimate the final size of the OSPF routing domain. (See “Enabling the OSPF Protocol” on page 16-5.)
2. Set the OSPF router ID. (See “Setting OSPF Router IDs” on page 16-6.)
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 16-6.)

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 16-10.)
5. If you want to forward IP multicasts (IP Class D addresses), enable IP multicast routing capability. (See "Multicast Forwarding" on page 16-12.)
6. If the router interfaces to non-broadcast networks such as ATM using RFC 1577 (Classical IP and ARP over ATM), you must set additional interface parameters. (See "Setting Non-Broadcast Network Interface Parameters" on page 16-12.) (See "Configuring Wide Area Subnetworks" on page 16-12.)
7. If you want the router to import routes learned from other routing protocols running on this router (BGP, RIP or statically configured routes), you have to enable AS boundary routing. In addition, you must define whether routes are imported as Type 2 or Type 1 externals. (See "Enabling AS Boundary Routing" on page 16-14.)
8. If you want to boot a neighboring router over an attached point-to-point interface, you must configure the neighbor's IP address. Do this by adding the neighbor for the point-to-point interface.

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 (DR) 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.

Enabling the OSPF Protocol

When enabling the OSPF routing protocol, you must supply the following two values to estimate the final size of the OSPF routing domain:

- Total number of AS external routes that will be imported into the OSPF routing domain. A single destination may lead to multiple external routes when it is imported by separate AS boundary routers. For example, if the OSPF routing domain has two AS boundary routers, both importing routes to the same 100 destinations, set the number of AS external routes to 200.
- Total number of OSPF routers in the routing domain.

Configure these two values identically in all of your OSPF routers. Each router running the OSPF protocol has a database describing a map of the routing domain. This database is identical in all participating routers. From this database the IP routing table is built through the construction of a shortest-path tree, with the router itself as root. The routing domain refers to an AS running the OSPF protocol.

To enable the OSPF routing protocol, use the **enable** command as shown in the following example.

```
OSPF Config> enable ospf
Estimated # external routes[0]? 200
Estimated # OSPF routers [0]? 60
```

Setting OSPF Router IDs

Every router in an OSPF routing domain must be assigned a unique 32-bit router ID. The value used for the OSPF router ID is chosen as follows:

If the IP configuration **set router ID** command is used, the value configured is used as an OSPF router ID.

If the IP configuration **set internal address** command is used, the address configured is used as the OSPF router ID. It is recommended that the same value be used for the router ID and internal address, if defined.

If neither the router ID nor the internal address are configured during IP configuration, the first OSPF interface address will be used as the OSPF router ID.

Defining Backbone and Attached OSPF Areas

Figure 16-1 on page 16-8 shows a sample diagram of the structure of an OSPF routing domain. One division is between IP subnetworks within the OSPF domain and IP subnetworks external to the OSPF domain. The subnetworks included within the OSPF domain are subdivided into regions called *areas*. OSPF areas are collections of contiguous IP subnetworks. The function of areas is to reduce the OSPF overhead required to find routes to destinations in a different area. Overhead is reduced both because less information is exchanged between routers and because fewer CPU cycles are required for a less complex route table calculation.

Every OSPF routing domain must have at least a *backbone area*. The backbone is always identified by area number 0.0.0.0.. For small OSPF networks, the backbone is the only area required. For larger networks with multiple areas, the backbone provides a core that connects the areas. Unlike other areas, the backbone's subnets can be physically separate. In this case, logical connectivity of the backbone is maintained by configuring *virtual links* between backbone routers across intervening non-backbone transit areas.

Routers that attach to more than one area function as area *border routers*. All area border routers are part of the backbone, so a border router must either attach directly to a backbone IP subnet or be connected to another backbone router over a virtual link. In addition, there must be a collection of backbone subnetworks and virtual links that connects all of the backbone routers.

The information and algorithms used by OSPF to calculate routes vary according to whether the destination IP subnetwork is within the same area, in

a different area within the same domain, or external to the OSPF domain. Every router maintains a complete map of all links within its area. All router to multi-access network, network to multi-access router, and router to router links are included in the map. A shortest path first algorithm is used to calculate the best routes to destinations within the area from this map. Routes between areas are calculated from summary advertisements originated by area border routers for IP subnetworks, IP subnetwork ranges, and autonomous system external (ASE) boundary routers located in other areas of the OSPF domain. External routes are calculated from ASE advertisements that are originated by ASE boundary routers and flooded throughout the OSPF routing domain.

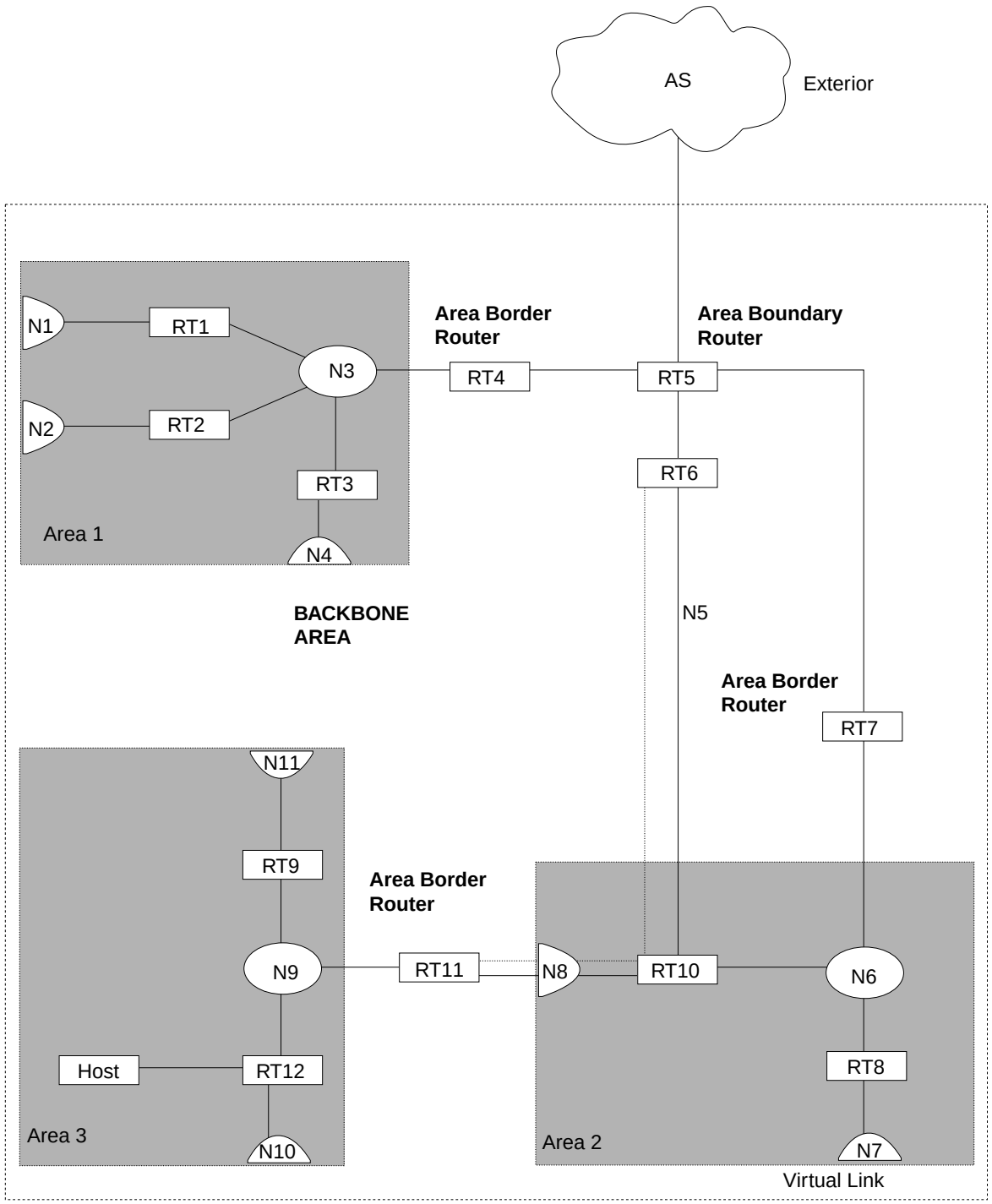


Figure 16-1. OSPF Areas

The backbone is responsible for distributing inter-area routing information. The backbone area consists of any of the following:

- Networks belonging to Area 0.0.0.0
- Routers attached to those networks
- Routers belonging to multiple areas
- Configured virtual links

The **set area** command is used to define areas that a router attaches to. If no set area command is used, the default is that all interfaces of the router attach to the backbone.

When area border routers are configured, options on the **set area** and **add range** commands can be used to control what OSPF route information crosses the area boundary.

One option is to use the set area command to define an area as a *stub*. ASE advertisements are never flooded into stub areas. In addition, the set area command has an option to suppress origination into the stub of summary advertisements for inter-area routes. Area border routers advertise default routes into stub areas. Traffic within the stub destined for unknown IP subnets is forwarded to the area border router. The border router uses its more complete routing information to forward the traffic on an appropriate path toward its destination. An area cannot be configured as a stub if it is used as a transit area for virtual links.

The other option is to use IP subnet address ranges to limit the number of summary advertisements that are used for inter-area advertisements of an area's subnets. A range is defined by an IP address and an address mask. Subnets are considered to fall within the range if the subnet IP address and the range IP address match after the range mask has been applied to both addresses. When a range is added for an area at an area border router, the border router suppresses summary advertisements for subnets in the areas that are included in the range. The suppressed advertisements would have been originated into the other areas that the border router attaches to. Instead, the area border router may originate a single summary advertisement for the range or no advertisement at all, depending on the option chosen with the add range command. Note that if the range is not advertised, there will be no inter-area routes for any destination that falls within the range. Also note that ranges cannot be used for areas that are used as transit areas by virtual links.

To set the parameters for an OSPF area, use the **set area** command and respond to the following prompts:

```
OSPF Config> set area
Area number [0.0.0.0]? 0.0.0.1
Authentication type [1]? 1
Is this a stub area?  [No]:
```

- Stub area designation

  Define an area as a stub when:

  1. There is no requirement for the area to handle transit backbone traffic.
  2. It is acceptable for area routers to use an area-border-router-generated default for traffic destined outside the AS.

  In this case, only the area border routers will have to store AS external routes.

Setting OSPF Interfaces

OSPF interfaces are a subset of the IP interfaces defined during IP configuration. The parameters configured for OSPF interfaces determine the topology of the OSPF domain, the routes that will be chosen through the domain, and the characteristics of the interaction between directly connected OSPF routers. The **set interface** command is used to define an OSPF interface and to specify some of its characteristics. Other characteristics of the interface were specified in response to the **add address** prompt during IP configuration.

OSPF Domain Topology

The definition of the topology of an OSPF domain depends on a definition of which routers are directly connected across some physical media or subnetwork technology and the area that those connections are part of. The basic case is for all routers attached to a physical subnetwork to be directly connected, but it is possible to define multiple IP subnetworks over a single subnetwork. In that case, OSPF will consider routers to be directly connected only when they have OSPF interfaces attached to the same IP subnetwork. It is also possible to have cases where routers attached to the same subnetwork do not have a direct link layer connection.

For LAN media, directly connected OSPF routers are determined from the IP subnetwork and physical media associated with an OSPF interface. The IP address of the OSPF interface is specified in response to the **Interface IP address** prompt. This address must match the address of an IP interface that was defined with the **add address** command during IP configuration. The IP address, along with the subnetwork mask defined with the **add address** command determine the IP subnetwork that the OSPF interface attaches to. The *net index* associated with the IP interface by the add address command determines the physical subnetwork to which the OSPF interface attaches. The broadcast capability of LANs allows OSPF to use multicast hello messages to discover other routers that have interfaces attached to the same IP subnetwork. Consequently, the interface parameters are all that are required for OSPF to determine which routers are directly connected across a LAN.

LANs may be used to connect an OSPF router with IP hosts. In this case, it is still necessary to define an OSPF interface to any IP subnetwork that is defined for the LAN. Otherwise, OSPF will not generate routes with those IP subnetworks as destinations.

The requirements for configuring OSPF interfaces that attach to serial lines vary with the lower layer technology.

For point-to-point lines, there is only one other router that is accessible over the interface, so the directly connected router can be determined without additional configuration. In fact, because there is no requirement to configure an IP subnetwork at all, unnumbered OSPF interfaces can be used for point-to-point lines. In this case, the same net index used as the IP address for the IP add address command is used as the IP address for the OSPF set interface command.

For subnetwork technologies like Frame Relay, ATM, and X.25 that support connections to multiple routers over a single serial line, the configuration of the OSPF interfaces is similar to that for a LAN, but because directly connected

routers are not discovered dynamically for these subnetwork technologies, additional configuration is required to specify directly connected neighbors. For more information on the required configuration, see “Configuring Wide Area Subnetworks” on page 16-12.

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.

Use a cost of 1 for both 155 Mbps ATM and 100 Mbps ATM. When using 155M as the cost of 1, use the cost of 100M = 2, Ethernet 10M = 16, TR 16M = 10, TR 4M = 39. Set the cost to 1 for 100 Mbps. With this approach, Ethernet interfaces would be configured with a cost of 10, 16 Mbps token-ring interfaces would be configured with a cost of 6, 4 Mbps token-ring interfaces would be configured with a cost of 25, and serial line interfaces would be configured with a cost that depends on their bandwidth.

Note: An Emulated Token-Ring or Ethernet will run at the interface speed (for example, 155 Mbps), and should be configured accordingly (with a cost of one).

ATM has the ability for attaching 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 host 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
- Authentication key

In most cases, the default values can be used.

Note: The hello interval, the dead router interval, and the authentication key must have the same value for all OSPF routers that attach to the same IP subnetwork. If the values are not the same, routers will fail to form direct connections (adjacencies).

Multicast Forwarding

To enable the routing of IP multicast (class D) datagrams, use the **enable multicast-routing** command. When enabling multicast routing, you will also be prompted as to whether you want the router to forward multicasts between OSPF areas.

```
OSPF Config>enable multicast forwarding
Inter-area multicasting enabled?  [No]: yes
```

When the **enable multicast forwarding** command is first invoked, multicast is enabled on all OSPF interfaces with default parameters.

If you want to change the MOSPF parameters, use the **set interface** command. You will be queried for multicast parameters only if you have first enabled multicast forwarding.

Setting Non-Broadcast Network Interface Parameters

If the router is connected to a non-broadcast, multi-access network, such as an ATM Network running RFC 1577, you have to configure the following parameters to help the router discover its OSPF neighbors. This configuration is necessary only if the router will be eligible to become designated router of the non-broadcast network.

First configure the OSPF poll interval with the following command:

```
OSPF Config> set non-broadcast
Interface IP address [0.0.0.0]? 128.185.138.19
Poll Interval [120]?
```

Then configure the IP addresses of all other OSPF routers that will be attached to the non-broadcast network. For each router configured, you must also specify its eligibility to become the designated router.

```
OSPF Config> add neighbor
Interface IP address [0.0.0.0]? 128.185.138.19
IP Address of Neighbor [0.0.0.0]? 128.185.138.21
Can that router become Designated Router [Yes]?
```

Configuring Wide Area Subnetworks

Frame Relay, Classical IP over ATM, and X.25 allow direct connections between multiple routers over a single serial line. Additional configuration beyond that achieved with the **set interface** command is required for OSPF interfaces that attach to this kind of network. Because OSPF protocol messages are sent directly to specific neighbors on these networks, configuration is used instead of dynamic discovery to determine neighbor relationships and router roles.

Note: The configurations described in this section do not apply to point-to-point networks.

OSPF can assume either of two patterns for the direct connections between routers across these subnetworks:

- Point-to-Multipoint
- Non-broadcast multiaccess (NBMA)

The key factor that distinguishes these two patterns is whether or not there is a direct connection between all pairs of routers that attach to the subnetwork (*full mesh connectivity*) or whether some of the routers are only connected through multihop paths with other routers as intermediates (*partial mesh connectivity*).

Non-broadcast multiaccess (NBMA) requires *full mesh connectivity* while point-to-multipoint requires only *partial mesh connectivity*.

Point-to-multipoint is the default choice because it works for both full mesh connectivity and partial mesh connectivity. But when full mesh connectivity is available, NBMA is a more efficient solution.

Configuring Point-to-Multipoint Subnetworks

Point-to-multipoint can be configured more easily than NBMA because there are no DRs, but neighbor relationships must be configured for all pairs of routers that will exchange data traffic directly across the point-to-multipoint subnet. Each pair of directly connected routers will exchange hello messages, so one side can discover the other through these messages. The router configured to send the first hello message, however, must have the IP address of its neighbor configured using the **add neighbor** command.

It is important to remember that OSPF will not calculate the correct routes if some of the routers attached to a subnetwork represent it as NBMA and others represent it as point-to-multipoint. Therefore, it is important that the **set non-broadcast** command is never used for any interface to a point-to-multipoint network.

Configuring NBMA Subnetworks

For NBMA IP subnetworks, some subset of the attached OSPF routers are configured to be eligible to be the designated router (DR). Each router eligible to be the DR periodically sends hello messages to all other routers eligible to be the DR. These messages are used in the protocol to elect a DR and a backup DR. Both the DR and the backup DR periodically exchange hello messages with all other OSPF routers that are attached to the NBMA IP subnetwork. Also, the flow of OSPF route information across the NBMA IP subnetwork is only between each of the attached routers and the DR or backup DR.

NBMA is selected by using the **set non-broadcast** command for interfaces that attach to an NBMA subnetwork. This command must be used for all interfaces that attach to the NBMA network.

The configuration required for an OSPF router that attaches to an NBMA subnetwork depends on whether or not that router is eligible to become the DR.

- For a router not eligible to become a DR, the **set interface** command must be used to set the router priority to 0.
- For a router eligible to become a DR, the **set interface** command must be used to set the router priority to a nonzero value and the **add neighbor** command must be used to identify all of the OSPF routers with interfaces attached to the NBMA subnetwork and to indicate which them are eligible to become DR.

Note: In a star configuration, use the **add neighbor** command at the hub (neighbors at the remote site do not need to be configured). The **add neighbor** command takes effect immediately without restarting the router.

Enabling AS Boundary Routing

To import routes learned from other protocols (RIP and statically configured information) into the OSPF domain, enable AS boundary routing. You must do this even if the only route you want to import is the default route (destination 0.0.0.0).

When enabling AS boundary routing, you are asked which external routes you want to import. You can choose to import, or not to import, routes belonging to several categories. The categories are as follows:

- BGP routes
- RIP routes
- Static routes
- Direct routes

For example, you can choose to import BGP and direct routes, but not RIP or static routes.

Independently of the above external categories, you can also configure whether or not to import subnet routes into the OSPF domain. This configuration item defaults to OFF (subnets not imported).

The metric type used in importing routes determines how the imported cost is viewed by the OSPF domain. When comparing two type 2 metrics, only the external cost is considered in picking the best route. When comparing two type 1 metrics, the external and internal costs of the route are combined before making the comparison. For example, you can set the router so that its default is originated only if a route to 10.0.0.0 is received from AS number 12. Setting the AS number to 0 means “from any AS.” Setting the network number to 0.0.0.0 means “any routes received.”

The syntax of the **enable** command is as follows:

```
OSPF Config>enable as boundary
Import BGP routes? [No]: yes
Import RIP routes? [No]:
Import static routes? [No]:
Import direct routes? [No]: yes
Import subnet routes? [No]:
Always originate default route? [No]: yes
Originate as type 1 or 2 [2]? 2
Default route cost [1]?
Default forwarding address [0.0.0.0]? 10.0.0.0
```

Other Configuration Tasks

Setting Virtual Links

To maintain backbone connectivity, you must have all of your backbone routers interconnected either by permanent or virtual links. You can configure virtual links between any two area border routers that share a common non-backbone and non-stub area. Virtual links are considered to be separate router interfaces connecting to the backbone area. Therefore, you are asked to also specify many of the interface parameters when configuring a virtual link.

The following example illustrates the configuration of a virtual link. Virtual links must be configured in each of the link's two endpoints. Note that you must enter OSPF router IDs in the same form as IP addresses.

```
OSPF Config> set virtual
Virtual endpt.  (Router ID) [0.0.0.0]? 128.185.138.21
Link's transit area [0.0.0.1]?
Retransmission Interval (in seconds) [10]?
Transmission Delay (in seconds) [5]?
Hello Interval (in seconds) [30]?
Dead Router Interval (in seconds) [180]?
Authentication Key []? 3-14159
```

Configuring for Routing Protocol Comparisons

If you use a routing protocol in addition to OSPF, or when you change your routing protocol to OSPF, you must set the Routing Protocol Comparison.

OSPF routing in an AS occurs on these three levels: intra-area, inter-area, and exterior.

Intra-area routing occurs when a packet's source and destination address reside in the same area. Information that is about other areas does not affect this type of routing.

Inter-area routing occurs when the packet's source and destination addresses reside in different areas of the same AS. OSPF does inter-area routing by dividing the path into three contiguous pieces: an intra-area path from source to an area border router; a backbone path between the source and destination areas; and then another intra-area path to the destination. You can visualize this high-level of routing as a star topology with the backbone as hub and each of the areas as a spoke.

Exterior routes are paths to networks that lie outside the AS. These routes originate either from routing protocols, such as Border Gateway Protocol (BGP), or from static routes entered by the network administrator. The exterior routing information provided by BGP does not interfere with the internal routing information provided by the OSPF protocol.

AS boundary routers can import exterior routes into the OSPF routing domain. OSPF represents these routes as AS external link advertisements.

OSPF imports external routes in separate levels. The first level, called type 1 routes, is used when the external metric is comparable to the OSPF metric (for example, they might both use delay in milliseconds). The second level, called external type 2 routes, assumes that the external cost is greater than the cost of any internal OSPF (link-state) path.

Imported external routes are tagged with 32 bits of information. In a router, this 32-bit field indicates the AS number from which the route was received. This enables more intelligent behavior when determining whether to re-advertise the external information to other Autonomous systems.

OSPF has a 4-level routing hierarchy (see Figure 16-2). 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.

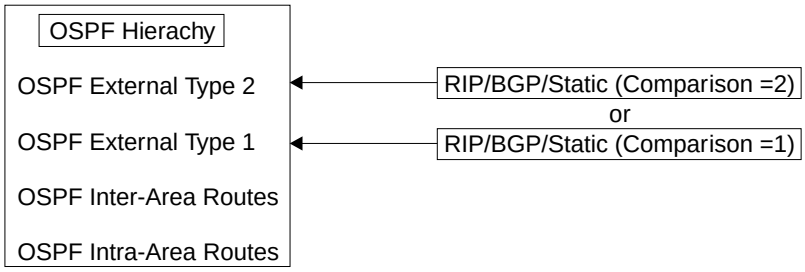


Figure 16-2. OSPF Routing Hierarchy

To put the BGP/RIP/static routes on the same level as OSPF external type 1 routes, set the comparison to 1. To put the BGP/RIP/static routes on the same level as OSPF external type 2 routes, set the comparison to 2. The default setting is 2.

For example, suppose the comparison is set to 2. In this case, when RIP routes are imported into the OSPF domain, they will be imported as type 2 externals. All OSPF external type 1 routes override received RIP routes, regardless of metric. However, if the RIP routes have a smaller cost, the RIP routes override OSPF external type 2 routes. The comparison values for all of your OSPF routers must match. If the comparison values set for the routers are inconsistent, your routing will not function correctly.

The syntax of the **set comparison** command is as follows:

```
OSPF Config> set comparison
Compare to type 1 or 2 externals [2]?
```

Converting from RIP to OSPF

To convert your Autonomous System from RIP to OSPF, install OSPF one router at a time, leaving RIP running. Gradually, all your internal routes will shift from being learned via RIP to being learned by OSPF (OSPF routes have precedence over RIP routes). If you want to have your routes look exactly as they did under RIP (in order to check that the conversion is working correctly) use hop count as your OSPF metric. This is done by assigning the cost of each OSPF interface to 1.

Remember that the size of your OSPF system must be estimated when the protocol is enabled. This size estimate should reflect the final size of the OSPF routing domain.

After installing OSPF on your routers, turn on AS boundary routing in all those routers that still need to learn routes via other protocols (BGP, RIP, and statically configured routes). The number of these AS boundary routers should be kept to a minimum.

Finally, you can disable the receiving of RIP information on all those routers that are not AS boundary routers.

Dynamically Changing Interface Costs

The cost of an OSPF interface can be dynamically changed from the router's console interface. This new cost is flooded quickly throughout the OSPF routing domain, and modifies the routing immediately.

When the router restarts/reloads, the cost of the interface reverts to the value that has been configured in SRAM.

Accessing the OSPF Configuration Environment

To access the OSPF configuration environment, enter the following command at the Config> prompt:

```
Config> protocol ospf
Open SPF-based Routing Protocol configuration console
OSPF Config>
```

OSPF Configuration Commands

Before you can use OSPF, you must configure it using the OSPF configuration commands. The following section summarizes and then explains the OSPF commands. Enter these commands at the OSPF config> prompt. Table 16-1 shows the commands.

Table 16-1. OSPF Configuration Command Summary

Command	Function
? (Help)	Lists the OSPF configuration commands or lists the options associated with specific commands.
Add	Adds to already existent OSPF information. You can add ranges to areas, and neighbors to non-broadcast networks.
Delete	Deletes OSPF information from SRAM.
Disable	Disables the entire OSPF protocol, AS boundary routing capability, or IP multicast routing.
Enable	Enables the entire OSPF protocol, AS boundary routing capability, or IP multicast routing.
Join	Configures the router to belong to one or more multicast groups.
Leave	Removes the router from membership in multicast groups.
List	Displays OSPF configuration.
Set	Establishes or changes the configuration information concerning OSPF areas, interfaces, non-broadcast networks, or virtual links. This command also allows you to set the way in which OSPF routes are compared with information gained from other routing protocols.
Exit	Exits the OSPF configuration process.

? (Help)

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

Syntax: ?

Example: ?

```
add
delete
disable
enable
exit
join
leave
list
set
```

Add

Use the **add** command to add more information to already existing OSPF information. With this command you can add ranges to areas as well as neighbors to non-broadcast networks.

Syntax: add range . . .
neighbor . .

range *area# IP-address IP-address-mask*

Adds ranges to OSPF areas. OSPF areas can be defined in terms of address ranges. External to the area, a single route is advertised for each address range. For example, if an OSPF area were to consist of all subnets of the class B network 128.185.0.0, it would be defined as consisting of a single address range. The address range would be specified as an address of 128.185.0.0 together with a mask of 255.255.0.0. Outside of the area, the entire subnetted network would be advertised as a single route to network 128.185.0.0.

Ranges can be defined to control which routes are advertised externally to an area. There are two choices:

- When OSPF is configured to advertise the range, a single inter-area route is advertised for the range if at least one component route of the range is active within the area.
- When OSPF is configured not to advertise the range, no inter-area routes are advertised for routes that fall within the range.

Ranges cannot be used for areas that serve as transit areas for virtual links. Also, when ranges are defined for an area, OSPF will not function correctly if the area is partitioned but is connected by the backbone.

Example: `add range 0.0.0.2 128.185.0.0 255.255.0.0`

1. The *IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

2. The *IP address mask* has:

   Valid Values: Any valid IP address mask.

   Default Value: none

neighbor

Configures neighbors adjacent to the router over this interface. Non-broadcast multi-access networks, neighbors need configured only to those routers that are eligible to become the designated router. In point-to-multipoint networks, at least one end of every logical connection must have a configured neighbor.

Example: **add neighbor**

```
Interface IP address [0.0.0.0]? 128.185.138.19
IP Address of Neighbor [0.0.0.0]? 128.185.138.21
Can that router become Designated Router on this net [Yes]?
```

1. The *Interface IP address* has:

   Valid Values: Any valid IP address. The last octet must be a zero.

   Default Value: none

2. The *IP Address of Neighbor* has:

   Valid Values: Any valid IP address.

   Default Value: none

3. Answer the question, *Can that router become designated router on this area?*.

Delete

Use the delete command to delete OSPF information from SRAM.

Syntax: delete range . . .
area . . .
interface . . .
neighbor . . .
non-broadcast . . .
virtual-link

range *area# IP-address*

Deletes ranges from OSPF areas.

Example: **delete range 128.185.0.0 255.255.0.0**

1. The *area number* of the range has:

   Valid Values:

   Default Value: none

2. The *IP Address of Neighbor* has:

   Valid Values: Any valid IP address. The last octet must be a zero.

   Default Value: none

3. The *IP Address Mask of Neighbor* has:

   Valid Values: Any valid IP address mask.

   Default Value: none

area *area#*

Deletes OSPF areas from the current OSPF configuration.

Example: **delete area 0.0.0.1**

interface *interface-IP-address*

Deletes an interface from the current OSPF configuration.

Example: **`delete interface 128.185.138.19`**

The *interface IP address* has:

Valid Values: Any valid IP address.

Default Value: none

neighbor

Deletes configured neighbors from the current OSPF configuration.

Example: **`delete neighbor`**

```
Interface IP address [0.0.0.0]? 128.185.138.19
IP Address of Neighbor [0.0.0.0]? 128.185.138.21
```

1. The *interface IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

2. The *neighbor IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

non-broadcast *interface-IP-address*

Deletes non-broadcast network information from the current OSPF configuration.

Example: **`delete non-broadcast 128.185.133.21`**

1. The *interface IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

virtual-link

Deletes a virtual link that you have set using the **set virtual-link** command.

Example: **`delete virtual-link`**

```
Virtual endpoint (Router ID) [0.0.0.0]?
Link's transit area [0.0.0.1]?
```

1. The *virtual endpoint (router ID)* that defines the id of the virtual neighbor has:

   Valid Values: Any valid IP address.

   Default Value: none

2. The *link's transit area* has:

   Valid Values: Any valid IP address.

   Default Value: 0.0.0.0

Disable

Use the **disable** command to disable either the entire OSPF protocol or just the AS boundary routing capability.

Syntax: disable as boundary routing
multicast forwarding
OSPF routing protocol

as boundary routing
Disables the AS boundary routing capability. When disabled, the router will not import external information into the OSPF domain.

Example: **disable as boundary routing**

multicast forwarding
Disables IP multicast routing on all interfaces. When disabled, the router will not forward IP multicast (Class D) datagrams.

Example: **disable multicast forwarding**

OSPF routing protocol
Disables the entire OSPF protocol.

Example: **disable OSPF routing protocol**

Enable

Use the **enable** command to enable either the entire OSPF protocol, the advertisement of a stub to route to a subnet, or just the AS boundary routing capability.

Syntax: enable as boundary routing
multicast forwarding
OSPF routing protocol
subnet

as boundary routing
Enables the AS boundary routing capability which allows you to import routes learned from other protocols (BGP, RIP, and statically configured information) into the OSPF domain. For additional information on the use of the **enable** command, see “Configuring OSPF” on page 16-4.

Example: **enable as boundary routing**

```
Import BGP routes? [No]:  yes
Import RIP routes? [No]:
Import static routes? [No]:
Import direct routes? [No]: yes
Import subnet routes? [No]:
Always originate default route? [No]: yes
Originate as type 1 or 2 [2]? 2
Default route cost [1]?
Default forwarding address [0.0.0.0]? 10.0.0.0
```

1. The *Default route cost* is the parameter that specifies the cost that OSPF associates with the default route to its area border router. The cost is used to determine the shortest path for the default route to its area border router.

   Valid Values: 0 to 16777215

   Default Value: 1

2. The *Default forwarding address* is the parameter that specifies the forwarding address that will be used in the imported default route.

   Valid Values: a valid IP address

   Default Value: none

multicast forwarding
: Enables the forwarding of IP multicast (Class D) datagrams. When enabling multicast routing, you are also prompted whether you want to forward IP multicast datagrams between OSPF areas and between Autonomous Systems. To run MOSPF (OSPF with multicast extensions), a router currently running OSPF needs only to use this command. You do not need to reenter its configuration information.

  Example: **`enable multicast forwarding`**

```
Inter-area multicasting enabled (Yes or No): yes
```

OSPF routing protocol
: Enables the entire OSPF protocol. When enabling the OSPF routing protocol, you must supply the following two values that will be used to estimate the size of the OSPF link state database:

  - Total number of AS external routes that will be imported into the OSPF routing domain. A single destination may lead to multiple external routes when it is imported by separate AS boundary routers. For example, if the OSPF routing domain has two AS boundary routers, both importing routes to the same 100 destinations, the number of AS external routes should be set to 200.

    Valid Values: 0 to 65535

    Default Value: 100

  - Total number of OSPF routers in the routing domain.

    Valid Values: 0 to 65535

    Default Value: 50

  Example: **`enable OSPF routing protocol`**

```
Estimated # external routes[0]? 200
Estimated # OSPF routers [0]? 60
```

subnet
: For an interface to a point-to-point serial line, this option enables the advertisement of a stub route to the subnet that represents the serial line rather than the host route for the other router's address. You must supply this router's address for the interface to identify it.

  Example:

```
OSPF Config> enable subnet
Interface IP address [0.0.0.0]? 8.24.3.1
```

  The *interface IP address* has:

  Valid Values: Any valid IP address.

  Default Value: none

Join

Use the **join** command to configure the router as a member of a multicast group. When the router is the member of a multicast group, it responds to PINGS and SNMP queries sent to the group address.

To request group membership in a more temporary and more immediate way (a restart/reload is not required), issue the **join** command from the OSPF monitoring console. Also, from the OSPF monitoring console, the join command keeps track of the number of times a particular group is joined.

Syntax: join *multicast-group-address*

Example: `join 224.185.0.0`

The *group address* parameter specifies the 6-byte (12-digit hexadecimal) group/multicast address.

Valid Values: class D IP address from 224.0.0.1 to 239.255.255.255

Default Value: none

Leave

Use the **leave** command to remove a router's membership from a multicast group. This will prevent the router from responding to PINGS and SNMP queries sent to the group address.

To delete group membership in a more immediate way (a restart/reload is not required), issue the **leave** command from the OSPF monitoring console. Also, from the OSPF monitoring console, group membership is not deleted until the number of leaves executed equals the number of joins previously executed.

Syntax: leave *multicast-group-address*

Example: `leave 224.185.0.0`

The *address to be deleted* has:

Valid Values: Any valid IP address.

Default Value: none

List

Use the **list** command to display OSPF configuration information.

Syntax: list all
areas
interfaces
neighbors
non-broadcast
virtual-links

all

Lists all OSPF-related configuration information.

Example: `list all`

```
      --Global configuration--
    OSPF Protocol:          Enabled
    # AS ext. routes:       300
    Estimated # routers:    100
    External comparison:    Type 2
    AS boundary capability: Enabled
    Import external routes: BGP RIP STA DIR SUB
    Orig. default route:    No (0,0.0.0.0)
    Default route cost:     (1, Type 2)
    Default forward. addr.: 0.0.0.0
    Multicast forwarding:   Enabled
    Inter-area multicast:   Enabled

           --Area configuration--
Area ID          AuType          Stub? Default-cost Import-summaries?
0.0.0.0          0=None          No         N/A           N/A


      --Interface configuration--
    IP address        Area    Cost  Rtrns  TrnsDly  Pri  Hello Dead
    128.185.184.11    0.0.0.1   1      5       1     1    10    60
    128.185.177.11    0.0.0.1   1      5       1     1    10    60
    128.185.142.11    0.0.0.0   1      5       1     1    10    60
```

OSPF protocol — Displays whether OSPF is enabled or disabled.

 # AS ext. routes — Displays the estimated number of Autonomous System external routes. The router cannot accept more than this number of AS external routes.

Estimated # routers — Displays the estimated number of routers found in the OSPF configuration.

External comparison — Displays the external route type used by OSPF when importing external information into the OSPF domain and when comparing OSPF external routes to RIP/BGP routes.

AS boundary capability — Displays whether the router will import external routes into the OSPF domain.

Import external — Displays which routes will be imported.

Orig default route — Displays whether the router will import a default into the OSPF domain. When the value is “YES,” a nonzero network number is displayed in parentheses. This indicates that the default route will originated only if a route to that network is available.

Default route cost — Displays the cost and type that will be used in the imported default route.

Default forward addr — Displays the forwarding address that will be used in the imported default route.

Multicast forwarding — Displays whether IP multicast datagrams will be forwarded.

Inter-area multicast — Displays whether IP multicast datagrams will be forwarded between areas.

Area-ID — Displays the attached area ID (area summary information)

AuType — Displays the method used for area authentication. “Simple-pass” means a simple password scheme is being used for the area’s authentication.

Stub area — Displays whether or not the area being summarized is a stub area. Stub areas do not carry external routes, resulting in a smaller routing database. However, stub areas cannot contain AS boundary routers, nor can they support configured virtual links.

OSPF interfaces — For each interface, its IP address is printed, together with configured parameters. "Area" is the OSPF area to which the interface attaches. "Cost" indicates the TOS 0 cost (or metric) associated with the interface. "Rtrns" is the retransmission interval, which is the number of seconds between retransmissions of unacknowledged routing information. "TrnsDly" is the transmission delay, which is an estimate of the number of seconds it takes to transmit routing information over the interface (it must be greater than 0). "Pri" is the interface's Router Priority, which is used when selecting the designated router. "Hello" is the number of seconds between Hello Packets sent out the interface. "Dead" is the number of seconds after Hellos cease to be heard that the router is declared down.

Virtual links — Lists all virtual links that have been configured with this router as endpoint. "Virtual endpoint" indicated the OSPF Router ID of the other endpoint. "Transit area" indicates the non-backbone area through which the virtual link is configured. Virtual links are considered treated by the OSPF protocol similarly to point-to-point networks. The other parameters listed in the command ("Rtrns," "TrnsDly," "Hello," and "Dead") are maintained for all interfaces. See the OSPF list interfaces command for more information.

areas

Lists all information concerning configured OSPF areas.

Example: `list areas`

```
                              --Area configuration--
Area ID          AuType          Stub? Default-cost Import-summaries?
0.0.0.0          0=None           No         N/A           N/A
0.0.0.1          1=Simp-Pass      No         N/A           N/A
```

Area-ID — Displays the attached area ID (area summary information).

AuType — Displays the method used for area authentication. "Simple-pass" means a simple password scheme is being used for the area's authentication.

Stub area — Displays whether or not the area being summarized is a stub area.

interfaces

For each interface, its IP address is printed, together with configured parameters. "Area" is the OSPF area to which the interface attaches. "Cost" indicates the TOS 0 cost (or metric) associated with the interface. "Rtrns" is the retransmission interval, which is the number of seconds between retransmissions of unacknowledged routing information. "TrnsDly" is the transmission delay, which is an estimate of the number of seconds it takes to transmit routing information over the interface (it must

be greater than 0). “Pri” is the interface’s router priority, which is used when selecting the designated router. “Hello” is the number of seconds between Hello Packets sent out the interface. “Dead” is the number of seconds after Hellos cease to be heard that the router is declared down.

Example: `list interfaces`

```
                --Area configuration--
IP address        Area      Cost  Rtrns  TrnsDly  Pri Hello Dead
128.185.208.43    1.1.1.1   1      5       1       1    10   40
10.1.155.43       0.0.0.0   1      5       1       1    10   40
10.1.152.43       0.0.0.0   1      5       1       1    10   40


                --Multicast Parameters--
IP address        MCForward   DLUnicast  IGMPoll  IGMPtimeout
128.185.208.43    On          Off        60       180
10.1.155.43       On          Off        60       180
10.1.152.43       On          Off        60       180
```

Note: Multicast parameters are not displayed if multicast is disabled.

neighbors

Lists neighbors to non-broadcast networks. It displays IP address of the neighbor and the IP address of the interface to that neighbor. It also indicates whether the neighbor is eligible to become the “Designated Router” on the net.

Example: `list neighbors`

```
          --Neighbor configuration--
Neighbor Addr     Interface Address   DR eligible?
2.3.4.5           1.2.3.4             yes
2.5.6.7           5.6.7.8             no
```

non-broadcast

Lists all information related to interfaces connected to non-broadcast multi-access networks. For each non-broadcast interface, as long as the router is eligible to become designated router on the attached network, the polling interval is displayed together with a list of the router’s neighbors on the non-broadcast network.

Example: `list non-broadcast`

```
          --NBMA configuration--
Interface Addr      Poll Interval
128.185.235.34      120
```

virtual-links

Lists all virtual links that have been configured with this router as endpoint. “Virtual endpoint” indicated the OSPF router ID of the other endpoint. “Transit area” indicates the non-backbone area through which the virtual link is configured. Virtual links are considered treated by the OSPF protocol similarly to point-to-point networks. The other parameters listed in the command (“Rtrns,” “TrnsDly,” “Hello,” and “Dead”) are maintained for all interfaces. See the OSPF **list interfaces** command for more information.

Example: `list virtual-links`

```
             --Virtual link configuration--
Virtual endpoint   Transit area   Rtrns  TrnsDly Hello Dead
0.0.0.0            0.0.0.1         10      5     30    180
```

Set

Use the **set** command to display or change the configuration information concerning OSPF areas, interfaces, non-broadcast networks, or virtual links. This command also allows you to set the way in which OSPF routes are compared to information obtained from other routing protocols.

Syntax: set area
comparison
interface
non-broadcast
virtual-link

area

Sets the parameters for an OSPF area. If no areas are defined, the router software assumes that all the router's directly attached networks belong to the backbone area (area ID 0.0.0.0).

Example: **set area**

```
Area number [0.0.0.0]? 0.0.0.1
Authentication type [1]? 1
Is this a stub area? [No]:  yes
Stub default cost? [0]:
Import summaries? [Yes]:
```

- *Area number* - is the OSPF area address.
- *Authentication type* - (security scheme) to be used in the area. The choices for authentication types are 1, which indicates a simple password; or 0, which indicates that no authentication is necessary to pass packets.

  All OSPF routers attached to the same subnet must have the same Authentication Key. For example, suppose the address mask for this network interface is 255.255.255.0, the IP address is 128.185.138.19 and the authentication key is *xyz123*. According to the subnet mask and IP address combination, the interface attaches to the subnet 128.185.138.0 of network 128.185.0.0. All other OSPF routers attached to subnet 128.185.138.0 must have their authentication key set to *xyz123*.

  Valid Values: any 8 characters

  Default Value: 0

- *Stub area designation.* If you designate YES:
  - The area does not receive any AS external link advertisements, reducing the size of your database and decreasing memory usage for routers in the stub area.
  - You cannot configure virtual links through a stub area.
  - You cannot configure a router within the stub area as an AS boundary router.

  External Routing in Stub Areas. You cannot configure the backbone as a stub area. External routing in stub areas is based on a default route. Each border area router attaching to a stub area originates a default route for this purpose. The cost of this default route is also configurable with the **set area** command.

comparison

Tells the router where the BGP/RIP/static routes fit in the OSPF hierarchy. The two lower levels consist of the OSPF internal routes. OSPF internal routes take precedence over information gained from any other sources, all of which are located on a single level.

Example: `set comparison`

```
OSPF Config> set comparison
Compare to type 1 or 2 externals [2]?
```

interface

Sets the OSPF parameters for the router's network interfaces.

1. The *interface IP address* is for each interface in the router.
2. *attaches to area* is the area to which the interface attaches.
3. The timer values are the same values for all routers attached to a common network segment.
   a. The *retransmission interval* is the interval after which a Link Request for one or more link state advertisements will be re-sent.

      Valid values: 1 to 65535 seconds

      Default Value: 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 65535 seconds

      Default Value: 1

   c. 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

   d. The *Hello Interval* is the interval between hello packets sent on the interface.

      Valid Values: 1 to 65535 seconds

      Default Value: 10

   e. 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: 1 to >65535 seconds

      Default Value: 40 (or four times the configured hello interval)

4. The *Type of service 0 cost*.

   Valid Values: 1 to 65535

   Default Value: 1

5. The *Authentication key* is the parameter that defines the password used for this OSPF area. When password authentication is used, only packets with the correct authentication key are accepted.

   Valid Values: any 8 characters

   Default Value: 0

Example: **`set interface`**

```
Interface IP address [0.0.0.0]? 128.185.138.19
Attaches to area [0.0.0.0]? 0.0.0.1
Retransmission Interval (in seconds) [5]?
Transmission Delay (in seconds) [1]? 1
Router Priority [1]? 1
Hello Interval (in seconds) [10]? 10
Dead Router Interval (in seconds) [60]? 40
Type Of Service 0 cost [1]? 5
Authentication Key []? xyz_q
Retype Auth.  Key []? xyz_q
Forward multicast datagrams (Yes or No)? Yes
Forward as datalink unicasts (Yes or No)? No
IGMP polling interval (in seconds) [60]? 60
IGMP timeout (in seconds) [180]? 180
```

When responding to the prompts, supply the IP address for each interface in the router and answer the questions that follow. For the following parameters, you must enter the same value for all routers attached to a common network:

- Hello interval
- Dead router interval
- Authentication key (if an authentication of 1 is used)

The first prompt asks for the OSPF area to which the interface attaches. For example, suppose that the interface address mask is 255.255.255.0, indicating that the interface attaches to a subnet (128.185.138.0) of network 128.185.0.0. All other OSPF routers attached to subnet 128.185.138.0 must also have their *hello interval* set to 10, *dead router interval* set to 40, and their interface *authentication key* set to xyz_q.

Note that IP interfaces to point-to-point lines may be unnumbered. In this case a net index is configured instead of an IP address. This implementation of OSPF will work with these unnumbered interfaces, but to work correctly, both ends of the point-to-point line must use an unnumbered interface.

In a multicast routing configuration (multicast has been enabled), the MOSPF parameters for each OSPF interface are set to their default values. This means that:

- Multicast forwarding is enabled.
- Multicast datagrams are forwarded as data-link multicasts.
- IGMP Host Membership is sent out on the interface every 60 seconds.

- Local group database entries are removed 180 seconds after IGMP Host Membership reports for the group cease to be received by the interface.

If you want to change the MOSPF parameters, use the **set interface** command. You will be queried for multicast parameters (the last five parameters shown in the output display above) only if you have first enabled multicast forwarding.

On networks that lie on the edge of an Autonomous System, where multiple multicast routing protocols (or multiple instances of a single multicast routing protocol) may exist, you may need to configure forwarding as data-link unicasts to avoid unwanted datagram replication. In any case, for all routers attached to a common network, the interface parameters "forward multicast datagrams" and "forward as data-link unicasts" should be configured identically.

non-broadcast

Overrides the point-to-multipoint default to select NBMA for X.25, Frame Relay or ATM networks. You must set non-broadcast consistently across all interfaces that attach to the same subnetwork for OSPF to function correctly.

The *interface IP address* has:

Valid Values: Any valid IP address.

Default Value: none

The NBMA Poll Interval is used to send Hello packets to inactive neighbors. (Inactive neighbors are those neighbors that the router has not heard from for a period greater than the Dead Router interval.) The router still polls these neighbors at a reduced rate. Set the NBMA Poll Interval much higher than the configured Hello Interval for the router.

Valid Values: 1 to 65535 seconds

Default Value: 120 seconds

Example: **`set non-broadcast`**

```
Interface IP address [0.0.0.0]? 128.185.138.19
Poll Interval [120]?
```

virtual-link

Configures virtual links between any two area border routers. To maintain backbone connectivity you must have all of your backbone routers interconnected either by permanent or virtual links. Virtual links are considered to be separate router interfaces connecting to the backbone area. Therefore, you are asked to also specify many of the interface parameters when configuring a virtual link.

Virtual links can be configured between any two backbone routers that have an interface to a common non-backbone area. Virtual links are used to maintain backbone connectivity and must be configured at both endpoints.

Note: This OSPF implementation supports the use of virtual links when one end of the virtual link may be an unnumbered point to point line. For this configuration to work, the router id must be used as the source address in OSPF protocol messages sent over the virtual link. Use of the router id can be insured by configuring the internal IP address with the

address used as the router id. Another requirement for this configuration to work is that the OSPF implementations at both ends of the virtual link support it.

1. The *virtual endpoint (router ID)* defines the ID of the virtual neighbor.

   Valid Values: Any valid IP address.

   Default Value: none

2. The *link's transit area*. is the non-backbone, non-stub area through which the virtual link is configured. Virtual links can be configured between any two area border routers that have an interface to a common non-backbone and non-stub area. Virtual links must be configured in each of the link's two endpoints.

   Valid Values: 0.0.0.1 to 255.255.255.255

   Default Value: 0.0.0.1

3. The timer values are the same values for all routers attached to a common network segment.

   a. The *retransmission interval* is the interval after which a Link Request for one or more link state advertisements will be re-sent.

      Valid Values: 1 to 65535 seconds

      Default Value: 10

   b. The *Transmission delay* parameter is an estimate of the number of seconds that it takes to transmit link-state information over the interface.

      Each link-state advertisement has a finite lifetime that is equal to the constant MaxAge (1 hour). As each link-state advertisement is sent to the particular interfaces, it is aged by this configured transmission delay. The minimum delay is 1 second.

      Valid Values: 1 to 65535 seconds

      Default Value: 5

      The *Hello Interval* is the interval between hello packets sent on the interface.

      Valid Values: 1 to 255 seconds

      Default Value: 30

   c. The *Dead Router Interval* is the interval after which a router that has not sent a hello will be considered dead. This parameter defaults to six times the configured Hello Interval and must be set to a value greater than the Hello Interval.

      Valid Values: 1 to 65535 seconds

      Default Value: 180

   This parameter defines the password used for this OSPF area. When password authentication is used, only packets with the correct authentication key are accepted.

Example: `set virtual-link`

```
Virtual endpt.  (Router ID) [0.0.0.0]? 128.185.138.21
Link's transit area [0.0.0.1]? 0.0.0.1
Retransmission Interval (in seconds) [10]?
Transmission Delay (in seconds) [5]?
Hello Interval (in seconds) [30]?
Dead Router Interval (in seconds) [180]?
Authentication Key []? 314159
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Chapter 17. Monitoring OSPF

This chapter describes the OSPF console commands and contains the following sections:

- "Accessing the OSPF Console Environment"
- "OSPF Console Commands."

Accessing the OSPF Console Environment

For information on how to access the OSPF console environment, refer to *Getting Started (Introduction to the User Interface)* in the *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

OSPF Console Commands

This section summarizes and then explains all the OSPF console monitoring commands. These commands enable you to monitor the OSPF routing protocol. Table 17-1 lists the OSPF console commands.

Enter the OSPF console commands at the `OSPF>` prompt.

Table 17-1 (Page 1 of 2). OSPF Console Command Summary

Command	Function
? (Help)	Lists the OSPF console commands or lists the options associated with specific commands.
Advertisement	Displays a link state advertisement belonging to the OSPF database.
Area summary	Displays OSPF area statistics and parameters.
AS external	Lists the AS external advertisements belonging to the OSPF link state database.
Database summary	Displays the advertisements belonging to an OSPF area's link state database.
Dump routing tables	Displays the OSPF routes contained in the routing table.
Interface summary	Displays OSPF interface statistics and parameters.
Join	Configures the router to belong to one or more multicast groups.
Leave	Removes the router from membership in multicast groups.
Mcache	Displays a list of currently active multicast forwarding cache entries.
Mgroups	Displays the group membership of the router's attached interfaces.
Mstats	Displays various multicast routing statistics.
Neighbor summary	Displays OSPF neighbor statistics and parameters.

Table 17-1 (Page 2 of 2). OSPF Console Command Summary

Command	Function
Ping	Continuously sends ICMP Echo Requests (or pings) a given destination, printing a line for each response received.
Routers	Displays the reachable OSPF area-border routers and AS-boundary routers.
Size	Displays the number of LSAs currently in the link state database, categorized by type.
Statistics	Displays OSPF statistics detailing memory and network usage.
Traceroute	Displays the complete route (hop-by-hop) to a given destination.
Weight	Dynamically changes the cost of an OSPF interface.
Exit	Exits the OSPF console process.

? (Help)

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

Syntax: ?

Example: ?

```
ADVERTISEMENT expansion
AREA summary
AS-EXTERNAL advertisement
DATABASE summary
DUMP routing tables
EXIT
INTERFACE summary
JOIN
LEAVE
MCACHE
MGROUPS
MSTATS
NEIGHBOR summary
PING address
ROUTERS
SIZE
STATISTICS
TRACEROUTE
WEIGHT
```

Advertisement Expansion

Use the **advertisement expansion** command to print the contents of a link state advertisement contained in the OSPF database. For a summary of the router's advertisements use the **database** command.

A link state advertisement is defined by its link state type, link state ID and its advertising router. There is a separate link state database for each OSPF area. Providing an area-id on the command line tells the software which database you want to search. The different kinds of advertisements, which depend on the value given for link-state-type, are:

- Router links - Contain descriptions of a single router's interface.
- Network links - Contain the list of routers attached to a particular interface.
- Summary nets - Contain descriptions of a single inter-area route.
- Summary AS boundary routers - Contain descriptions of the route to an AS boundary router in another area.
- AS external nets - Contain descriptions of a single route.
- Multicast group memberships - Contain descriptions of a particular group's membership in the neighborhood of the advertising router.

  Note: Link State IDs, advertising routers (specified by their router IDs), and area IDs take the same format as IP addresses. For example, the backbone area can be entered as 0.0.0.0.

Example 1 shows an expansion of a router links advertisement. The router's ID is 128.185.184.11. It is an AS boundary router and has three interfaces to the backbone area (all of cost 1). Multicast routing has been enabled. Detailed field descriptions are provided with the example.

This command has also been enhanced in two ways. First of all, when displaying router-LSAs and network-LSAs, the reverse cost of each router-to-router link and router-to-transit-network link is displayed, as well as the previously displayed forward cost. This is done because routing of multicast datagrams whose source lies in different areas/Autonomous systems is based on reverse cost instead of forward cost. In those cases where there is no reverse link (which means that the link will never be used by the Dijkstra), the reverse cost is shown as “1-way.”

In addition, the LSA's OSPF options are displayed in the same manner as they were displayed in the detailed OSPF **neighbor** command.

New group-membership-LSAs can also be displayed. The “LS destination” of each group-membership-LSA is a group address. A router originates a group-membership-LSA for each group that has members on one or more of the router's attached networks. The group-membership-LSA for the group lists those attached transit networks having group members (the type “2” vertices), and when there are members belonging to one or more attached stub networks, or if the router itself is a member of the multicast group, a type “1” vertex whose ID is the router's OSPF router ID is included.

Syntax: advertisement *ls-type link-state-id [advertising-router] [area-id]*

Example 1: `advertisement 1 128.185.184.11 0.0.0.0`

```
LS age:     173
LS options:  E,MC
LS type:     1
LS destination (ID): 128.185.184.11
LS originator:   128.185.184.11
LS sequence no:  0x80000047
LS checksum:     0x122
LS length:       60
Router type:  ASBR,W
# router ifcs:   3
         Link ID:          128.185.177.31
         Link Data:        128.185.177.11
         Interface type:   2
                  No. of metrics: 0
```

```
         TOS 0 metric:   3 (0)
Link ID:          128.185.142.40
Link Data:        128.185.142.11
Interface type:   2
         No. of metrics: 0
         TOS 0 metric:   4 (0)
Link ID:          128.185.184.0
Link Data:        255.255.255.0
Interface type:   3
         No. of metrics: 0
         TOS 0 metric:   1
```

LS age	Indicates the age of the advertisement in seconds.
LS options	Indicates the optional OSPF capabilities supported by the piece of the routing domain described by the advertisement. These capabilities are denoted by E (processes type 5 externals; when this is not set to the area to which the advertisement belongs has been configured as a stub), T (can route based on TOS) and MC (can forward IP multicast datagrams).
LS type	Classifies the advertisement and dictates its contents: 1 (router links advertisement), 2 (network link advertisement), 3 (summary link advertisement), 4 (summary ASBR advertisement), 5 (AS external link) and 6 (group-membership advertisement).
LS destination	Identifies what is being described by the advertisement. Depends on the advertisement type. For router links and ASBR summaries, it is the OSPF router ID. For network links, it is the IP address of the network's designated router. For summary links and AS external links, it is a network/subnet number. For group-membership advertisements, it is a particular multicast group.
LS originator	OSPF router ID of the originating router.
LS sequence number	Used to distinguish separate instances of the same advertisement. Should be looked at as a signed 32-bit integer. Starts at 0x80000001, and increments by one each time the advertisement is updated.
LS checksum	A checksum of advertisement contents, used to detect data corruption.
LS length	The size of the advertisement in bytes.
Router type	Indicates the level of function of the router. ASBR means that the router is an AS boundary router, ABR that the router is an area border router, and W that the router is a wildcard multicast receiver.
# Router ifcs	The number of router interfaces described in the advertisement.
Link ID	Indicates what the interface connects to. Depends on Interface type. For interfaces to routers (i.e., point-to-point links), the Link ID is the neighbor's router ID. For interfaces to transit networks, it is the IP address of the network designated router. For interfaces to stub networks, it is the network's network/subnet number.
Link Data	4 bytes of extra information concerning the link, it is either the IP address of the interface (for interfaces to point-to-point networks and transit networks), or the subnet mask (for interfaces to stub networks).

Interface type	One of the following: 1 (point-to-point connection to another router, 2 (connection to transit network, 3 (connection to stub network) or 4 (virtual link).
No. of metrics	The number of non-zero TOS values for which metrics are provided for this interface.
TOS 0 metric	The cost of the interface. In parenthesis the reverse cost of the link is given (derived from another advertisement). If there is no reverse link, “1-way” is displayed.

The LS age, LS options, LS type, LS destination, LS originator, LS sequence no, LS checksum and LS length fields are common to all advertisements. The Router type and # router ifcs are seen only in router links advertisements. Each link in the router advertisement is described by the Link ID, Link Data, and Interface type fields. Each link can also be assigned a separate cost for each IP Type of Service (TOS); this is described by the No. of metrics and TOS 0 metric fields (the router currently does not route based on TOS, and looks at the TOS 0 cost only).

Example 2 shows an expansion of a group-membership advertisement. A group-membership advertisement for a given group/advertising router combination lists those networks directly attached to the advertising router which have group members. It also lists whether the router itself is a member of the specified group. The example below shows that network 128.185.184.0 has members of group 224.0.1.1.

Example 2: `adv 6 224.0.1.1 128.185.184.114`

```
For which area [0.0.0.0]?

LS age:     168
LS options:  E
LS type:     6
LS destination (ID): 224.0.1.1
LS originator:   128.185.184.114
LS sequence no:  0x80000001
LS checksum:     0x7A3
LS length:       28
Vertex type: 2
Vertex ID:   128.185.184.114
```

Vertex type	Describes the object having group members, one of: 1 (the router itself, or stub networks attached to the router) or 2 (a transit network).
Vertex ID	When the vertex type is 1, always the advertising router's ID. When the vertex type is 2, the IP address of the transit network's designated router.

Area Summary

Use the **area summary** command to display the statistics and parameters for all OSPF areas attached to the router.

In the example below, the router attaches to a single area (the backbone area). A simple password scheme is being used for the area's authentication. The router has three interfaces attaching to the area, and has found 4 transit networks, 7 routers and no area border routers when doing the SPF tree calculation for the backbone.

Syntax: area

Example: **area**

```
Area ID      Authentication   #ifcs  #nets  #rtrs  #brdrs
0.0.0.0       Simple-pass       3      4      7      0
```

 # ifcs Indicates the number of router interfaces attached to the particular area. These interfaces are not necessarily functional.

 # nets Indicates the number of transit networks that have been found while doing the SPF tree calculation for this area.

 # rtrs Indicates the number of routers that have been found when doing the SPF tree calculation for this area.

 # brdrs Indicates the number of area border routers that have been found when doing the SPF tree calculation for this area.

AS-external advertisements

Use the **AS-external advertisements** command to list the AS external advertisements belonging to the OSPF routing domain. One line is printed for each advertisement. Each advertisement is defined by the following three parameters: its link state type (always 5 for AS external advertisements), its link state ID (called the LS destination), and the advertising router (called the LS originator).

Syntax: as-external

Example: **as-external**

```
Type LS destination LS originator  Seqno      Age   Xsum
  5  0.0.0.0       128.185.123.22  0x80000084 430   0x41C7
  5  128.185.131.0 128.185.123.22  0x80000080 450  0x71DC
  5  128.185.132.0 128.185.123.22  0x80000080 450  0x66E6
  5  128.185.144.0 128.185.123.22  0x80000002 329  0xF2CA
  5  128.185.178.0 128.185.123.22  0x80000081 450  0x72AA
  5  128.185.178.0 128.185.129.40  0x80000080 382  0xDD28
  5  129.9.0.0     128.185.123.22  0x80000082 451  0x4F30
  5  129.9.0.0     128.185.126.24  0x80000080 676  0x324A
  5  134.216.0.0   128.185.123.22  0x80000082 451  0x505A
  5  134.216.0.0   128.185.126.24  0x80000080 676  0x3374
  5  192.9.3.0     128.185.123.22  0x80000082 451  0xF745
  5  192.9.3.0     128.185.126.24  0x80000080 677  0xDA5F
  5  192.9.12.0    128.185.123.22  0x80000082 452  0x949F
  5  192.9.12.0    128.185.128.41  0x80000080 679  0x31B2
  5  192.26.100.0  128.185.123.22  0x80000081 452  0xFDCD
  5  192.26.100.0  128.185.126.24  0x80000080  21  0xDEE8
                              etc.
             # advertisements:            133
             Checksum total:              0x43CC41
```

Type Always 5 for AS external advertisements.

LS destination Indicates an IP network/subnet number. These network numbers belong to other Autonomous Systems.

LS originator Advertising router.

Seqno, Age, Xsum

It is possible for several instances of an advertisement to be present in the OSPF routing domain at any one time. However, only the most recent instance is kept in the OSPF link state database (and printed by this command). The LS sequence number (Seqno), LS age (Age) and LS checksum fields (Xsum) are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600.

At the end of the display, the total number of AS external advertisements is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement's LS checksum fields. This information can be used to quickly determine whether two OSPF routers have synchronized databases.

Database Summary

Use the **database summary** command to display a description of the contents of a particular OSPF area's link state database. AS external advertisements are omitted from the display. A single line is printed for each advertisement. Each advertisement is defined by the following three parameters: its link state type (called Type), its link state ID (called the LS destination) and the advertising router (called the LS originator).

Syntax: database *area-id*

Example: **database 0.0.0.0**

```
Type LS destination LS originator    Seqno     Age  Xsum
  1  128.185.123.22 128.185.123.22 0x80000084  442 0xCE2D
  1  128.185.125.38 128.185.125.38 0x80000082  470 0x344D
  1  128.185.126.24 128.185.126.24 0x80000088 1394 0xCC47
  1  128.185.128.41 128.185.128.41 0x80000082  471 0x16A2
  1  128.185.129.25 128.185.129.25 0x8000008D 1624 0x8B64
  1  128.185.129.40 128.185.129.40 0x8000008A 1623 0xABBE
  1  128.185.136.39 128.185.136.39 0x80000082  469 0x5045
  2  128.185.125.40 128.185.129.40 0x80000049  457 0xA31
  2  128.185.126.25 128.185.129.25 0x80000002 1394 0x56B8
  2  128.185.127.24 128.185.126.24 0x8000007F 1031 0x592D
  2  128.185.129.25 128.185.129.25 0x8000005F 2295 0x8219
  2  128.185.129.40 128.185.129.40 0x80000001 1623 0x12C9
  6  224.0.2.6      128.185.142.9  0x8000003D  232 0x513F
  6  224.0.2.6      128.185.184.11 0x80000003  376 0x2250

             # advertisements:            14
             Checksum total:              0x4BBC2
```

Type

Separate LS types are numerically displayed: type 1 (router links advertisements), type 2 (network links advertisements), type 3 (network summaries), type 4 (AS boundary router summaries), and type 6 (group-membership-LSAs).

LS destination

Indicates what is being described by the advertisement.

LS originator

Advertising router.

Seqno, Age, Xsum — It is possible for several instances of an advertisement to be presenting the OSPF routing domain at any one time. However, only the most recent instance is kept in the OSPF link state database (and printed by this command). The LS sequence number (Seqno), LS age (Age) and LS checksum fields (Xsum) are compared to see which instance is most recent. The LS age field is expressed in seconds. Its maximum value is 3600.

At the end of the display, the total number of advertisements in the area database is printed, along with a checksum total over all of their contents. The checksum total is simply the 32-bit sum (carries discarded) of the individual advertisement's LS checksum fields. This information can be used to quickly determine whether two OSPF routers have synchronized databases.

Note: When comparing multicast-capable to non-multicast routers, the above database checksum (and also # advertisements) will not necessarily match, because non-multicast routers do not handle or store group-membership-LSAs.

Dump Routing Tables

Use the **dump routing tables** command to display all the routes that have been calculated by OSPF and are now present in the routing table. Its output is similar in format to the IP console's dump routing tables command.

Syntax: dump

Example: **dump**

```
Type   Dest net         Mask     Cost Age   Next hop(s)
SPE1   0.0.0.0          00000000  4    3    128.185.138.39
SPF*   128.185.138.0    FFFFFF00  1    1    Eth/0
Sbnt   128.185.0.0      FFFF0000  1    0    None
SPF    128.185.123.0    FFFFFF00  3    3    128.185.138.39
SPF    128.185.124.0    FFFFFF00  3    3    128.185.138.39
SPF    192.26.100.0     FFFFFF00  3    3    128.185.131.10
RIP    197.3.2.0        FFFFFF00  10   30   128.185.131.10
RIP    192.9.3.0        FFFFFF00  4    30   128.185.138.21
Del    128.185.195.0    FFFFFF00  16   270  None

Default gateway in use.

Type Cost Age  Next hop
SPE1 4    3    128.185.138.39

Routing table size: 768 nets (36864 bytes), 36 nets known
```

Type (route type) — Indicates how the route was derived.

Sbnt - indicates that the network is subnetted; such an entry is a placeholder only.

Dir - Indicates a directly connected network or subnet.

RIP - Indicates the route was learned through the RIP protocol.

Del - Indicates the route has been deleted.

Stat - Indicates a statically configured route.

BGP - Indicates routes learned through the BGP protocol.

BGPR - Indicates routes learned through the BGP protocol that are readvertised by OSPF and RIP.

Fltr - Indicates a routing filter.

	SPF - Indicates that the route is an OSPF intra-area route. SPIA - Indicates that it is an OSPF inter-area routes. SPE1, SPE2 - Indicates OSPF external routes (type 1 and 2 respectively). Rnge - Indicates a route type that is an active OSPF area address range and is not used in forwarding packets.
Dest net	IP destination network/subnet.
Mask	IP address mask.
Cost	Route Cost.
Age	For RIP and BGP routes, the time that has elapsed since the routing table entry was last refreshed.
Next Hop	IP address of the next router on the path toward the destination host. Also displayed is the interface type used by the sending router to forward the packet.

An asterisk (*) after the route type indicates the route has a static or directly connected backup. A percent sign (%) after the route type indicates that RIP updates will always be accepted for this network/subnet.

A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination. The first hops belonging to these routes can be displayed with the IP console's **route** command.

Interface Summary

Use the **interface summary** command to display statistics and parameters related to OSPF interfaces. If no arguments are given (see Example 1), a single line is printed summarizing each interface. If an interface's IP address is given (see Example 2), detailed statistics for that interface will be displayed.

Syntax: interface *interface-ip-address*

Example 1: `interface`

```
Ifc Address      Phys    assoc. Area     Type   State   #nbrs #adjs
9.67.217.66      TKR/0   2.2.2.2        Brdcst   64        0    0
```

Ifc Address	Interface IP address.
Phys	Displays the physical interface.
Assoc Area	Attached area ID.
Type	Can be either Brdcst (broadcast, e.g., an Ethernet interface), P-P (a point-to-point network, e.g., a synchronous serial line), P-2-MP (point-to-multipoint, e.g., a Frame-Relay network), Multi (non-broadcast, multi-access, e.g., an X.25 connection) or VLink (an OSPF virtual link).
State	Can be one of the following: 1 (down), 2 (looped back), 4 (waiting), 8 (point-to-point), 16 (DR other), 32 (backup DR) or 64 (designated router).
#nbrs	Number of neighbors. This is the number of routers whose hellos have been received, plus those that have been configured.

#adjs Number of adjacencies. This is the number of neighbors in state Exchange or greater. These are the neighbors with whom the router has synchronized or is in the process of synchronization.

Example 2: `interface 128.185.125.22`

```
Interface address:      128.185.125.22
Attached area:          0.0.0.1
Physical interface:     Eth/1
Interface mask:         255.255.255.0
Interface type:         Brdcst
State:                  32
Designated Router:      128.185.184.34
Backup DR:              128.185.184.11

DR Priority:       1  Hello interval:   10  Rxmt interval:     5
Dead interval:    40  TX delay:          1  Poll interval:     0
Max pkt size:   2044  TOS 0 cost:        1

# Neighbors:       0  # Adjacencies:     0  # Full adjs.:      0
# Mcast floods:    0  # Mcast acks:      0

MC forwarding:    on  DL unicast:      off  IGMP monitor:     on
# MC data in:      0  # MC data acc:     0  # MC data out:     0
IGMP polls snt:   75  IGMP polls rcv:    0  Unexp polls:      0
IGMP reports:      0
```

Interface Address Interface IP address.

Attached Area Attached area ID.

Physical interface Displays physical interface type and number.

Interface Mask Displays interface subnet mask.

Interface type Can be either Brdcst (broadcast, e.g., an Ethernet interface), PP (a point-to-point network, e.g., a synchronous serial line), P-2-MP (point-to-multipoint, e.g., a Frame-Relay network), Multi (non-broadcast, multi-access, e.g., an X.25 connection) and VLink (an OSPF virtual link).

State Can be one of the following: 1 (Down), 2 (Attempt), 4 (Init), 8 (2-Way), 16 (ExStart), 32 (Exchange), 64 (Loading) or 128 (Full).

Designated Router IP address of the designated router.

Backup DR IP address of the backup designated router.

DR Priority Displays priority assigned to designated router.

Hello interval Displays the current hello interval value.

Rxmt interval Displays the current retransmission interval value.

Dead interval Displays the current dead interval value.

TX delay Displays the current transmission delay value.

Poll interval Displays the current poll interval value.

Max pkt size Displays the maximum size for an OSPF packet sent out this interface.

TOS 0 cost Displays the interface's TOS 0 cost.

 # Neighbors Number of neighbors. This is the number of routers whose hellos have been received, plus those that have been configured.

 # Adjacencies Number of adjacencies. This is the number of neighbors in state Exchange or greater.

# Full adj	Number of full adjacencies. The number of full adjacencies is the number of neighbors whose state is Full (and therefore, with which the router has synchronized databases).
# Mcast Floods	Number of link state updates flooded out the interface (not counting retransmissions).
# Mcast acks	Number of link state acknowledgements flooded out the interface (not counting retransmissions).
MC forwarding	Displays whether multicast forwarding has been enabled for the interface.
DL unicast	Displays whether multicast datagrams are to be forwarded as data-link multicasts or as data-link unicasts.
IGMP monitor	Displays whether IGMP is enabled on the interface.
# MC data in	Displays the number of multicast datagrams that have been received on this interface and then successfully forwarded.
# MC data acc	Displays the number of multicast datagrams that have been successfully forwarded.
# MC data out	Displays the number of datagrams that have been forwarded out the interface (either as data-link multicasts or data-link unicasts).
IGMP polls sent	Displays the number of IGMP Host Membership Queries that have been sent out the interface.
IGMP polls rcv	Displays the number of IGMP Host Membership Queries that have been received on the interface.
Unexp polls	Displays the number of IGMP Host Membership Queries that have been received on the interface that were unexpected (i.e., received when the router itself was sending them).
IGMP reports	Displays the number of IGMP Host Membership Reports received on the interface.
Nbr node: type and ID	Displays the identity of the upstream node if the router were supposed to receive datagrams on this interface. Type here is an integer from 1 to 3, with 1 indicating router, 2 indicating transit net and 3 indicating stub net.

Join

Use the **join** command to establish the router as a member of a multicast group.

This command is similar to the join command in the OSPF configuration console with two differences:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (i.e., a restart/reload is not required).
- The command keeps track of the number of times a particular group is "joined."

When the router is the member of a multicast group, it responds to pings and SNMP queries sent to the group address.

Syntax: join *multicast-group-address*

Example: `join 224.185.0.0`

Leave

Use the **leave** command to remove a router's membership in a multicast group. This will keep the router from responding to pings and SNMP queries sent to the group address.

This command is similar to the leave command in the OSPF configuration console with two differences:

- The effect on group membership is immediate when the commands are given from the OSPF monitor (i.e., a restart/reload is not required).
- The command will not delete group membership until the “leaves” executed equals the number of “joins” previously executed.

Syntax: leave *multicast-group-address*

Example: **`leave 224.185.0.0`**

Mcache

Use the **mcache** command to display a list of currently active multicast cache entries. Multicast cache entries are built on demand, whenever the first matching multicast datagram is received. There is a separate cache entry (and therefore a separate route) for each datagram source network and destination group combination.

Cache entries are cleared on topology changes (e.g., a point-to-point line in the MOSPF system going up or down), and on group membership changes.

Syntax: mcache

Example 1: mcache

```
          0: TKR/0          1: SDLC/0          2: FR/0
          3: Internal

     Source          Destination     Count   Upst    Downstream
     133.1.169.2     225.0.1.10      8       Local   2 (4),3
     133.1.169.2     225.0.1.20      8       Local   2 (4),3
     3.3.3.3         225.0.1.10      8       2       3
```

Source	Source network/subnet of matching datagrams.
Destination	Destination group of matching datagrams.
Count	Displays the number of received datagrams that have matched the cache entry.
Upst	Displays the neighboring network/router from which the datagram must be received in order to be forwarded. When this reads as “none,” the datagram will never be forwarded.
Downstream	Displays the total number of downstream interfaces/neighbors to which the datagram will be forwarded. When this is 0, the datagram will not be forwarded.

Mgroups

Use the **mgroups** command to display the group membership of the router's attached interfaces. Only the group membership for those interfaces on which the router is either designated router or backup designated router are displayed.

Syntax: mgroups

Example: **mgroups**

```
                  Local Group Database
 Group              Interface                Lifetime (secs)

224.0.1.1         128.185.184.11 (Eth/1)       176
224.0.1.2         128.185.184.11 (Eth/1)       170
224.1.1.1         Internal                     1
```

Group — Displays the group address as it has been reported (via IGMP) on a particular interface.

Interface — Displays the interface address to which the group address has been reported (via IGMP).

The router's internal group membership is indicated by a value of “internal.” For these entries, the lifetime field (see below) indicates the number of applications that have requested membership in the particular group.

Lifetime — Displays the number of seconds that the entry will persist if Membership Reports cease to be heard on the interface for the given group.

Mstats

Use the **mstats** command to display various multicast routing statistics. The command indicates whether multicast routing is enabled and whether the router is an inter-area and/or inter-AS multicast forwarder. inter-area multicast forwarder.

Syntax: mstats

Example: **mstats**

```
                    MOSPF forwarding:       Enabled
                    Inter-area forwarding:  Enabled
                    DVMRP forwarding:       Disabled

Datagrams received:            2496  Datagrams (ext source):    0
Datagrams fwd (multicast):        0  Datagrams fwd (unicast):   0
Locally delivered:                0  No matching rcv interface: 0
Unreachable source:               3  Unallocated cache entries: 0
Off multicast tree:               0  Unexpected DL multicast:   0
Buffer alloc failure:             0  TTL scoping:               0

# DVMRP routing entries:          0  # DVMRP entries freed:     0
# fwd cache alloc:                1  # fwd cache freed:         0
# fwd cache GC:                   0  # local group DB alloc:    0
# local group DB free:            1
```

MOSPF forwarding — Displays whether the router will forward IP multicast datagrams.

Inter-area forwarding — Displays whether the router will forward IP multicast datagrams between areas.

DVMRP forwarding	Displays whether the router is configured to use DVMRP for multicast routing.
Datagrams received	Displays the number of multicast datagrams received by the router (datagrams whose destination group lies in the range 224.0.0.1 - 224.0.0.255 are not included in this total).
Datagrams (ext source)	Displays the number of datagrams that have been received whose source is outside the AS.
Datagrams fwd (multicast)	Displays the number of datagrams that have been forwarded as data-link multicasts (this includes packet replications, when necessary, so this count could very well be greater than the number received).
Datagrams fwd (unicast)	Displays the number of datagrams that have been forwarded as data-link unicasts.
Locally delivered	Displays the number of datagrams that have been forwarded to internal applications.
No matching rcv interface	Displays the count of those datagrams that were received by a non-inter-AS multicast forwarder on a non-MOSPF interface.
Unreachable source	Displays a count of those datagrams whose source address was unreachable.
Unallocated cache entries	Displays a count of those datagrams whose cache entries could not be created due to resource shortages.
Off multicast tree	Displays a count of those datagrams that were not forwarded either because there was no upstream neighbor or no downstream interfaces/neighbors in the matching cache entry.
Unexpected DL multicast	Displays a count of those datagrams that were received as data-link multicasts on those interfaces that have been configured for data-link unicast.
Buffer alloc failure	Displays a count of those datagrams that could not be replicated because of buffer shortages.
TTL scoping	Indicates those datagrams that were not forwarded because their TTL indicated that they could never reach a group member.
DVMRP routing entries	Displays the number of DVMRP routing entries
DVMRP entries freed	Indicates the number of DVMRP entries that have been freed.The size will be the number of routing entries minus the number of entries freed.
# fwd cache alloc	Indicates the number of cache entries allocated. The current forwarding cache size is the number of entries allocated (“# fwd cache alloc”) minus the number of cache entries freed (“# fwd cache freed”).
# fwd cache freed	Indicates the number of cache entries freed. The current forwarding cache size is the number of entries allocated (“# fwd cache alloc”) minus the number of cache entries freed (“# fwd cache freed”).
# fwd cache GC	Indicates the number of cache entries were cleared because they were not recently used and the cache overflowed.

# local group DB alloc	Indicates the number of local group database entires allocated. The number allocated ("# local group DB alloc") minus the number freed ("# local group DB free") equals the current size of the local group database.
# local group DB free	Indicates the number of local group database entires freed. The number allocated ("# local group DB alloc") minus the number freed ("# local group DB free") equals the current size of the local group database.

The number of cache hits can be calculated as the number of datagrams received ("Datagrams received") minus the total of datagrams discarded due to "No matching rcv interface," "Unreachable source" and "Unallocated cache entries," and minus "# local group DB alloc." The number of cache misses is simply "# local group DB alloc."

Neighbor Summary

Use the **neighbor summary** command to display statistics and parameters related to OSPF neighbors. If no arguments are given (see Example 1), a single line is printed summarizing each neighbor. If a neighbor's IP address is given (see Example 2), detailed statistics for that neighbor will be displayed.

Syntax: neighbor *neighbor-ip-address*

Example 1: `neighbor`

```
Neighbor addr   Neighbor ID     State  LSrxl  DBsum  LSreq  Ifc
128.185.125.39  128.185.136.39  128      0      0      0   PPP/1
128.185.125.41  128.185.128.41    8      0      0      0   PPP/1
128.185.125.38  128.185.125.38    8      0      0      0   PPP/1
128.185.125.25  128.185.129.25    8      0      0      0   PPP/1
128.185.125.40  128.185.129.40  128      0      0      0   PPP/1
128.185.125.24  128.185.126.24    8      0      0      0   PPP/1
```

Neighbor addr	Displays the neighbor address.
Neighbor ID	Displays the neighbor's OSPF router ID.
Neighbor State	Can be one of the following: 1 (Down), 2 (Attempt), 4 (Init), 8 (2-Way), 16 (ExStart), 32 (Exchange), 64 (Loading) or 128 (Full).
LSrxl	Displays the size of the current link state retransmission list for this neighbor.
DBsum	Displays the size of the database summary list waiting to be sent to the neighbor.
LSreq	Displays the number of more recent advertisements that are being requested from the neighbor.
Ifc	Displays the interface shared by the router and the neighbor.

Example 2: `neighbor 128.185.138.39`

The meaning of most of the displayed fields is given in section 10 of the OSPF specification (RFC 1131).

```
Neighbor IP address:    128.185.184.34
OSPF Router ID:         128.185.207.34
Neighbor State:         128
Physical interface:     Eth/1
DR choice:              128.185.184.34
```

```
            Backup choice:          128.185.184.11
            DR Priority:            1
            Nbr options:            E,MC


            DB summ qlen:      0  LS rxmt qlen:      0  LS req qlen:
0
            Last hello:        7


            # LS rxmits:     108  # Direct acks:    13  # Dup LS rcvd:
572
            # Old LS rcvd:     2  # Dup acks rcv:  111  # Nbr losses:
29
            # Adj. resets:    30
```

Neighbor IP addr	Neighbor IP address.
OSPF router ID	Neighbor's OSPF router ID.
Neighbor State	Can be one of the following: 1 (Down), 2 (Attempt), 4 (Init), 8 (2-Way), 16 (ExStart), 32 (Exchange), 64 (Loading) or 128 (Full).
Physical interface	Displays physical interface type and number of the router and neighbor's common network.
DR choice, backup choice, DR priority	Indicate the values seen in the last hello received from the neighbor.
Nbr options	Indicates the optional OSPF capabilities supported by the neighbor. These capabilities are denoted by E (processes type 5 externals; when this is not set the area to which the common network belongs has been configured as a stub), T (can route based on TOS) and MC (can forward IP multicast datagrams). This field is valid only for those neighbors in state Exchng or greater.
DBsumm qlen	Indicates the number of advertisements waiting to be summarized in Database Description packets. It should be zero except when the neighbor is in state Exchange.
LS rxmt qlen	Indicates the number of advertisements that have been flooded to the neighbor, but not yet acknowledged.
LS req qlen	Indicates the number of advertisements that are being requested from the neighbor in state Loading.
Last hello	Indicates the number of seconds since a hello has been received from the neighbor.
# LS rxmits	Indicates the number of retransmissions that have occurred during flooding.
# direct acks	Indicates responses to duplicate link state advertisements.
# Dup LS rcvd	Indicates the number of duplicate retransmissions that have occurred during flooding.
# Old LS rcvd	Indicates the number of old advertisements received during flooding.
# Dup acks rcvd	Indicates the number of duplicate acknowledgements received.
# Nbr losses	Indicates the number of times the neighbor has transitioned to Down state.
# Adj. resets	Counts entries to state ExStart.

Ping

See “Ping” on page 15-6 for an explanation of the **Ping** command.

Traceroute

See “Traceroute” on page 15-8 for an explanation of the **Traceroute** command.

Routers

Use the **routers** command to display all router routes that have been calculated by OSPF and are now present in the routing table. With the **dump routing tables** command, the Net field indicates that the destination is a network. The routers command covers all other destinations.

Syntax: routers

Example: **`routers`**

```
DType RType Destination       AREA          Cost   Next hop(s)

 ASBR  SPF   128.185.142.9    0.0.0.1         1   128.185.142.9
 Fadd  SPF   128.185.142.98   0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.7    0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.48   0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.111  0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.38   0.0.0.1         1    0.0.0.0
 Fadd  SPF   128.185.142.11   0.0.0.1         1    0.0.0.0
   BR  SPF   128.185.142.9    0.0.0.2         1   128.185.142.9
   BR  SPF   128.185.142.9    0.0.0.2         2   128.185.184.114
 Fadd  SPF   128.185.142.47   0.0.0.2         1    0.0.0.0
```

DType — Indicates destination type:

Net	indicates that the destination is a network
ASBR	indicates that the destination is an AS boundary router
ABR	indicates that the destination is an area border router
Fadd	indicates a forwarding address (for external routes)

RType — Indicates route type and how the route was derived:

SPF	indicates that the route is an intra-area route (comes from the Dijkstra calculation)
SPIA	indicates that it is an inter-area route (comes from considering summary link advertisements).

Destination — Destination router's OSPF ID. For Type D entries, one of the router's IP addresses is displayed (which corresponds to a router in another AS).

Area — Displays the AS area to which it belongs.

Cost — Displays the route cost.

Next hop — Address of the next router on the path toward the destination host. A number in parentheses at the end of the column indicates the number of equal-cost routes to the destination.

Size

Use the **size** command to display the number of LSAs currently in the link state database, categorized by type.

Syntax: size

Example: **size**

```
# Router-LSAs:            6
# Network-LSAs:           2
# Summary-LSAs:           45
# Summary Router-LSAs:    6
# AS External-LSAs:       2
# Group-membership-LSAs:  11

# Intra-area routes:      11
# Inter-area routes:      15
# Type 1 external routes: 0
# Type 2 external routes: 2
```

Statistics

Use the **statistics** command to display statistics generated by the OSPF routing protocol. The statistics indicate how well the implementation is performing, including its memory and network utilization. Many of the fields displayed are confirmation of the OSPF configuration.

Syntax: statistics

Example: **statistics**

```
S/W version:            2.1
OSPF Router ID:         128.185.184.11
External comparison:    Type 2
AS boundary capability: Yes
Import external routes: BGP RIP STA DIR SUB
Orig. default route:    No (0,0.0.0.0)
Default route cost:     (1, Type 2)
Default forward. addr:  0.0.0.0

Attached areas:               1  Estimated # external routes: 10
Estimated # OSPF routers:    30  Estimated heap usage:      2368
OSPF packets rcvd:            0  OSPF packets rcvd w/ errs:   0
Transit nodes allocated:      6  Transit nodes freed:         0
LS adv. allocated:           24  LS adv. freed:               2
Queue headers alloc:         32  Queue headers avail:         3

 # Dijkstra runs:             1  Incremental summ. updates:   0
Incremental VL updates:       0  Buffer alloc failures:       0
Multicast pkts sent:        625  Unicast pkts sent:           0
LS adv. aged out:             0  LS adv. flushed:             0

External LSA database:
Current state:           Normal
Number of LSAs:               9
Number of overflows           0
```

S/W version	Displays the current OSPF software revision level.
OSPF Router ID	Displays the router's OSPF ID.
External comparison	Displays the external route type used by the router when importing external routes.
AS boundary capability	Displays whether external routes will be imported.
Import external routes	Displays which external routes will be imported.

Orig default route	Displays whether the router will advertise an OSPF default route. If the value is "Yes" and a nonzero number is displayed in parentheses, then a default route will be advertised only when a route to the network exists.
Default route cost	Displays the cost and type of the default route (if advertised).
Default forward addr	Displays the forwarding address specified in the default route (if advertised).
Attached areas	Indicates the number of areas that the router has active interfaces to.
Estimated heap usage	Rough indication of the size of the OSPF link state database (in bytes).
Transit nodes	Allocated to store router links and network links advertisements.
LS adv.	Allocated to store summary link and AS external link advertisements.
Queue headers	Form lists of link state advertisements. These lists are used in the flooding and database exchange processes; if the number of queue headers allocated is not equal to the number freed, database synchronization with some neighbor is in progress.
# Dijkstra runs	Indicates how many times the OSPF routing table has been calculated from scratch.
Incremental summ updates, incremental VL updates	Indicate that new summary link advertisements have caused the routing table to be partially rebuilt.
Buffer alloc failures.	Indicate buffer allocation failures. The OSPF system will recover from temporary lack of packet buffers.
Multicast pkts sent	Covers OSPF hello packets and packets sent during the flooding procedure.
Unicast pkts sent	Covers OSPF packet retransmissions and the Database Exchange procedure.
LS adv. aged out	Counts the number of advertisements that have hit 60 minutes. Link state advertisements are aged out after 60 minutes. Usually they will be refreshed before this time.
LS adv. flushed	Indicates number of advertisements removed (and not replaced) from the link state database.
Incremental ext. updates.	Displays number of changes to external destinations that are incrementally installed in the routing table.
External LSA database:	Provides information about the LSA database:
	Current state — Whether the database of current AS external LSAs is in normal or overload state.
	Number of LSA — The number of external LSAs currently in the database
	Number of overflows — Number of times the external AS LSA database has entered overload state.

Weight

Use the **weight** command to change the cost of one of the routers OSPF interfaces. This new cost is immediately flooded throughout the OSPF routing domain, causing routes to be updated accordingly.

The cost of the interface will revert to its configured cost whenever the router is restarted or reloaded. To make the cost change permanent, you must reconfigure the appropriate OSPF interface after invoking the weight command. This command will cause a new router links advertisement to be originated, unless the cost of the interface does not change.

Syntax: weight *ip-interface-address new-cost*

Example: **`weight 128.185.124.22 2`**

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Chapter 18. Configuring SNMP

This chapter describes the SNMP configuration commands and includes the following sections:

- "Accessing the SNMP Configuration Environment"
- "SNMP Configuration Commands"

Accessing the SNMP Configuration Environment

To access the SNMP configuration environment, enter the following command at the `Config>` prompt:

```
Config> protocol snmp
SNMP user configuration
SNMP Config>
```

SNMP Configuration Commands

This section summarizes and then explains all the SNMP configuration commands.

Table 18-1 on page 18-2 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 8210.

Enter the SNMP configuration commands at the `SNMP Config>` prompt.

Table 18-1. SNMP Configuration Commands Summary

Command	Function
? (Help)	Lists all the SNMP configuration commands or lists the options associated with specific commands.
Add	Adds a community to the list of SNMP communities, an IP address with mask to a community, or a subtree to a MIB view.
Delete	Removes a community from the list of SNMP communities, an IP address with mask from a community, or a subtree from a MIB view.
Enable/Disable	Enables/disables SNMP protocol and traps associated with named communities.
List	Displays the current communities with their associated access modes, enabled traps, IP addresses, and views. Also displays all views and their associated MIB subtrees.
Set	Sets a community's access mode or view. A community's access mode is one of the following: Read and trap generation Read, write and trap generation Trap generation only Also allows setting of trap UDP port.
Exit	Exits the SNMP configuration process and returns to the CONFIG environment.

? (Help)

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

Syntax: ?

Example: ?

```
ADD
DELETE
SET
ENABLE
DISABLE
LIST
EXIT
```

Add

Use the **add** command to add a community name to the list of SNMP communities, add an address to a community, or assign a portion of the MIB (subtree) to a view.

Syntax: add community
address
sub_tree

community

Use the **add community** command to create a community. It will be created with a default access of read_trap, a view of all, all traps disabled and all IP addresses allowed.

Note: The **add community** command no longer allows you to select access type or trap control. Use the set community access command to assign access types to existing SNMP communities and use the **enable trap** or the **disable trap** command for trap control.

The *community name*. parameter provides the community name used by the SNMP client. This community name is used when accessing the management information base (MIB) in the device from the host specified by the Community IP address parameter.

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

Example: **`add community <community_name>`**

```
Community Name  []?
```

Community Name	Specifies the name of community (32 visual characters maximum). Characters such as spaces, tabs, or <esc> key sequences are not accepted.

address

Use the **add address** command to add to the community definition an address of a network management station in the network that should be allowed to communicate with this box. You must supply the name of the community and the network address (in standard a.b.c.d notation). You also may supply a net mask to restrict access to either an individual host (mask = 255.255.255.255) or to a network of hosts. More than one address can be added to a community; enter the command each time you want to add another address.

If you do not specify an address for a community, requests are handled from any host. Addresses specify hosts that receive the traps. If no address is specified, no trap is generated.

Also, a trap is sent to a specified host only if the associated net mask is defined as 255.255.255.255.

1. The *community name* has:

   Valid Values: A string of 1 to 32 alphanumeric characters.

   Characters such as spaces, tabs, or <ESC> key sequences are not supported.

   Default Value: none

2. The *IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

3. You also may supply a *net mask* to restrict access to either an individual host (mask = 255.255.255.255) or to a network of hosts.

Valid Values: 0.0.0.0 - 255.255.255.255

Default Value: none

Example: **`add address <community_name> <ipAddress> <ipMask>`**

```
Community Name []?
New Address [0.0.0.0]?
```

sub_tree

Use the **add sub_tree** command to add a portion of the MIB to a view or to create a new view. The default is the entire MIB. The **add sub_tree** command is used to manage MIB views. More than one subtree can be added to a view defined by <view_text_name>. To create a new MIB view, issue the **add sub_tree** command with the new view name.

Note: You must assign a view to one or more communities using the **set community view** command to have it take effect. The subtree definitions are inclusive; that is, the subtree OID specified and any OID that is lexicographically greater than the specified OID is considered part of the MIB view.

Valid Values:

- All - Assigns all supported MIB views to the named community.
- View - Assigns a specified MIB view to the named community.

Default Value: All

The *MIB OID name* is the parameter that specifies the MIB Object ID for the sub_tree. This must be entered as a numeric value, not a symbolic value.

This parameter contains a MIB subtree name included in the view defined with the View name parameter. All children of a specified MIB subtree are also included in the view.

For example, to provide a view that would give access to the system group in MIB-II, specify **1.3.6.1.2.1.1**.

Valid Values:

An object identifier in the form of <element1>.<element2>.<element3>..., where:

- You need a minimum of 3 elements.
- You can define a maximum of 49 elements.
- element1 is 0, 1, or 2.
- element2 is an integer between 1 and 40.
- element3 and subsequent elements are integers between 1 and the size of an unsigned byte integer.

Default Value: None

Example: **`add sub_tree`**

```
View Name  []?
MIB OID name []?
```

View Name	Specify the name of the view (32 visual characters maximum). Characters such as spaces, tabs, or <Esc> key sequences are not accepted.
MIB OID	Specifies the MIB Object ID for the sub_tree. This must be entered as a numeric value in dotted notation, *not* a symbolic value.

Delete

Use the **delete** command to delete:

- a specific address.
- a community and all of its addresses.
- a subtree from a view.

Syntax: delete community
address
sub_tree

community

Removes a community and its IP addresses. You must supply the community name.

The *community name*.

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

This parameter provides a community name used by the SNMP client. This community name is used when accessing the management information base (MIB) in the device from the host specified by the Community IP address parameter.

Example: `delete community <community_name>`

address

Removes an address from a community. You must supply the name.

1. The *community name* has:

   Valid Values: A string of 1 to 31 alphanumeric characters.

   Characters such as spaces, tabs, or <ESC> key sequences are not supported.

   Default Value: public

   This parameter provides a community name used by the SNMP client. This community name is used when accessing the management information base (MIB) in the device from the host specified by the Community IP address parameter.

2. The *IP address* has:

   Valid Values: Any valid IP address.

   Default Value: none

3. You also may supply a *net mask* to restrict access to either an individual host (mask = 255.255.255.255) or to a network of hosts.

   Valid Values: 0.0.0.0 - 255.255.255.255

   Default Value: none

Example: `delete address <comm_name> <ipAddress> <ipMask>`

sub_tree

Removes a MIB or a portion of the MIB from a view. You must supply the name of the subtree. If all subtrees are deleted, the MIB view is also deleted and all references to it from any associated SNMP communities are removed.

1. The *view name* to be removed is the parameter that allows you to select the view used by the community defined in the Community name parameter. This view determines which MIB objects this community may access. If no view is specified, the community may access all objects known to the router's SNMP agent.

   This parameter should be answered if you decide to restrict a community from accessing the entire MIB managed by the router's SNMP agent.

   You must configure the View name parameter and the MIB Subtree parameter before you can configure this parameter.

   Valid Values:

   - All - Assigns all supported MIB views to the named community.
   - View - Assigns a specified MIB view to the named community.

   Default Value: All

2. The *MIB OID name* is the parameter that specifies the MIB Object ID for the sub_tree. This must be entered as a numeric value, not a symbolic value.

   This parameter contains a MIB subtree name included in the view defined with the View name parameter. All children of a specified MIB subtree are also included in the view.

   For example, to provide a view that would give access to the system group in MIB-II, specify **1.3.6.1.2.1.1**.

   Valid Values:

   An object identifier in the form of <element1>.<element2>.<element3>..., where:

   - You need a minimum of 3 elements.
   - You can define a maximum of 49 elements.
   - element1 is 0, 1, or 2.
   - element2 is an integer between 1 and 40.
   - element3 and subsequent elements are integers between 1 and the size of an unsigned byte integer.

   Default Value: None

Example: `delete sub_tree <view_text_name> <oid>`

Disable

Use the **disable** command to disable the SNMP protocol or specified traps on the router.

Syntax: disable snmp
trap

snmp
Disables SNMP

The *community name* has:

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

Example: `disable snmp`

trap
Disables specified traps or all traps. You must specify the trap type from the following options.

Example: `disable trap <trap_type> <community_name>`

Trap Type	Description
all	Disables all traps in a specified community. Specify the community name as part of the command line.
cold_start	Disables cold start traps in a specified community. A cold start trap (0) means that the transmitting router is reinitializing and that the agent's configuration or the protocol entity implementation may be altered. Specify the community name as part of the command line.
link_down	Disables link_down traps in a specified community. A link_down trap (2) recognizes a failure in one of the communication links represented in the agent's configuration. The link_down trap-PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable-bindings.
link_up	Disables link_up traps in a specified community. A link_up trap recognizes that a previously inactive link in the network has come up. The link_up trap-PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable-bindings.
auth_fail	Disables authentication failure traps for a specified community. Authentication failure traps indicate that the sender of the SNMP request does not have the proper permission to talk to this box's SNMP agent.
enterprise	Disables enterprise specific traps in a specified community. Enterprise specific traps indicate that some enterprise specific event has occurred. The specific-trap field identifies the particular trap that occurred. For example, when configured to do so, ELS event messages are sent in enterprise-specific traps.

Enable

Use the **enable** command to enable the SNMP protocol or specified traps on the router.

Syntax: enable snmp
trap

snmp

Enables SNMP

Example: `enable snmp`

trap

Enables specified traps or all traps. You must specify the trap type from the options shown below.

The *community name* has:

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

Example: `enable trap <trap_type> <community_name>`

Trap Type	Description
all	Enables all traps in a specified community. Specify the community name as part of the command line.
cold_start	Enables cold start traps in a specified community. A cold start trap (0) means that the transmitting router is reinitializing and that the agent's configuration or the protocol entity implementation may be altered. Specify the community name as part of the command line.
link_down	Enables link_down traps in a specified community. A link_down trap (2) recognizes a failure in one of the communication links represented in the agent's configuration. The link_down trap-PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable-bindings.
link_up	Enables link_up traps in a specified community. A link_up trap recognizes that a previously inactive link in the network has come up. The link_up trap-PDU contains the name and value of the ifIndex instance for the affected link as the first element of its variable-bindings.
auth_fail	Enables authentication failure traps for a specified community. Authentication failure traps indicate that the sender of the SNMP request does not have the proper permission to talk to this box's SNMP agent.
enterprise	Enables enterprise specific traps in a specified community. Enterprise specific traps indicate that some enterprise specific event has occurred. The specific-trap field identifies the particular trap that occurred. For example, when configured to do so, ELS event messages are sent in enterprise-specific traps.

List

Use the **list** command to display the current configuration of SNMP communities, access modes, traps, network addresses, and views.

Syntax: list all
community
views

list all

Displays the current configuration of SNMP communities for Access, Traps, Address, and View. See the description for the list community command on the next page for details on the options.

Example: `list all`

```
SNMP is enabled.
Trap UDP port: 162

Community Name   Access
public           Read, Write, Trap
oxnard           Read, Trap

Community Name   Enabled Traps
public           Link Down, Cold Restart
oxnard           NONE

Community Name   IP Address  IP Mask
public           All         N/A
oxnard           1.1.1.2     255.255.255.255

Community Name   View
public           All
oxnard           mib2
```

list community *option*

Displays the current attributes of an SNMP community. Options are access, traps, address, view.

Option	Description
Access	Displays the access modes for the community.
Address	Displays the network address for the community.
Traps	Displays the types of traps generated for the community.
View	Displays the MIB view for the community.

list community access

Example: `list community access`

```
Community Name   Access
public           Read, Write, Trap
oxnard           Read, Trap
```

list community traps

Example: `list community traps`

```
Community Name   Enabled Traps
public           Link Down, Cold Restart
oxnard           NONE
```

list community address

Example: `list community address`

```
Community Name   IP Address   IP Mask
public           All          N/A
oxnard           1.1.1.2      255.255.255.255
```

list community view

Example: `list community view`

```
Community Name   View
public           All
oxnard           mib2
```

list views

Displays the current views for a specified SNMP community.

Example: `list views`

```
View Name    Sub-Tree
mib2         1.3.6.1.2.1
```

Set

Use the **set** command to assign a MIB view to a community, to set the SNMP UDP trap port number, or set the access mode of the community.

Syntax: set community access
community view
trap_port

community access

Use the **set community access** command to assign one of three access types to a community. You must supply the name of the community and the access type.

The *community name* has:

Valid Values: A string of 1 to 31 alphanumeric characters.

Characters such as spaces, tabs, or <ESC> key sequences are not supported.

Default Value: public

Example: `set community access <options> <comm_name>`

Options	Description
read_trap	Sets read access and trap generation to the named community.
write_read_trap	Sets write and read and trap generation access to the community specified.
trap_only	Indicates the community is used only when sending an SNMP trap.

community view

Use the **set community view** command to assign a MIB view to a community.

Example: `set community view <comm_name> <options>`

Options	Description
all	Allows access to all MIB objects for the named community. All is the default.
view_text_name	Assigns a specified MIB view to the named community.

trap_port

Use the **set trap_port** command to specify a UDP port number, other than the default standard port 162, to send traps to. The default is the standard port.

Example: `set trap_port <udpport#>`

UDP Port Number	Specifies a User Datagram Protocol port other than the standard UDP port (default # 162).

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 19. Monitoring SNMP

This chapter describes the SNMP console commands and includes the following sections:

- "Accessing the SNMP Console Environment"
- "SNMP Console Commands"

Accessing the SNMP Console Environment

To access the SNMP console environment, enter the following command at the **+** (GWCON) prompt:

```
+ protocol snmp
SNMP>
```

SNMP Console Commands

This section summarizes and then explains all of the SNMP console commands.

Table 19-1 on page 19-2 lists the SNMP console commands. The SNMP console commands allow you to view the parameters of the SNMP configuration and display some statistics relating to the SNMP agent.

Temporary changes to the runtime SNMP parameters can be made through the console. They will immediately affect the operation of the SNMP agent. If you want to make the temporary changes permanent, then use the SAVE command. If the original SNMP configuration needs to be restored, use the REVERT command. This feature allows you to temporarily alter the behavior of the SNMP agent, without permanently changing the configuration. For the temporary changes to take affect, you must EXIT the SNMP console process.

Enter the SNMP console commands at the SNMP> prompt.

Table 19-1. SNMP Console Command Summary

Command	Function
? (Help)	Lists all the SNMP console commands or lists the options associated with specific commands.
Add	Adds a community to the list of SNMP communities, an IP address with mask to a community, or a subtree to a MIB view.
Delete	Removes a community from the list of SNMP communities, an IP address with mask from a community, or a subtree from a MIB view.
Enable/Disable	Enables/disables SNMP protocol and traps associated with named communities. These actions are only allowed in the SNMP Configuration environment.
List	Displays the current configuration of SNMP communities, views, access modes, traps, and network addresses.
Revert	Erases the specified changes and restores the settings to the values in the permanent SNMP configuration.
Save	Takes the specified changes and saves then permanently in the SNMP configuration.
Set	Sets a community's access mode or view. A community's access mode is one of the following: • Read and trap generation • Read, write and trap generation • Trap generation only Also allows setting of trap UDP port.
Statistics	Displays statistics about the SNMP agent.
Exit	Exits the SNMP console process and returns to the GWCON environment.

? (Help)

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

Syntax: *?*

Example: **?**

```
list
statistics
exit
```

Add

Use the **add** command to add a community name to the list of SNMP communities, add an address to a community, or assign a portion of the MIB (subtree) to a view.

For information on using the **add** command, see “add” on “Add” on page 18-2.

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 18-5.

Disable

Use the **disable** command to disable the SNMP protocol or specified traps on the router. This command is available only in the SNMP Configuration environment.

For information on using the **disable** command, see “Disable” on page 18-7.

Enable

Use the **enable** command to enable the SNMP protocol or specified traps on the router. This command is available only in the SNMP Configuration environment.

For information on using the **enable** command, see “Enable” on page 18-8.

List

Use the **list** command to display the current configuration of SNMP communities, views, access modes, traps, and network addresses.

Syntax: list all
community
views

list all

Displays the current configuration of SNMP communities for Access, Traps, Address, and View. See the description for the list community command on the next page for details on the options.

Example: `list all`

```
SNMP is enabled.
Trap UDP port: 162

Community Name   Access
public           Read, Write, Trap
oxnard           Read, Trap


Community Name   Enabled Traps
public           Link Down, Cold Restart
oxnard           None


Community Name   IP Address  IP Mask
public           All         N/A
oxnard           1.1.1.2     255.255.255.255


Community Name   View
public           All
oxnard           mib2


View Name        Sub-Tree
mib2             1.3.6.1.2.1
```

list community option

Displays the current attributes of a specified SNMP community. Options are access, traps, address, view.

Example: **`list community`** option

Option	Description
Access	Displays the access modes for the community.
Address	Displays the network address for the community.
Traps	Displays the types of traps generated for the community.
View	Displays the MIB view for the community.

list community access

Example: **`list community access`**

```
Community Name   Access
public           Read, Write, Trap
oxnard           Read, Trap
```

list community traps

Example: **`list community traps`**

```
Community Name   Enabled Traps
public           Link Down, Cold Restart
oxnard           None
```

list community address

Example: `list community address`

```
Community Name   IP Address    IP Mask
public           All           N/A
oxnard           1.1.1.2       255.255.255.255
```

list community view

Example: `list community view`

```
Community Name   View
public           All
oxnard           mib2
```

list views

Displays the current views for a specified SNMP community.

Example: `list views`

```
View Name    Sub-Tree
mib2         1.3.6.1.2.1
```

Revert

Use the **revert** command to erase the specified changes and restore the settings to the values in the permanent SNMP configuration.

Save

Use the **save** command to save the specified changes permanently.

Set

For information on using the **set** command, see “Set” on page 18-11.

Statistics

Use the **statistics** command to display statistics about the SNMP agent.

Syntax: statistics

Example: `statistics`

```
SNMP memory in use = 9416
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 20. Using and Configuring IPX

This chapter describes how to use the IPX protocol on your IBM 8210 and how to configure the IPX protocol using the IPX configuration commands. 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 standalone NetWare bridge.
- Support for the Novell NetBIOS emulator.

IPX Addressing

Every IPX interface must have a unique address that corresponds to the network number assigned to the attached IPX network. A simple solution is to use multipart addresses like the city-street-house address on a piece of mail. For example, IPX refers to network numbers (city), host numbers (street), and socket numbers (house).

Network Numbers

An IPX network number specifies the location of a particular network in an internetwork. These addresses allow communication between two entities on different networks.

Host Numbers

Each IPX interface needs a 6-byte host (node) number.

Token-Ring and Ethernet interfaces use their hardware MAC address as their host number, and you cannot change them.

Because serial lines have no hardware MAC addresses, you must specify a unique host number.

| ATM interfaces use their End System Identifier (ESI) as their host number.
| Their burned-in ESI will be used if one has not been configured.

Configuring IPX

This section describes how to initially configure IPX. The following sections describe optional parameters you can set.

1. Display the IPX configuration prompt as shown here:

```
* talk 6
  Config> protocol ipx
  IPX protocol user configuration
  IPX config>
```

2. Enable the IPX protocol on the router.

```
IPX Config>enable ipx
```

3. Enable IPX and assign an IPX network number on each interface on which you want to run IPX. Every interface must have a unique network number that corresponds to the network number that is assigned to the attached IPX network.

   Repeat this step for each interface.

```
IPX Config>enable interface
Which interface [0]? 0
Configure an IPX network number for this interface.
Network number in hex [1]? 180
```

   Note: You cannot enter 0 or 0xFFFFFFFF as network numbers. Novell has reserved these numbers.

4. Optionally change the frame type for Ethernet or Token Ring. You do not have to set the frame type for interfaces other than Ethernet or Token Ring. See “Frame” on page 20-21 for a description of available frame types.

   The default encapsulation formats are:

   - Ethernet - Ethernet_8023
   - Token Ring - Token-ring MSB

   Use the **frame** command as shown here:

```
IPX config> frame ethernet_8023
Which interface [0]? 1
```

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 interfaces. It also determines how long a RIP entry remains before being aged-out.

Valid entries remain in the routing tables for a period of three multiples of the RIP update interval, and the router broadcasts its RIP tables once every update interval.

For example, the default interval is 1 minute, which allows a valid entry to remain in the table for 3 minutes. After this time, if an entry is not refreshed by a RIP update, the route is marked with a hop count of infinity (16) and then it is deleted. Every 60 seconds the router broadcasts its RIP tables to corresponding interfaces.

You can configure the RIP interval from 1 to 1440 minutes (24 hours). Increasing the RIP interval reduces traffic on WAN lines and dial circuits. It also prevents dial-on-demand circuits from dialing out as often.

Note: While complete RIP advertisements are controlled by the interval, the router still propagates network topology changes as quickly as it learns them.

The RIP interval is not configurable on the Novell file server.

```
IPX config>set rip-update-interval
Which interface [0]? 2
RIP timer value(minutes) [1]? 2
```

Specifying the Size of IPX SAP Services Table

The IPX Service Advertising Protocol (SAP) services table is a distributed database used to find NetWare Services, such as file servers. Services are uniquely identified by a 2-byte numeric type and a 47-character name. Each service provider advertises its services, specifying service type, name, and address. The router accumulates this information in a table and sends it to other routers. The default table size is 32.

You can configure the table size from 1 to 2048; router memory constraints may prevent the maximum table size from being used.

```
IPX config>set maximum services
New Service table size [32]? 32
```

Specifying SAP Update Interval

The IPX Service Advertising Protocol (SAP) interval lets you configure the time between IPX SAP updates on a per-interface basis. All router interfaces on the same network must use the same SAP interval. This interval determines both the age-out time for table information, and the interval between broadcasts to router interfaces.

You can configure the SAP interval from 1 to 1440 minutes (24 hours). Increasing the SAP interval reduces traffic on WAN lines and dial circuits. It also prevents dial-on-demand circuits from dialing out as often.

Note: While complete SAP advertisements are controlled by this interval, the router still propagates network topology changes as quickly as it learns them.

The SAP interval is not configurable on the Novell file server.

```
IPX config>set sap-update
Which interface [0]? 2
SAP timer value(minutes) [1]? 4
```

Configuring Multiple Routes

You can configure IPX so that it keeps more than one routing table entry for the same destination network. The benefit of this feature is that if a route goes down, the alternate route is used immediately. The router does not have to wait for a RIP broadcast, which could take from a few seconds to a minute, to learn a new route. The router stores only equal-cost paths in the routing table.

Use the following command to configure the maximum number of routes that will be stored in the routing table for each destination. The range is 1 to 64. The default is 1.

```
IPX config>set maximum routes-per-destination
New maximum number of routes per destination net [1]? 4
```

Use the following command to set the total number of entries kept in the routing table. The range is 1 to 4096. The default is 32. Set the number of entries to at least the same size as the RIP network table. (Configure the size of the RIP network table using the **set maximum networks** command explained in this chapter.)

```
IPX config> set maximum total-route-entries
New route table size [32]? 40
```

Configuring Global IPX Filters (IPX Access Controls)

Global IPX filters are applied to all IPX interfaces. They can be used to prevent the router from forwarding packets based on IPX addresses (network/host/socket). You can use global IPX filters to provide security or to stop the forwarding of packets from “noisy” applications beyond the area of interest.

Global IPX filters are based on the originating IPX source address and the ultimate destination IPX address. Intermediate hop addresses are not important.

An IPX address (source or destination) for a global filter consists of an IPX network number, an IPX host number, and a range of IPX socket numbers that are specified in hexadecimal. The network number and host number can be

specified as 0, which is a wildcard that matches all network and host numbers, respectively. A range of 0 to FFFF is a wildcard for sockets.

The global filter list is an ordered list of entries. Each global filter entry can be configured as inclusive or exclusive. The router compares packets it receives against the global filter list.

- If a packet matches an inclusive entry, the router forwards the packet.
- If a packet matches an exclusive entry, the router drops the packet.
- If the router reaches the end of the list without matching the packet to an entry, the router drops the packet. (This is equivalent to having a wildcard exclusive entry at the end of the list.)

When creating global filter lists, consider the following things about IPX:

- First, never block the RIP and SAP sockets (0x0453 and 0x0452). RIP and SAP are required to correctly forward IPX packets.
- Remember that the global filter list applies to all interfaces. You will have to use source and/or destination network numbers in the global filters to enact directional controls.
- Understand where the services you are trying to protect are located. At the `IPX>` prompt, enter the **slist** command to determine the address of a service.

  Note: All services on a Novell file server (version 3.0 or higher) are on the server's internal network, usually at host 0000 0000 0001. Because that internal network number is unique over an entire IPX network, you can protect it by blocking all packets to the internal network socket range 0–FFFF. To block only the file server, use a socket range of 0451–0451.
- When extracting socket numbers from an **slist** to build a global filter list, remember that some services have fixed socket numbers and some have dynamic (temporary) socket numbers. Because sockets in the range 4000–7FFF are dynamic, there is no guarantee that the service will have the same socket number the next time the file server is rebooted. However, socket numbers in the range 8000–FFFF are assigned by Novell, and will generally remain constant.

Note: The global filters and interface filters are mutually-exclusive. If global SAP filtering is enabled, interface SAP filters cannot be enabled (and vice versa). If global IPX filtering is enabled (*access-controls*), interface IPX filters cannot be enabled (and vice versa).

The router examines each IPX frame to see if it matches an entry in the global filter list. It applies the first match, therefore the order of global filters is critical. The router examines IPX packets for the following criteria:

1. Type of global filter (two types):
   a. Inclusive, indicating that if the packet matches the following criteria, forward it
   b. Exclusive, indicating that if the packet matches the following criteria, discard it
2. Destination network - taken directly from the packet's IPX destination network field.

3. Destination host - taken directly from the packet's IPX destination host field.
4. Starting/Ending destination socket - taken directly from the packet's IPX destination socket field (not host field). (The socket number is the location within the protocol that binds the packet to an application service.)
5. Source network - taken directly from the packet's IPX source network field.
6. Source host - taken directly from the packet's IPX source host field.
7. Starting/Ending source socket - taken directly from the packet's IPX source socket field.

The result of the following example would be to forward only those IPX packets from any client on IPX net 1871, destined for the NCP application, on the Novell File Server 0000 C93A 0912, on network 18730. All other traffic would be dropped.

```
IPX config>add access control
Enter type [E]? I
Destination network number (in hex) [ ]? 18730
Destination host number (in hex) [ ]? 0000C93A0912
Starting destination socket number (in hex) [ ]? 0451
Ending destination socket number (in hex) [ ]? 0451
Source network number (in hex) [ ]? 1871
Source host number (in hex) [ ]? 0
Starting source socket number (in hex) [ ]? 4000
Ending source socket number (in hex) [ ]? 7FFF
```

Global SAP Filters

Global SAP filters apply to all interfaces. They can be used to prevent service advertising information from being propagated through the router. There are four primary reasons to use global SAP filters:

- You are using servers with small bindery sizes (for example, NetWare Version 2.15 or lower) and must limit the amount of information in the SAP database.
- You do not want to advertise certain services outside the local area, because remote access to them would be inappropriate.
- You want to remove clutter from the SAP table.
- You want to reduce needless SAP advertisements on WAN links, since SAP advertisements can consume a considerable amount of WAN bandwidth.

Note: None of these reasons explicitly mentions security. Global SAP filters cannot protect a service. All that SAP does is provide a name-to-address translation for services. If a potential intruder knows the address of the service, blocking its advertisement via global SAP filters will not protect the service. Only access controls can provide security.

The global SAP filter is based on setting a maximum hop count for a particular service, or group of services. Any matching service advertisement received with the specified hop count (or less) is accepted into the SAP table. Others are ignored. Only those services in the SAP database are re-advertised or used to answer queries.

Note: The router allows you to enter service names in 7-bit ASCII only. Some service names use binary data, in violation of Novell SAP specifications. You will not be able to filter those services by name.

A global SAP filter can apply to all services of a type. Novell assigns 4-digit hexadecimal type numbers for each type of service. Alternately, a global SAP filter can apply to one particular service of a type. This is done by specifying the name of the service.

There can be several servers of the same service type, each with a unique service name. In this case, you can configure multiple global SAP filters with the same service type to filter unique service names, or you can configure a single SAP filter which filters the service type for all service names (wildcard filter).

Creating Global SAP Filters

To configure global SAP filters:

1. Enter **add filter** at the `IPX Config>` prompt. You must specify several key entries that are normally found in the SAP broadcasts:
   a. Number of hops. This entry indicates the hop count allowed for a SAP entry (if higher, discard).
   b. Service type
   c. Service name
2. Enter **set filter on** at the `IPX Config>` prompt to enable the filter.

The following example shows the creation of a global SAP filter against a specific print server.

```
IPX config>add filter
Maximum number of hops allowed [1]? 2
Service type [0]? 0047
Optional service name [ ]? rem-ptr1
IPX config> set filter on
```

This global SAP filter causes the router to ignore SAP advertisements from any print server (service type 0047) named **rem-ptr1** that is more than two hops away. The filter prevents the router from propagating advertisements that match these criteria.

Determining the Service Type for a Global SAP Filter

To determine the SAP type for a filter you want to establish, follow these steps:

1. At the * prompt, enter **talk 5**. Then, at the + prompt, enter **protocol ipx**.

   At the `IPX>` prompt enter **slist**. Note the entry for the services you want to filter.
2. At the `*` prompt, enter **talk 6**. Then, at the `Config>` prompt, enter **protocol ipx**. Add the appropriate global SAP filter and the appropriate hop count for the service you want to filter.

3. After creating the filter, restart the router.
4. If you have successfully filtered a service, it should no longer be listed. Check that the service is no longer listed by entering **slist** at the `IPX>` prompt.

IPX Interface Filters - Overview

The IPX routing feature supports four types of interface-based filters: ROUTER, RIP, SAP, and IPX. One *input* and one *output filter* can be defined per interface. Filter criteria, referred to as *items*, are assembled into *filter-lists* and are then attached to the input and/or output filters. A filter-list can be attached to more than one filter. This prevents you from having to configure the same filter criteria on multiple interfaces.

Note: The global filters and interface filters are mutually-exclusive. If global SAP filtering is enabled, interface SAP filters cannot be enabled (and vice versa). If global IPX filtering is enabled (*access-controls*), interface IPX filters cannot be enabled (and vice versa).

Configuring IPX Interface Filters

To configure IPX Interface Filters:

1. Create a filter-list and give it a name, using the **create list** command.
2. Modify the filter-list using the **update** command and its subcommands to specify the filter criteria and whether this filter-list is inclusive or exclusive.
3. Create a filter on the desired interface using the **create filter** command, specifying whether it is an input or output filter.
4. Enable the filter using the **enable** command.
5. Attach filter-lists to the filter using the **attach** command.
6. Set the default action for the filter using the **default** command. The default action will be taken if no match is made on any of the attached filter-lists.

There are also commands to delete a filter on a network interface, disable a filter on a network interface (or all network interfaces), detach a filter-list from a filter, move the filter-lists within the filter (because the filter-lists are ordered), list a filter, and set the size of the filter cache (for IPX Filtering only).

ROUTER Filtering

The ROUTER Filter operates on the IPX header of all received RIP response packets. Output ROUTER filtering is not supported. ROUTER filtering can be used to group individual IPX networks into several distinct IPX internets by controlling which routers are allowed to exchange routing information.

RIP Router Filters are kept in ordered lists of items by interface. The items are applied in order to each received RIP response packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = discard packet, Include = receive packet for processing). Because Excluded packets are discarded, the information contained in their network entries is not entered into the RIP routing tables. If no match is found, the specified default filter action is performed.

RIP Filtering

The RIP filter operates on the network entries of RIP response packets. It can be used to control the extent to which routing information about selected networks is disseminated. As an *input* filter, this filter can prevent the *storing* of routing information about selected networks. This prevents ***all*** other networks from learning about the selected networks (at least through this router).

RIP filters (input) are kept in ordered lists of items by interface. The items are applied in order to each network entry in each received RIP response packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = ignore network entry, Include = process network entry). Because Excluded network entries are ignored, they are not entered into the RIP routing tables. If no match is found, the specified default filter action is performed.

As an *output* filter, this filter can prevent the *advertising* (as opposed to the storing) of routing information about selected networks. It prevents *some* (as opposed to all) networks from learning about the selected networks (at least through this router).

RIP filters (output) are kept in ordered lists of items by interface. The items are applied in order to each network entry to be transmitted in a RIP response packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = exclude network entry from packet, Include = include network entry in packet). This filter has no effect on the contents of the RIP routing tables. If no match is found, the specified default filter action is performed.

SAP Filtering

The SAP filter operates on the server entries of all SAP response packets. It can be used to control the extent to which information about services is disseminated, and can reduce the amount of SAP traffic on lower speed WANs.

As an *input* filter, this filter can prevent the *storing* of service information about selected servers. This prevents ***all*** other networks from learning about the selected servers (at least through this router).

SAP filters (input) are kept in ordered lists of items by interface. The items are applied in order to each server entry in each received SAP response packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = ignore server entry, Include = process server entry). Because Excluded server entries are ignored, they are not entered into the SAP services table. If no match is found, the specified default filter action is performed.

As an *output* filter, this filter can prevent the *advertising* (as opposed to the storing) of service information about selected servers. This prevents *some* (as opposed to all) networks from learning about the selected servers (at least through this router).

SAP filters (output) are kept in ordered lists of items by interface. The items are applied in order to each server entry in each SAP response packet to be transmitted. If a match is found, the action specified in the matching filter-list is performed (Exclude = exclude server entry, Include = include server entry in packet). This filter has no effect on the contents of the SAP services table. If no match is found, the specified default filter action is performed.

IPX Filtering

The IPX Filter operates on the IPX header of IPX packets. It can be used to control the extent to which selected servers and workstations are allowed to communicate with other selected servers and workstations, based on source and destination network, node, and socket fields, as well as protocol type and hop count.

As an *input* filter, a match that indicates that the packet should be discarded prevents the packet from being transmitted on ***all*** interfaces.

IPX Filters (input) are kept in ordered lists of items by interface. The items are applied in order to each received IPX packet. If a match is found, the action specified in the matching filter-list is performed (Exclude = discard packet, Include = receive packet for processing or forwarding). If no match is found, the specified default filter action is performed.

As an *output* filter, the decision whether to forward the packet is made based on the output interface, and therefore might allow a received packet to be forwarded out on one interface but not out on some other interface.

IPX filters (output) are kept in ordered lists of items by interface. The items are applied in order to each IPX packet to be transmitted. If a match is found, the action specified in the matching filter-list is performed (Exclude = discard packet, Include = transmit packet). If no match is found, the specified default filter action is performed.

Because IPX filters are invoked for each received packet, it is recommended that they be used only where a high degree of specificity is required (that is, where the RIP Router, RIP and SAP filters cannot be used). Generally, the RIP filters deal with internetworking between ***all*** stations on a particular set of networks; the SAP filters control which servers are reachable by workstations throughout the internetwork; the IPX filters deal with internetworking between ***individual*** workstations (or individual applications on individual workstations).

“ IPX Interface Filter Configuration Commands” on page 20-29 describes in more detail the commands used to configure IPX Interface Filters.

IPX Performance Tuning

The IPX router implements a dual path for packet forwarding, a fast path and a slow path, to route traffic more efficiently.

The fast path forwards only data packets, while a slower path handles administration packets, such as RIP and SAP packets. Fast path uses an address cache that enables the router to forward a packet quickly.

The slower routing table lookups are performed only during the creation of a cache entry. The cache has an aging mechanism that allows overflows to be dealt with intelligently. You can configure the cache size through the IPX configuration menu.

The IPX fast path cache includes two entries: local and remote. Each entry can handle the requirements of that type of addressing.

The cache commands are used to set a limit on the maximum number of entry types allowed in the cache.

Local Cache

The size of the local cache should equal the total number of clients on each router's local or client network plus a 10% buffer to prevent excessive purge requests. Using the example in Figure 20-1 on page 20-12, 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 20-1 on page 20-12, 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
```

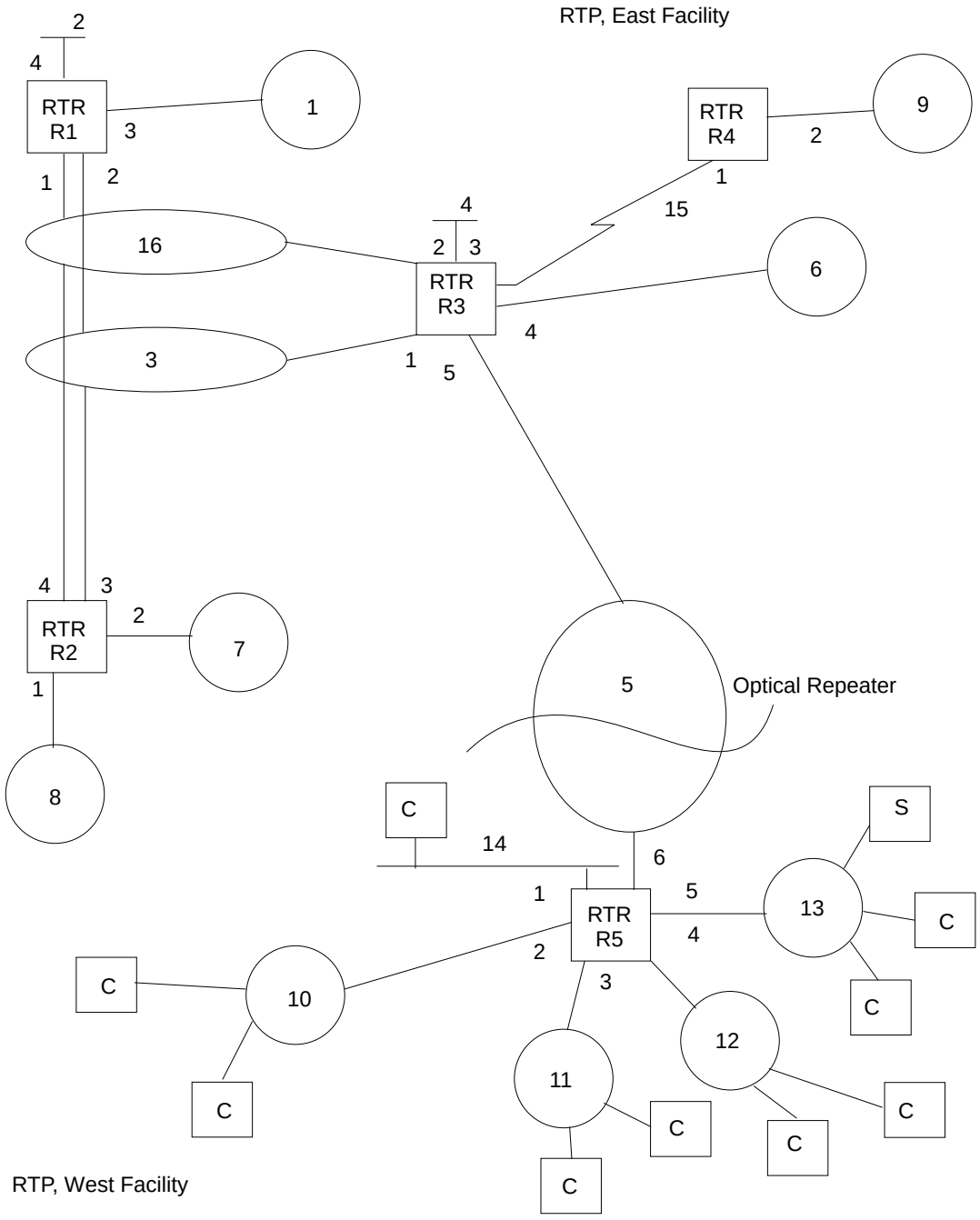


Figure 20-1. Sample IPX Network

Split-Horizon Routing

Split-horizon is a method of routing that avoids broadcasting RIP and SAP updates to the router from which they were learned.

Generally, split-horizon should be enabled on every interface to prevent packets from counting to infinity and to avoid unnecessary RIP and SAP advertisements. However, there are some cases, such as partially-meshed frame-relay, ATM, and X.25 configurations, where it may be necessary to disable split-horizon.

A Partially-meshed RFC 1483-Supported IPX Routing configuration is another case where it may be necessary to disable split-horizon.

In a partially-meshed frame-relay network, as shown in Figure 20-2, the routers at the branches cannot communicate with each other unless the router at headquarters broadcasts all routing information to all other routers. In this case, split-horizon should be disabled on the frame-relay interface at headquarters, and enabled at each of the branches to keep them from generating unnecessary traffic.

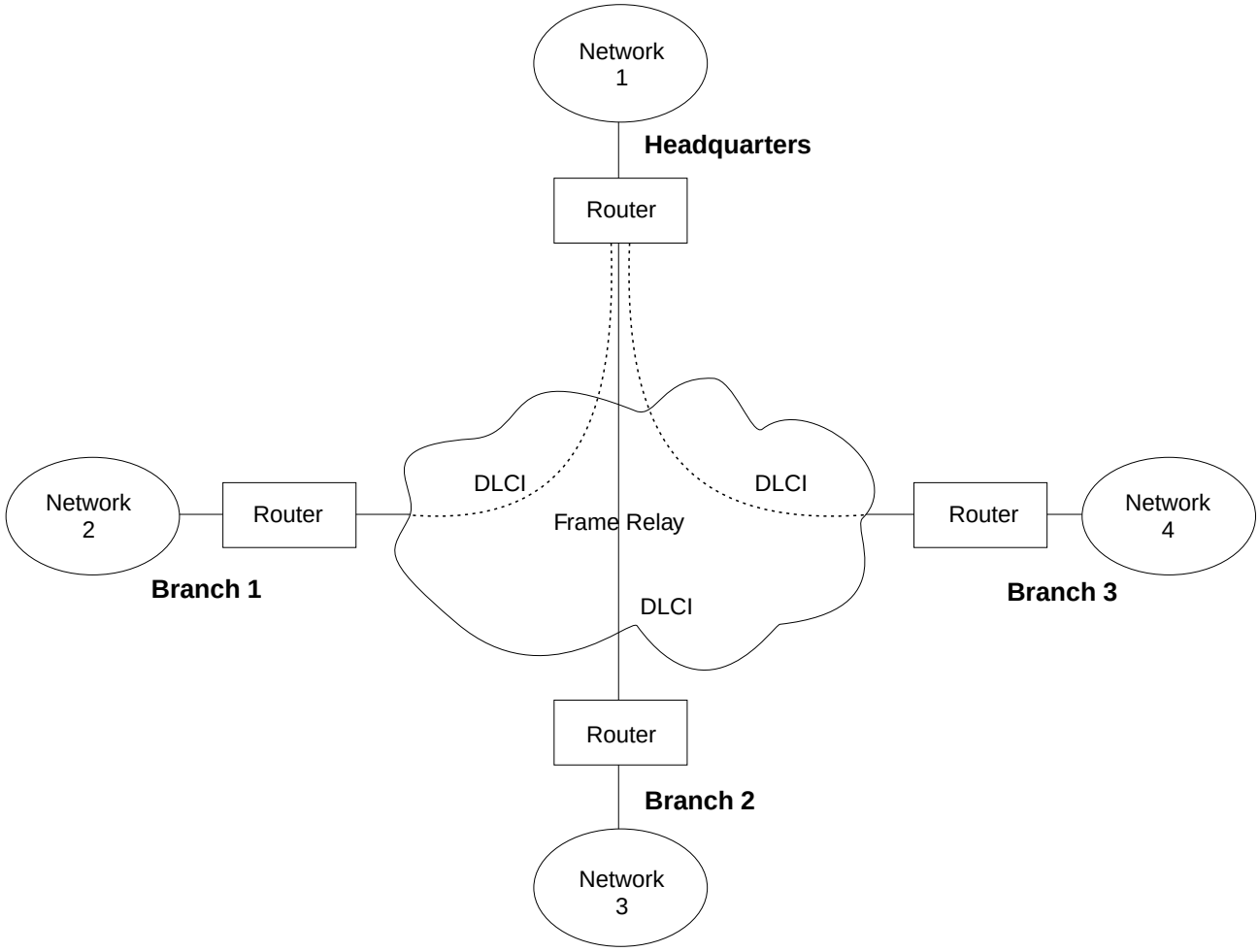


Figure 20-2. Partially Meshed Frame-Relay Network

If you do need to change the split-horizon setting, use the **set split-horizon** command as follows:

```
IPX Config>set split-horizon enabled
Which interface [0]? 2

IPX Config>set split-horizon disabled
Which interface [0]? 2

IPX Config>set split-horizon heuristic
Which interface [0]? 2
```

Accessing the IPX Configuration Environment

To access the IPX configuration environment, enter the following command at the Config> prompt:

```
Config> protocol IPX
IPX Protocol user configuration
IPX Config>
```

IPX Configuration Commands

This section discusses the IPX configuration commands. Table 20-1 lists the IPX configuration commands. These commands specify the network parameters for router interfaces transmitting IPX packets. These commands are entered at the IPX config> prompt. To activate the configuration changes, restart the router.

Table 20-1. IPX Configuration Commands Summary

Command	Function
? (Help)	Lists all of the IPX configuration commands or lists the options associated with specific commands.
Add	Adds global IPX filters (access controls), and global SAP filters.
Delete	Deletes global IPX filters (access controls), and global SAP filters.
Disable / Enable	Disable or enable IPX globally or on specific interfaces, disable or enable reply to SAP get-nearest-server requests, disable or enable RIP-SAP broadcast pacing, and disable or enable IPXWAN on specific interfaces.
Filter-lists	Accesses the IPX *filter-type*-List Config> prompt. This is the environment where the IPX interface filters (Router, RIP, SAP, and IPX) are configured.
Frame	Specifies the data link format for Ethernet and Token-Ring interfaces. This also applies to Token-Ring and Ethernet LAN Emulation Clients.
List	Displays the current IPX configuration.
Move	Changes the line numbers set when adding access control.
Set	Sets the host number, IPXWAN router name and node ID, IPXWAN connection timeout and retry timer, IPX network numbers, maximum RIP and SAP table sizes, local and remote cache sizes, global IPX filter (access controls) and global SAP filter states, cache sizes, RIP and SAP update intervals, and split-horizon usage.
Exit	Exits the IPX configuration process and returns to the CONFIG environment.

? (Help)

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

Syntax: ?

Example: ?

```
disable
enable
exit
frame
list
set
add
delete
move
filter-lists
```

Add

Use the **add** command to add global IPX filters (access controls) to your IPX configuration, which determines whether the router drops or forwards IPX packets. The **add** command also adds global SAP filters to your IPX configuration; this determines which SAP service advertisements will be ignored or accepted by the router.

Syntax: add access-control . . .
filter . . .

access-control *type dest-net dest-host dest-socket-range src-net src-host src-socket-range*

Determines whether to pass a packet at the IPX level. IPX access controls provide a global access control function at the IPX packet level for the IPX protocol. The access control list is an ordered set of entries that the router uses to filter packets. Each entry can be either Inclusive or Exclusive. Each entry has source and destination network numbers, host addresses, and socket ranges.

When a packet is received from a network for the IPX protocol, and access control is enabled, it is checked against the access control list. It is compared with the net/address/socket pairs in the list until there is a match. If there is a match and the entry is of the Inclusive type, reception of the packet (and potential forwarding) proceeds. If the matching entry is of the Exclusive type, the packet is dropped. If there is no match, the packet is also dropped.

After you create an access-control list with the **add access-control** command, enable the entries with the **set access-control on** command. Use the **move** command to change the order of the access-control list.

Note: Access controls apply to all received packets. If you do not enable reception of RIP (socket 453 hexadecimal) or SAP (socket 452 hexadecimal) packets, the IPX forwarder will be nonfunctional.

```
add access I 0 0 453 453 0 0 0 FFFF
add access I 0 0 452 452 0 0 0 FFFF
```

```
Enter type [E] i
Destination network number (in hex) [0]? 0
Destination host (in hex) [ ]? 0
Starting destination socket number in hex [0]? 452
Ending destination socket number in hex [0]? 453
Source network number (in hex) [0]? 0
Source host number (in hex) [ ]? 0
Starting source socket number in hex [0]? 0
Ending source socket number in hex [452]? FFFF
```

Type
Identifies whether packets are sent or dropped for a specific address or set of addresses. Enter I for include. This causes the router to receive the packet and to forward it if it matches criteria in the remaining arguments. Enter E for exclude. This causes the router to discard the packets.

Dest-net
Network number of the destination. Enter the network number in hexadecimal.

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

Zero (0) specifies all networks.

Default Value: 0

Dest-host
Host number on the destination network. Enter the host number in hexadecimal.

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

Zero (0) specifies all hosts on the network.

Default Value: None

Dest-socket-range
Two numbers that specify an inclusive range of destination sockets. The destination socket value is used for filtering IPX packets.

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

Default Value: 0

Src-net
Network number of the source. Enter the network number in hexadecimal.

This parameter defines the network number of the source IPX network whose packets are filtered by this router.

If you choose to filter on *only* the source network value, the filter applies to all source sockets, source networks, packet types, and number of hops.

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

Zero (0) specifies all networks.

Default Value: 0

Src-host
Host number on the source network. Enter the host number in hexadecimal.

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

Zero (0) specifies all hosts on the network.

Default Value: None

Src-socket-range — Two numbers that specify an inclusive range of source sockets.

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

Default Value: 0

Note: It is not necessary to use access controls and SAP filters for IPX to work in a NetWare environment. Use them only if necessary.

Example: `add access-control E 201 1 451 451 329 0 0 FFFF`

This access control prevents all nodes on network 329 from accessing the file server with internal network number 201.

filter *hops service-type service-name*

Prevents NetWare bindery overflows for users on large networks by enabling you to determine the number of hops reasonable for a given service. IPX SAP filters allow the protocol to be configured to ignore certain entries in SAP advertisements. This is done to limit the size of the SAP database. This could be necessary due to size limitations in older versions of NetWare file servers. This could also be necessary to limit the amount of SAP data sent across WAN links.

The SAP filters are a global ordered list of filter entries. Each filter entry has a maximum hop count, a service type, and an optional service name. When a SAP response packet is received, each SAP entry is compared with the filter list. If the SAP entry matches an entry in the filter list and is greater than the specified hops, it is ignored and not entered into the local SAP database. If the SAP entry matches an entry in the filter list, and is less than or equal to the specified hops, it is accepted and entered into the local SAP database. If there is no match, the SAP entry is accepted. The arguments for this command are as follows:

Hops — Maximum number of hops permitted for the service.

Valid Values: An integer in the range of 0 - 16.

Default Value: 16

Service-type — Numeric service class.

Valid Values: A hexadecimal value in the range of 0000 - FFFF.

Use a value of X'0000' to filter all service types.

Default Value: none

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 48 ASCII characters (X'20' through X'7E'), with the exception of the following special characters: plus (+), minus (-), comma (,), semicolon (;), colon (:), slash (/), and back slash (\).

The question mark (?) and asterisk (*) characters serve as wildcard characters. The question mark may be used multiple times to represent any single character within the server name. The asterisk may be used multiple times to represent any portion of the server name. The question mark and asterisk may also be used together.

Default Value: none

Example: `add filter 2 039B NOTES-CHICAGO`

This example ignores all SAP advertisements for the Lotus Notes server "NOTES-CHICAGO" at more than 2 hops.

Delete

Use the **delete** command to delete a global IPX filter (access control) or global SAP filter.

Syntax: delete access-control . . .
filter . . .

access-control *line#*

Deletes the access control that matches the line number you enter. Enter the **list** command to display the current line numbers.

Example: `delete access-control 2`

filter *hops service-type service-name*

Deletes the specified SAP filter. You must type the SAP filter exactly as it appears when you run the list command. The arguments are as follows:

Hops — Maximum number of hops permitted for the service.

Valid Values: 0 to 16

Default Value: 16

Service-type — Numeric service class. Enter a 2-byte hexadecimal number.

Valid Values: X'0' 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 48 ASCII characters (X'20' through X'7E'), with the exception of the following special characters: plus (+), minus (-), comma (,), semicolon (;), colon (:), slash (/), and back slash (\).

The question mark (?) and asterisk (*) characters serve as wildcard characters. The question mark may be used multiple times to represent any single character within the server name. The asterisk may be used multiple times to represent any portion of the server name. The question mark and asterisk may also be used together.

Default Value: none

Example: `delete filter 2 039B NOTES-CHICAGO`

Disable

Use the **disable** command to disable IPX on specific interfaces, or to globally disable the IPX protocol. Also, use the **disable** command to disable replies to SAP get-nearest-server requests, RIP-SAP Broadcast Pacing on specific interfaces, or IPXWAN on specific interfaces.

Syntax: disable interface . . .
ipx
ipxwan . . .
reply-to-get-nearest-server . . .
rip-sap-pacing . . .

interface *interface#*
Prevents the router from sending IPX packets over specific interfaces.

Example: `disable interface 2`

ipx
Prevents the router from sending IPX packets over any of the interfaces.

Example: `disable ipx`

ipxwan *interface#*
Disables IPXWAN on specific serial interfaces using the point-to-point protocol (PPP).

Example: `disable ipxwan 2`

reply-to-get-nearest-server *interface#*
Prevents the router from responding to SAP get-nearest-server requests from workstations that are attempting to locate a server.

Note: Disabling this feature should be done with great caution. This command should be used only when there are multiple routers (or servers) on an IPX network and it is known that the “best” server is not behind this router.

Example: `disable reply 3`

rip-sap-pacing *interface#*
Prevents the router from pacing RIP and SAP periodic broadcast packets on specific interfaces. When pacing is disabled, RIP and SAP periodic broadcasts are transmitted on the interface with a 55 msec interpacket gap (the default setting). Enable pacing only on interfaces where RIP and

SAP broadcasts might cause congestion (for example, you can enable pacing on frame-relay or X.25 interfaces with many virtual circuits).

Example: `disable rip-sap-pacing 3`

Enable

Use the **enable** command to enable IPX on specific interfaces, or to globally enable the IPX protocol. Also use the **enable** command to enable replies to SAP get-nearest-server requests, RIP-SAP Broadcast Pacing on specific interfaces, or IPXWAN on specific interfaces.

Syntax: enable interface . . .
ipx
ipxwan . . .
reply-to-get-nearest-server . . .
rip-sap-pacing . . .

interface *interface# network#*

Allows the router to send IPX packets over specific interfaces. Every interface must have a unique network number that corresponds to the network number that is assigned to the attached IPX network. You will be prompted for a valid IPX network number if one has not already been configured.

Example: **`enable interface 2 4`**

ipx

Allows the router to send IPX packets over all of the interfaces on which IPX has been enabled.

Example: **`enable ipx`**

ipxwan *interface# timeout retry_timer*

Enables IPXWAN on specific interfaces using the Point-to-Point protocol (PPP). This command also queries for a connection timer value and a retry timer value. The enable command prompts for the same parameters as the **set ipxwan** command. This allows you to initially set IPXWAN parameters without having to use the set command. If you need to modify pre-configured parameters, then use the **set ipxwan** command.

timeout

This value specifies the IPX connection timeout period. A connection will time out if no IPXWAN packets are exchanged within the number of seconds specified by this parameter.

Valid values: An integer number of seconds in the range of 5 - 300.

Default Value: 60 seconds

retry_timer

This parameter specifies the amount of time to wait after a connection is timed out before trying to re-establish the connection.

Valid values: An integer number of seconds in the range of 5 - 600.

Default Value: 60 seconds

Example: **enable ipxwan 0 60 60**

```
Which interface [0]? 0
Connection Timeout (in sec) [60]?60
Retry Timer (in sec) [60]? 60
```

reply-to-get-nearest-server *interface#*

Allows the router to respond to SAP get-nearest-server requests from workstations that are attempting to locate a server. This is the default setting.

Note: Disabling this feature should be done with great caution. The **disable reply-to-get-nearest-server** command should be used only when there are multiple routers (or servers) on an IPX network and it is known that the "best" one is not behind this router.

Example: **enable reply 2**

rip-sap-pacing *interface#*

Enables pacing of RIP and SAP periodic broadcast packets on specific interfaces. When pacing in enabled, RIP and SAP periodic broadcasts are transmitted on the interface with an interpacket gap calculated by the router (a value from 55 msec to 5 seconds).

Note: The router calculates an interpacket gap that guarantees that broadcast completion within the configured RIP and SAP update intervals. Configuring these intervals to a larger value may be necessary for the router to calculate a sufficiently large interpacket gap.

Pacing should be enabled only on interfaces where RIP and SAP broadcasts might cause congestion (for example, on frame-relay or X.25 interfaces with many virtual circuits).

Example: **enable rip-sap-pacing 3**

Filter-lists

Use the **filter-lists** command to access the `IPX filter-type-List Config>` prompt. Valid filter list types are router, rip, sap, and ipx.

For information about the commands available at the `IPX filter-type.-List Config>` prompt, see " IPX Interface Filter Configuration Commands" on page 20-29.

Syntax: filter-lists router
rip
sap
ipx

Example: **filter-lists router**

Frame

Use the **frame** command to specify the packet format for IPX interfaces. (Encapsulation can also be set using the CONFIG **network** command.)

Note: When there are incorrect or invalid configuration records, the default frame values are used.

Syntax: frame ethernet_II . . .
ethernet_8022 . . .

ethernet_8023 . . .
ethernet_SNAP . . .
token-ring MSB . . .
token-ring LSB . . .
token-ring_SNAP MSB. . .
token-ring_SNAP LSB. . .
fddi
fddi_snap

ethernet_type *interface#*

Selects the Ethernet encapsulation format. This is required if you are using NetWare-VMS on the Ethernet, and is often used when there are ISO nodes on the same Ethernet. The following options are available:

- ethernet_II (default of NetWare 4.0 and greater) - uses Ethernet version 2.0 protocol 81-37.
- ethernet_8022 - uses Ethernet 802.3 with 802.2 SAP E0.
- ethernet_8023 (default of pre-NetWare 4.0 and lower) - uses Ethernet 802.3 without any 802.2 header.
- ethernet_SNAP - uses 802.3, 802.2 with SNAP PID 00-00-00-81-37.

  Note: The ethernet_SNAP encapsulation it is not architecturally valid and is not fast-pathed. No cache entries will appear for network entries using this encapsulation.

The default value for Ethernet frames is “ethernet_8023.”

Example: `frame ethernet_II 1`

token-ring_type *interface#*

Selects the token-ring encapsulation format. The default value is “token-ring MSB.” The following options are available:

- token-ring MSB (router default) - uses 802.5 with 802.2 SAP E0, and uses the noncanonical format for host addresses in the IPX packet header. The router builds outgoing packets with a three-byte 802.2 header (0xE0, 0xE0, 0X03).
- token-ring LSB - uses 802.5 with 802.2 SAP E0, and uses the canonical format for host addresses in the IPX packet header.
- token-ring_SNAP MSB - uses 802.5, 802.2 with SNAP PID 00-00-00-81-37, and uses the noncanonical format for host addresses in the IPX packet header.
- token-ring_SNAP LSB - uses 802.5,802.2 with SNAP PID 00-00-00-81-37, and uses the canonical format for host addresses in the IPX packet header.

Example: `frame token-ring_SNAP MSB 3`

```
which interface [0]? 3
```

Example: `frame token-ring SNAP PID 00-00-00-81-37`

fddi

Selects the IPX encapsulation to FDDI IEEE 802.2.

Example: `IPX Config> fddi`

fddi_snap

Selects the IPX encapsulation to FDDI SNAP.

Example: `IPX Config>` **`fddi_snap`**

List

Use the **list** command to display the current IPX configuration.

Syntax: list

Example: **`list`**

```
IPX globally                enabled
Host number (serial line)   020000000200
Router Name (IPXWAN)
NodeID (IPXWAN)             0
Maximum networks            32
Maximum total route entries             128
Maximum routes per dest. network        3
Maximum services                        32
Maximum Network Cache entries           64
Maximum Local Cache entries             64

List of configured interfaces:
                Frame                   SAP nearest   Split
Ifc  IPX net #  Encapsulation          server reply  Horizon    IPXWAN

0    177        TOKEN-RING      MSB    Enabled       Heuristic  N/A
1    183        N/A                    Enabled       Heuristic  N/A
5    184        N/A                    Enabled       Heuristic  N/A

RIP/SAP Timer Intervals and Pacing:

                 SAP Interval  RIP Interval
Ifc  IPX net #   (Minutes)     (Minutes)     Pacing
---  ---------   ------------  ------------  -------
0        177        1             1          Disabled
1        183        1             1          Disabled
5        184        30            30         Enabled


IPX SAP Filter is: disabled
No IPX SAP Filter records in configuration.
IPX Access Controls are: disabled
No IPX Access Control records in configuration.
```

IPX globally	Indicates whether IPX is globally enabled or disabled.
Host number	The IPX host number to be assigned to serial interfaces. You can change this number with the IPX **set** command.
Router name	The user-assigned IPXWAN router name.
Node ID	The user-assigned IPXWAN node-id.
Maximum networks	The size of the IPX RIP network table, which is the maximum number of IPX networks.
Maximum routes	The size of the IPX RIP routes table, which is the maximum number of routes to IPX networks.
Maximum routes-per-network	The configured number of maximum routes-per-network.
Maximum services	The size of the IPX SAP service table, which is the maximum number of IPX servers.
Maximum network cache entries	The maximum number of network cache entries.

Maximum local cache entries	The maximum number of local cache entries.
List of configured interfaces	The following is displayed for each interface on which IPX is enabled: • Interface number • IPX network number • Type of encapsulation • Whether reply to SAP get-nearest-server requests is enabled • Whether split-horizon is enabled, disabled, or heuristic. • Whether IPXWAN is enabled.
RIP/SAP Timer Intervals and Pacing	The following is displayed for each interface on which IPX is enabled: • Interface number • IPX network number for a particular interface • Delay in minutes between complete RIP advertisements • Delay in minutes between complete SAP advertisements • Whether RIP-SAP broadcast pacing is enabled.
IPX SAP filter	Indicates whether the Global SAP filter is enabled or disabled, and lists the configured Global SAP filters.
IPX access controls	Indicates whether the Global IPX filters (access controls) are enabled or disabled, and lists the configured Global IPX filters (access controls).

Move

Use the **move** command to change the line numbers for the Global IPX filters (access controls) After you move the lines, they are renumbered to reflect the new order.

Syntax: move access-control *line# line#*

Example: **`move 5 2`**

```
About to move:
 # T Dest Net         Host  Sck  Sck  Src Net         Host  Sck  Sck
 5 E       30 020000000006   30   32        2 020000000004   45   46
to be after:
 2 I     2487 020300000008    0    0    45230 020000000042    0    0
           Are you sure this is what you want to do(Yes or [No]):
```

Set

Use the **set** command to configure many of the operational parameters of the IPX protocol, such as the IPX network number for each interface, the sizes of the RIP and SAP tables, the sizes of the routing caches, and the IPXWAN parameters.

Syntax: set
- access-control . . .
- filter . . .
- host-number . . .
- ipxwan . . .
- local-cache size . . .
- maximum routes-per-destination . . .
- maximum networks . . .
- maximum services . . .
- maximum total-route-entries . . .
- name . . .
- net-number . . .
- node-id . . .
- remote-cache size . . .
- rip-update-interval . . .
- sap-update-interval . . .
- split-horizon . . .

access-control *on* or *off*

Turns the global IPX filters (access controls) on or off. Enter **on** or **off**.

Example: `set access-control on`

filter *on* or *off*

Turns the global SAP filters on or off. Enter **on** or **off**.

Example: `set filter on`

host-number *host#*

Specifies the host number used for serial interfaces running IPX. Each IPX router operating over serial interfaces must have a unique host number. This is required because serial interfaces do not have hardware node addresses from which to build a host number.

Valid Values: An 12-digit hexadecimal number in the range of 0000 0000 0001 - FFFF FFFF FFFE.

Default Value: none

This number must be unique on each router.

Example: `set host-number 0000000000F4`

Note: IPXWAN requires a router node-ID and name to be configured. Use the **set node-ID** and **set name** commands to configure these parameters. In addition, in order to interoperate with the IBM 6611, the node address used on the IPXWAN interface must be set to the node-ID ***followed by four zeros***. For example, if the IPXWAN node-ID is 454, the host-number must be set to 4540000. Use the **set host-number** command to configure this parameter.

ipxwan *interface# timeout retry_timer*

Sets up or modifies an interface to use the IPXWAN protocol when starting IPX on a serial interface using the PPP. Before the **set ipxwan** command can be invoked, IPXWAN must be enabled using the **enable ipxwan** command. This command also queries for a connection timer value and a retry timer value.

connection timeout

This value specifies the IPX connection timeout period. A connection will time out if no IPXWAN packets are exchanged within the number of seconds specified by this parameter.

Valid values: An integer number of seconds in the range of 5 - 300.

Default Value: 60 seconds

retry timer

This parameter specifies the amount of time to wait after a connection is timed out before trying to re-establish the connection.

Valid values: An integer number of seconds in the range of 5 - 600.

Default Value: 60 seconds

Example: **`set ipxwan`**

```
Which interface [0]? 1
Connection Timeout (in sec) [60]?60
Retry timer (in sec) [60]? 60
```

local-cache size *size*

Specifies the size of the local cache routing table.

The size of the local cache should equal the total number of clients on each router's local or client network plus a 10% buffer to prevent excessive purge requests.

Valid Values: The range is 1 to 10000.

Default Value: 32. For more information, see "Local Cache" on page 20-11 and "Remote Cache" on page 20-11.

Example: **`set local-cache size`**

```
New IPX local node cache size [32]? 64
```

maximum routes-per-destination *routes*

Specifies the maximum number of routes per destination network to store in the IPX RIP routes table.

Valid Values: An integer in the range of 1 - 64.

Default Value: 1. For additional information on multiple routes, see "Configuring Multiple Routes" on page 20-4.

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 - 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 20-4.

Example: `set maximum total-route-entries 40`

name *router_name*

Lets you assign a symbolic name to the router. IPXWAN requires a router to have a primary network number and a name.

Valid Values: A variable length string of 1 to 47 characters.

The router_name can contain the characters A through Z, 0 through 9, underscore (_), hyphen (-), and "at" sign (@).

Default Value: none.

Example: `set name newyork_accounting`

net-number *interface# ipx-net#*

Assigns an IPX network number to the associated directly-connected network. Every IPX interface must have a unique network number.

Valid Values: 1 to 8 hexadecimal digits in the range of 1 - FFFF FFFE.

Do not assign 0 or FFFF FFFF.

Default Value: none

Example: `set net-number 2 180`

node-id *primary-net#*

Lets you assign a primary network number. IPXWAN requires a router to have a primary network number and a name. The “node-id” is the primary network number for the router and must be assigned before the exchange of IPXWAN packets can begin.

Valid Values: 1 to 8 hexadecimal digits in the range of 1 - FFFF FFFE.

Default Value: none

Do not assign 0 or FFFF FFFF as the *primary-net#.* This number is for the router as a whole. In NetWare file server terms, it is the “internal” network number. This number must be unique among all the network numbers in the IPX internet.

Example: `set node-id 23`

remote-cache size *size*

Specifies the size of the remote cache routing table.

The size of the remote cache should equal the total number of remote networks used by the router plus a 10% buffer to prevent excessive purge requests.

Valid Values: The range is 1 to 10000.

Default Value: 32.

Example: **`set remote-cache size`**

```
New IPX remote network cache size [32]? 64
```

rip-update-interval *interface# minutes*

Specifies the time delay in minutes between complete RIP updates given on an interface.

Increasing the RIP interval reduces traffic on WAN lines and dial circuits. It also prevents dial-on-demand circuits from dialing out so often.

Note: While complete RIP advertisements are controlled by the interval, the router still propagates network topology changes as quickly as it learns them.

Valid Values: The range is from 1 to 1440 minutes.

Default Value: 1 minute. For additional information on RIP interval, see "Specifying RIP Update Interval" on page 20-3.

Example: **`set rip-update-interval`**

```
Which interface [0]? 0
RIP Timer Value (minutes) [1]?2
```

sap-update-interval *interface# minutes*

Specifies the time delay in minutes between complete SAP updates given on an interface.

Valid Values: The range is from 1 to 1440 minutes.

Default Value: 1 minute.

Example: **`set sap-update-interval`**

```
Which interface [0]? 0
SAP Timer Value (minutes) [1]? 2
```

split-horizon *value interface#*

Specifies the type of split-horizon on the specified interface.

The value *enabled* enables split-horizon on the specified interface. The value *disabled* disables split-horizon on the specified interface. The value *heuristic* enables split-horizon on all types of interfaces except frame-relay. For frame-relay, split-horizon is enabled only if the specified interface has exactly one PVC defined, otherwise split-horizon is disabled.

Generally, split-horizon should be set to *enabled*. It is sometimes necessary to disable split-horizon for partially-meshed frame-relay, X.25, and ATM configurations. For additional information on split-horizon, see "Split-Horizon Routing" on page 20-12.

Example: **`set split-horizon enabled 0`**

keepalive-table-size *value interface#*

Sets the number of entries that the Keepalive table holds. These entries include all current client/server and server/server pairs connected over the WAN link. The default is 32. The range is 1 to 250.

Example: `set keepalive-table-size`

```
which interface[0]? 0
Number of entries [32]?
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Accessing the IPX Interface Filter Configuration Environment

To access the IPX Interface Filter configuration environment, enter the following command at the `IPX config>` prompt:

```
IPX Config> filter-lists type
IPX type-List Config>
```

Where *type* is the type of IPX filter to be configured. Valid types are *router, rip, sap,* and *ipx*.

IPX Interface Filter Configuration Commands

This section lists and then explains the commands to configure the IPX interface-based filters; ROUTER, RIP, SAP, and IPX. To configure these filters, enter the `filter-lists type` command at the `IPX Config>` prompt, and then enter the configuration commands at the `IPX type-List Config>` prompt.

Table 20-2 (Page 1 of 2). IPX Filter Configuration Command Summary

Command	Function
? (Help)	Lists all interface filter configuration commands or lists the options associated with specific commands.
Attach	Attaches a specified filter-list to a specified filter.
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.

Table 20-2 (Page 2 of 2). IPX Filter Configuration Command Summary	
Command	**Function**
Exit	Exits the IPX Interface Filter configuration environment.

? (Help)

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

Syntax: ?

Example: ?

```
attach
create
default
delete
detach
disable
enable
list
move
set-cache
update
exit
```

Attach

Use the **attach** command to attach a filter-list to a filter.

Syntax: attach *list-name filter#*

list-name

Specifies the name of the filter-list. The **list** command can be used to display a list of the configured filter-list names.

Valid Values: Any alphanumeric string up to 16 characters

Default Value: None

filter#

Specifies the number of the filter. A numbered list of configured filters can be obtained using the list command.

Example: attach test_list 1

Create

Use the **create** command to create a filter-list or filter.

Syntax: create list ...
filter ...

list *list-name*

Creates a list with the specified name.

Valid Values: Any alphanumeric string up to 16 characters

Default Value: none

You can also enter the **create list** command with no list name. You will then be prompted for the list name.

Example: `create list example_list`

filter *direction interface_number*

Creates a filter for the specified direction on the specified interface. Specify *input* to filter packets received on the specified interface. Specify *output* to filter packets to be sent by the specified interface.

A number is automatically assigned to a filter when it is created and from that point on is used to identify the filter, rather than having to key in the interface and direction (input or output) for all subsequent commands.

Example: `create filter input 1`

Default

Use the **default** command to set the default action for a filter. The default action is taken when no match is found for any of the filter items.

Syntax: default *action filter#*
Example: `default exclude 1`

action

Specifies the default action. **Include** specifies that when no match is found to any of the filter items, the packet is processed. **Exclude** indicates that when no match is found, the packet is dropped.

filter#

Specifies the number of the filter. Use the **list** command to display a numbered list of configured filters.

Delete

Use the **delete** command to delete a filter-list or filter.

Syntax: delete list ...
filter ...

list *list-name*

Deletes the specified list. The list command can be used to display the configured filter list names.

Example: `delete list example_list`

filter *filter#*

Deletes the specified filter. The list command can be used to display a numbered list of configured filters.

Example: `delete filter 1`

Detach

Use the **detach** command to detach a filter-list from a filter.

Syntax: detach *list-name filter#*

list-name

Specifies the name of the filter-list. The list command can be used to display a list of the configured filter names.

Valid Values: Any alphanumeric string up to 16 characters

Default Value: None

filter#

Specifies the number of the filter. The list command can be used to display a numbered list of configured filters.

Example: detach test_list 1

Disable

Use the **disable** command to disable filtering globally or for a specified filter.

Syntax: disable all
filter ...

all

Disables all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: disable all

filter *filter#*

Disables the specified filter. Use the list command to display a numbered list of configured filters.

Example: disable filter 1

Enable

Use the **enable** command to enable filtering globally or for a specified filter.

Syntax: enable all
filter ...

all

Enables all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: enable all

filter *filter#*

Enables the specified filter. Use the list command to display a numbered list of configured filters

Example: enable filter 1

List

Use the **list** command to globally display the state of the current filtering type, or to display information about a specific filter.

Syntax: list all
filter ...

all

Lists information about the state of all filters of the current type.

Example: list all

```
IPX IPX-List Config>list all
Filtering: ENABLED

Filter Lists:
Name                          Action
----------------------------- ----------
ipx01                         EXCLUDE
ipx02                         INCLUDE
ipx03                         EXCLUDE

Filters:
Id   Default    State      Ifc  Direction  Cache
---- ---------- ---------- ---- ---------- -----
1    INCLUDE    ENABLED    0    INPUT      10
2    INCLUDE    ENABLED    0    OUTPUT     10
3    INCLUDE    DISABLED   1    INPUT      10
4    INCLUDE    DISABLED   1    OUTPUT     10
```

filter *filter#*

Lists information about the specified filter. Use the list command to display a numbered list of configured filters.

Example: list filter 1

```
Filter:
Id   Default    State      Ifc  Direction  Cache
---- ---------- ---------- ---- ---------- -----
1    INCLUDE    ENABLED    0    INPUT      10

Filter Lists:
Name                          Action
----------------------------- ----------
ipx01                         EXCLUDE
ipx02                         INCLUDE
```

Move

Use the **move** command to change the order of filter lists within a filter. Packets are evaluated against the filter lists in the order the lists occur. The first match stops the filtering process.

Syntax: move *src-list-name dst-list-name filter#*

src-list-name

Specifies the list to be moved within the filter.

dst-list-name

Specifies the list before which the src-list-name will be moved.

filter#

Specifies the filter to which the lists belong. The list command can be used to display a list of the configured filters and their attached filter lists.

Example: `move test-list-1 test-list-2 2`

Set-cache

Use the **set-cache** command to set the size of the filter cache. A filter cache is only supported for the IPX interface filter; the ROUTER, RIP and SAP interface filters do not support a cache.

Syntax: set-cache *size filter#*
Example: `set-cache 10 1`

size

Specifies the size of the filter cache (in number of entries).

Valid Values: 4 to 64 cache entries.

Default Value: 10 entries.

filter#

Specifies the number of the filter. The list command can be used to display a numbered list of configured filters.

Example: `set-cache 10 1`

Update

The **update** command accesses the `IPX type-List list-name Config>` prompt. From this prompt you can issue commands to add, delete, or move items within the list being updated. From this prompt you can also set the action for the filter-list being updated.

Syntax: update *list-name*

list-name

Specifies the name of the filter-list. The list command can be used to display the configured filter-list names.

Example: update test-list

Add (Update subcommand)

Use the **add** subcommand to add items to a filter-list. The list item parameters vary based on the type of interface filter (ROUTER, RIP, SAP, or IPX) being configured. For all types of interface filter, the **add** command can be entered without parameters. You will then be prompted for the required parameters.

Add (ROUTER)

Syntax: add *address mask*

address

Specifies the value to be compared against the source node address of the router which sent the RIP response packet (after being ANDed with the mask). If you want to match on a single address, set the address parameter to the address and set the mask to FFFF FFFF FFFF. If you want to match on all addresses, set the address parameter and the mask parameter to 0000 0000 0000.

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

Default Value: 0.

mask

Specifies the value to be ANDed with the source node address of the router which sent the RIP response packet (before being compared with the address parameter).

If you want to match on a single address, set the address parameter to the address and set the mask to FFFF FFFF FFFF. If you want to match on all addresses, set the address parameter and the mask parameter to 0000 0000 0000.

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

Default Value: 0.

Example: `add 400000001000 ffffffff0000`

Add (RIP)

Syntax: add *net-range-start net-range-end*

net-range-start
Specifies the start of a range (inclusive) of IPX network numbers to be filtered. If you want to match on a single network number, set the net-range-start and net-range-end parameters to that network number. If you want to match on all network numbers, set the net-range-start to 0000 0001 and the net-range-end to FFFF FFFE.

Valid Values: X'1' to X'FFFFFFFE'

Default Value: X'1'

net-range-end
Specifies the end of a range (inclusive) of IPX network numbers to be filtered.

Valid Values: X'1' to X'FFFFFFFE'

Default Value: X'1'

Example: `add 00000001 FFFFFFFE`

Add (SAP)

Syntax: add *comparitor hops sap-type name*

comparitor
Specifies the type of hop count comparitor for this list item.

Valid Values:

- <
- ≤
- =
- ≥
- >

Default Value: ≤ The comparitor and hops parameters are ignored on output filters.

hops
Specifies the hop count for this list item. If you do not want to filter based on hop count, enter <= 16 for the comparitor and hop count. The comparitor and hops parameters are ignored on output filters.

Valid Values: 0 to 16

Default Value: 16

sap-type
Specifies the service type to be filtered. Enter the service type, or 0000 for all service types.

Valid Values: X'0' to X' FFFF'

Default Value: None

name
Specifies the service name to be filtered.

Valid Values:

A string of 1 to 48 ASCII characters (X'20' through X'7E'), with the exception of the following special characters: plus (+), minus (-), comma (,), semicolon (;), colon (:), slash (/), and back slash (\).

The question mark (?) and asterisk (*) characters serve as wildcard characters. The question mark may be used multiple times to represent any single character within the server name. The asterisk may be used multiple times to represent any portion of the server name. The question mark and asterisk may also be used together.

Default Value: none

Example: `add < 6 0004 *`

Add (IPX)

Syntax: add *comparitor hops ipx-type dst-net-range-start dst-net-range-end dst-address dst-mask dst-sck-range-start dst-sck-range-end src-net-range-start src-net-range-end src-address src-mask src-sck-range-start src-sck-range-end*

comparitor

Specifies the type of hop count comparitor for this list item. The comparitor and hops parameters are ignored on output filters.

Valid Values:

- <
- ≤
- =
- ≥
- >

Default Value: ≤

hops

Specifies the hop count for this list item. If you do not want to filter based on hop count, enter <= 16 for the comparitor and hop count. The comparitor and hops parameters are ignored on output filters.

ipx-type

Specifies the IPX packet type to be filtered. Enter the packet type, or 00 for all packet types.

Valid Values: X'0' - X'FF'

Default Value: X'0'

dst-net-range-start

Specifies the start of a range (inclusive) of destination IPX network numbers to be filtered. If you want to match on a single network number, set the dst-net-range-start and dst-net-range-end parameters to that network number. If you want to match on all network numbers, set the dst-net-range-start to 0000 0001 and the dst-net-range-end to FFFF FFFE.

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

Default Value: X'0000 0000'

dst-net-range-end
Specifies the end of a range (inclusive) of destination IPX network numbers to be filtered. If you want to match on a single network number, set the dst-net-range-start and dst-net-range-end parameters to that network number. If you want to match on all network numbers, set the dst-net-range-start to 0000 0001 and the dst-net-range-end to FFFF FFFE.

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

Default Value: X'0000 0000'

dst-address
Specifies the value to be compared against the destination node address (after being ANDed with the dst-mask). If you want to match on a single address, set the dst-address parameter to the address and set the dst-mask to FFFF FFFF FFFF. If you want to match on all addresses, set the dst-address parameter and the dst-mask parameter to 0000 0000 0000.

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

Default Value: X'0000 0000'

dst-mask
Specifies the value to be ANDed with the destination node address (before being compared with the dst-address parameter). If you want to match on a single address, set the dst-address parameter to the address and set the dst-mask to FFFF FFFF FFFF. If you want to match on all addresses, set the dst-address parameter and the dst-mask parameter to 0000 0000 0000.

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

Default Value: X'0000 0000'

dst-sck-range-start
Specifies the start of a range (inclusive) of destination IPX sockets to be filtered. If you want to match on a single socket, set the dst-sck-range-start and dst-sck-range-end parameters to that socket. If you want to match on all sockets, set the dst-sck-range-start to 0000 and the dst-sck-range-end to FFFFF.

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

Default Value: X'0000'

dst-sck-range-end
Specifies the end of a range (inclusive) of destination IPX sockets to be filtered. If you want to match on a single socket, set the dst-sck-range-start and dst-sck-range-end parameters to that socket. If you want to match on all sockets, set the dst-sck-range-start to 0000 and the dst-sck-range-end to FFFFF.

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

Default Value: X'0000'

src-net-range-start
Specifies the start of a range (inclusive) of source IPX network numbers to be filtered. If you want to match on a single network number, set the src-net-range-start and src-net-range-end parameters to that network

number. If you want to match on all network numbers, set the src-net-range-start to 0000 0001 and the src-net-range-end to FFFF FFFE.

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

Default Value: X'0000 0000'

src-net-range-end
Specifies the end of a range (inclusive) of source IPX network numbers to be filtered. If you want to match on a single network number, set the src-net-range-start and src-net-range-end parameters to that network number. If you want to match on all network numbers, set the src-net-range-start to 0000 0001 and the src-net-range-end to FFFF FFFE.

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

Default Value: X'0000 0000'

src-address
Specifies the value to be compared against the source node address (after being ANDed with the src-mask). If you want to match on a single address, set the src-address parameter to the address and set the src-mask to FFFF FFFF FFFF. If you want to match on all addresses, set the src-address parameter and the src-mask parameter to 0000 0000 0000.

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

Default Value: X'0000 0000'

src-mask
Specifies the value to be ANDed with the source node address (before being compared with the src-address parameter). If you want to match on a single address, set the src-address parameter to the address and set the src-mask to FFFF FFFF FFFF. If you want to match on all addresses, set the src-address parameter and the src-mask parameter to 0000 0000 0000.

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

Default Value: X'0000 0000'

src-sck-range-start
Specifies the start of a range (inclusive) of source IPX sockets to be filtered. If you want to match on a single socket, set the src-sck-range-start and src-sck-range-end parameters to that socket. If you want to match on all sockets, set the src-sck-range-start to 0000 and the src-sck-range-end to FFFFF.

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

Default Value: X'0000'

src-sck-range-end
Specifies the end of a range (inclusive) of source IPX sockets to be filtered. If you want to match on a single socket, set the src-sck-range-start and src-sck-range-end parameters to that socket. If you want to match on all sockets, set the src-sck-range-start to 0000 and the src-sck-range-end to FFFFF.

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

Default Value: X'0000'

Example:

```
 add  <= 16  0  00000004 00000004 000000000000 000000000000
0000 FFFF 0000005A 0000006A 000000000000 000000000000 0000 FFFF
```

This example filters all packets from IPX networks 5A through 6A to IPX network 4.

Delete (Update subcommand)

Use the **delete** subcommand to delete an item from the current filter-list.

Syntax: delete *item#*

Example: **`delete 4`**

item#

Specifies the number of the item in the list. The number can be obtained by using the list command to list the items in the filter-list.

List (Update subcommand)

Use the **list** subcommand to display the filter-list action and list filter items.

Syntax: list

Example: **`list`**

```
IPX IPX-List 'ipx01' Config>list
Action: EXCLUDE
Id   Hops Type Net Range           Address      Mask         Sock Range
---- ---- ---- ------------------- ------------ ------------ -----------
1    <=16    0     4320 -     4324 4000003A0002 FFFFFFFFFFFF    0 - FFFF (Dest)
                  3A33 -    13A33 400000010000 FFFFFFFF0000    0 - FFFF (Source)
```

Move (Update subcommand)

Use the **move** subcommand change the order of filter items. After you change the order of filter items, they are renumbered to reflect the new order. The list command can be used to display a numbered list of configured filter items.

The *src-line#* parameter indicates the line to be moved. This line will be moved to precede the item specified by the *dest-line#* parameter.

Syntax: move *src-line# dest-line#*

Example: **`move 5 2`**

Set-action (Update subcommand)

Use the **set-action** subcommand to indicate the action to be taken when a match is made to a filter-list.

Syntax: set-action include
exclude

include

Specifies that if a match is found for the current filter, the packet will be processed (included) for ROUTER and IPX filters. For RIP and SAP filters, **include** specifies that the RIP or SAP entry will be processed.

Example: **`set-action include`**

exclude

Specifies that if a match is found for the current filter, the packet will be dropped (excluded) for ROUTER and IPX filters. For RIP and SAP filters, **exclude** specifies that if a match is found, the RIP or SAP entry will be ignored.

Example: `set-action exclude`

Exit

Use the **exit** subcommand to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 21. Monitoring IPX

This chapter describes how to monitor IPX protocol activity and use the IPX console commands. It includes the following sections:

- "Accessing the IPX Console Environment"
- "IPX Console Commands"

Accessing the IPX Console Environment

For information on how to access the IPX console environment, refer to "Getting Started (Introduction to the User Interface)" in the *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

IPX Console Commands

Table 21-1 lists the IPX console commands. The IPX console commands allow you to view the parameters and statistics of the interfaces and networks that transmit IPX packets. Console commands display configuration values for the physical, frame, and packet levels. You also have the option of viewing the values for all three protocol levels at once.

Enter the IPX console commands at the `IPX>` prompt. Table 21-1 summarizes the IPX monitoring commands.

Table 21-1 (Page 1 of 2). IPX Console Command Summary

Command	Function
? (Help)	Lists all the IPX console commands or lists the options associated with specific commands.
Access-controls	Displays whether the global IPX filter (access control) is enabled, the IPX access-control statements, and the number of packets that have matched each access-control statement.
Cache	Lists the current contents of the routing cache.
Config	Lists the IPXWAN router name and node-ID. Lists the number, name and type of each interface on which IPX is enabled, as well as the IPX network number and host number, the frame encapsulation type, whether RIP/SAP Broadcast Pacing is enabled on those interfaces, and the RIP and SAP update intervals on those interfaces.
Counters	Displays the number of routing errors and packet overflows.
Disable	Disables specific IPX interfaces or globally disables IPX.
Dump routing tables	Displays the contents of the IPX RIP table.
Enable	Enables IPX on specific interfaces or globally enables IPX.
Filters	Displays whether global SAP filtering is enabled, the SAP filter statements, and a count of the SAP advertisements which have been filtered.

Table 21-1 (Page 2 of 2). IPX Console Command Summary

Command	Function
Filter-Lists	Accesses the IPX *type*-Lists> prompt. This is the environment where information regarding the Interface filters (ROUTER, RIP, SAP, and IPX) can be displayed.
IPXWAN	Lists IPXWAN configuration information for each serial interface on which IPXWAN is enabled.
Ping	Sends IPXPING packets to another host once a second and watches for a response. This command can be used to isolate trouble in an internetwork environment.
Sizes	Displays the configured sizes of the local node and remote network caches, and the number of cache entries currently in use.
Slist	Displays the contents of the IPX SAP server table.
Exit	Exits the IPX console process and returns to the GWCON environment.

? (Help)

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

Syntax: ?

Example: ?

```
access controls
cache
config
counters
disable
dump routing tables
enable
exit
filters
filter-lists
ipxwan
keepalive
ping
sizes
slist
```

Access Controls

Use the **access-controls** command to list the status of global IPX filters (access controls), the IPX access control statements, and a count of how many times each control statement has been followed.

Syntax: access-controls

Example: **access-controls**

```
Access Control currently enabled
List of Access Control records:
# T   Dest Net  Host          Sck  Sck  Src Net  Host         Sck Sck  Count
1 E         179  123456789ABC 1234 1234      176  000000000000   0   0    3
2 I           0  000000000000    0  FFF        0  000000000000 426 426    0
```

Type — Identifies whether packets are sent or dropped for a specific address or set of addresses. I means include. This allows the packets to be sent. E means exclude. This causes the router to discard the packets.

Dest-net — Network number of the destination. Zero (0) means all networks.

Dest-host — Host number on the destination network (0) means all hosts on the network.

Des-sck — Two numbers that specify an inclusive range of destination sockets.

Src-net — Network number of the source. Zero (0) means all networks.

Src-host — Host number on the source network. Zero means all hosts on the network.

Src-sck — Two numbers that specify an inclusive range of source sockets.

Count — Specifies the number of incoming IPX packets that have matched each access-control statement, causing the associated Type (Include or Exclude) to be performed.

Cache

Use the **cache** command to display the contents of the IPX routing cache.

Syntax: ca**c**he

Example: **cache**

```
Dest Net/Node         Use Count    via Net/Node        via Int
162                    56476       162/000000000000     Eth/0
162/0000C0239F71       56476       162/0000C0239F71     Eth/0
```

The first entry shows that the remote network 152 can be reached over the serial interface with IPX network number 161. The second entry is the IPX network 162. It is an Ethernet directly attached to the router. This entry is a general local network entry. There will be one general local network entry for each of the directly attached networks after they have begun forwarding IPX packets. The last entry is a local entry on an Ethernet. This IPX cache entry has been used to send 56,476 packets to the IPX node number 0000 C023 9F71 on net number 162.

Config

Use the **config** command to list the number, name, and type of each network interface on which IPX is enabled, as well as listing the IPX network number and host number, the frame encapsulation type and the RIP and SAP update intervals on those interfaces. The **config** command also displays whether RIP/SAP Broadcast Pacing is enabled on specific interfaces, and the IPXWAN router name and node-ID.

Syntax: config

Example: **config**

```
Router Configuration
IPX Name:  Node ID: 0
Net   Name   Type                    Network/Address
0     Eth/0  Ethernet/802.3          162/000093908468

IPX Encapsulation/Frame Types
Net   Name   Type                    Encapsulation
0     Eth/0  Ethernet/802.3          ETHERNET_802.3

RIP/SAP Timer Intervals and Pacing
Net  Name   Type                          SAP Interval  RIP Interval Pacing
0    Eth/0  Ethernet/802.3                       1             1      Disabled
```

Router Configuration	The current router configuration information.
IPX Name	The IPXWAN router name.
Node ID	The IPXWAN node-id (primary network number).
Net	The interface number.
Name	The interface name.
Type	The hardware type of the interface.
Network/Address	The user-assigned network number and host number. Except for serial interfaces, the host number is the node address of the network interface. For serial interfaces, it is the user-configured IPX host number.
IPX Encapsulation/Frame Types	The encapsulation type for each interface on which IPX is enabled.
Net	The interface number.
Name	The interface name.
Type	The hardware type of the interface.
Encapsulation	The encapsulation type configured for the interface.
RIP/SAP Timer Intervals and Pacing	The delay between the transmission of complete RIP and SAP advertisements on an interface, and whether RIP/SAP Broadcast Pacing is enabled on an interface.
Net	The interface number.
Name	The interface name.
Type	The hardware type of the interface.
SAP Interval	The number of minutes between complete SAP advertisements on the interface. The range is 1 through 1440. The default is 1.

RIP Interval	The number of minutes between complete RIP advertisements on the interface. The range is from 1 to 1440. The default is 1.
Pacing	Indicates whether RIP/SAP Broadcast Pacing is enabled.

Counters

Use the **counters** command to display the number of routing errors and packet overflows that have occurred. In the example, the counters show no recorded errors.

Syntax: counters

Example: **`counters`**

```
Routing errors
Count     Type
   0      Unknown
   0      Checksum error
   0      Destination unreachable
   0      Hop count expired
   0      Interface size exceeded


Destination errors
Count      Type
   0       Unknown
   0       Checksum error
   0       Nonexistent socket
   0       Congestion


IPX input packet overflows
    Net     Count
   Eth/0    0
```

Routing Errors

Unknown	An unspecified error occurred before reaching the destination.
Checksum	The checksum is incorrect, or the packet had some other serious inconsistency before reaching the destination.
Destination unreachable	The destination host cannot be reached from here.
Hop count expired	The packet has passed through 15 internet routers without reaching its destination.
Interface size exceeded	The packet is too large to be forwarded through some intermediate network.

Destination errors

Unknown	An unspecified error was detected at destination.
Checksum	The checksum is incorrect, or the packet has some other serious inconsistency detected at destination.
Nonexistent socket	The specified socket does not exist at the specified destination host.
Congestion	The destination cannot accept the packet due to resource limitations.

IPX Input Packet Overflows

Net	Specifies the interface name.
Count	Specifies the number of packets that could not be received due to resource limitations.

Delete

Use the **delete** command to remove a Keepalive filtering table entry.

Syntax: delete *entry#*

entry#

Specifies the table entry to be deleted. The **keepalive** command can be used to list the contents of the Keepalive filtering table.

Example: `delete 1`

Disable

Use the **disable** command to disable IPX on specific interfaces, or to disable IPX globally on all interfaces.

Syntax: disable interface ...
ipx

interface *interface#*

Disables IPX on the specified interface. IPX can be re-enabled on the interface using the **enable** command.

Example: `disable interface 0`

ipx

Disables IPX globally on all interfaces. IPX can be globally re-enabled using the **enable** command.

Example: `disable ipx`

Dump

Use the **dump** command to display the contents of the current IPX RIP routing tables.

Syntax: dump

Example: `dump`

The screen displays the following information:

```
11 route entries used out of 32
11 net entries used out of 32

Type  Dest net  Hops  Delay  Age(M:S)   via Router
 Dir      124     0      1       0: 0     124/AA0004001A04
 Dir      131     0      1       0: 0     131/00000000001A
 Dir      177     0      1       0: 0     177/00000000001A
 Dir       41     0      1       0: 0      41/4000C90401FA
 Dir      249     0      1       0: 0     249/0000C9084F34
 RIP      250     1      2       0:10     249/0000C9093250
 RIP 2C39ABE9     2      3       0:10     249/0000C9093250
 RIP       BB     1      2       0:50      41/4000C9050971
 RIP        1     2      3       0:50      41/4000C9050971
 RIP       31     2      3       0:50      41/4000C9050971
 RIP      703     1      2       0:20      41/4000C9041243
```

Type	Specifies one of the following: Dir - specifies that this network is directly connected to the router. RIP - specifies that this route was provided by the IPX routing protocol, RIP. Old - specifies that this route has timed out and is no longer being used. The route remains in the table briefly to inform other routers that the route is no longer valid; after this brief interval, it is no longer displayed.
Dest net	Specifies the destination network number.
Hops	Specifies the number of router hops to this destination.
Delay	Specifies the estimate of how long it takes the router to transmit and for the packet to arrive at its destination. The unit of delay is the number of IBM PC clock ticks to send a 576-byte packet, which is 18.21 clock ticks per second. The minimum delay is 1 unit.
Age	Specifies the age of the routing information in minutes and seconds. If an entry in the routing table is not updated, the router takes the following actions: • After three RIP update intervals have passed, the route is specified as Old and the router advertises that the route is no longer valid. The RIP update interval can be displayed using the IPX **config** command. For additional information on RIP intervals, see “Specifying RIP Update Interval” on page 20-3. • After an additional 60 seconds, the route is deleted and does not appear in the dump display.
Via router	Specifies the next hop for packets going to networks that are not directly connected. For directly connected networks, this is the address of the router interface that transmits the packet.

At the top of the display is the number of route and network entries used and the total available. If all the network entries are used, it is likely that the routing table is not large enough. Use the IPX configuration **set maximum networks** command to increase the size.

If all of the route entries are used, then there may be routes to IPX networks that cannot be kept, including new, incoming networks. If you do not want to increase the number of available routes, reduce the number of maximum routes per network.

Enable

Use the **enable** command to enable IPX on specific interfaces or to globally enable IPX on all interfaces on which IPX has been configured and enabled.

Syntax: enable interface ...
ipx

interface *interface#*

Enables IPX on the specified interface. An IPX network number must have been configured for the interface before IPX can be enabled. For serial interfaces, a host-number must have been configured before IPX can be enabled.

Example: `enable interface 0`

ipx

Enables IPX on all interfaces on which IPX has been configured and enabled.

Example: `enable ipx`

Filters

Use the **filters** command to display whether global SAP filtering is enabled, the SAP filter statements, and a count of the SAP advertisements that have been filtered.

Syntax: filters

Example: `filters`

```
IPX SAP Filter currently enabled
List of IPX SAP Filter records:
Count  Max Hops  Type  Service Name
  0          8     4  ?
  0          1  1234  SomeServer
```

Count	Indicates the number of SAP advertisements that have been filtered (discarded).
Max Hops	Indicates the maximum number of hops permitted for the service.
Type	Is the numeric service class.
Service name	Is the name of the service if it has a name.

Filter-lists

Use the **filter-lists** command to access the `IPX type-Lists>` prompt. Valid types are: router, rip, sap, and ipx.

For information about the commands available from this prompt, see “IPX Interface Filter Monitoring Commands” on page 21-13.

Syntax: filter-lists router
rip
sap
ipx

Example: `filter-lists router`

IPXWAN

Use the **ipxwan** command to list the current configuration information for serial interfaces on which IPXWAN is enabled.

Syntax: ipxwan detailed . . .
summary

detailed *interface#*

Lists the complete current configuration information for the specified serial interface on which IPXWAN is enabled.

Example: `ipxwan detailed`

```
Network number [0]? 2
Detailed information for IPXWAN link over interface 2, PPP/1
This side is the IPXWAN slave
Neighbor Name: SKYSURF2
Neighbor Node ID: 727299
Negotiated Routing Type: RIP/SAP
Link Delay: 330 1/18th sec ticks
Common Net#: 132
Connection Timeouts: 0
Connection Retries: 0
Timer Requests Sent: 1
Timer Requests Received: 1
Timer Responses Sent: 1
Timer Responses Received: 0
Info Requests Sent: 0
Info Requests Received: 1
Info Responses Sent: 1
Info Responses Received: 0
```

Network number	The network interface number.
Neighbor Name	The router name of the neighbor as received in the RIP/SAP Information Request Packet.
Neighbor Node ID	The node ID (also known as the primary network number) of the neighbor. This is a IPX network number unique to the entire internetwork. It is a 32-bit quantity.
Negotiated Routing Type	The negotiated routing type. Currently supported is RIP/SAP. The default is RIP/SAP.
Link Delay	The link delay in 1/18th second ticks calculated by the master. It is a 16-bit quantity. It is always calculated, therefore there is no default.
Common Net#	The network number agreed upon by both ends of the link. This number must be unique to the entire internetwork. It is a 32-bit quantity. There is no default, it must be negotiated.
Connection Timeouts	The number of times the connection timed out. A connection will timeout periodically if the exchange of IPXWAN packets does not proceed. You can configure the timeout period using the **set ipxwan** command. The default for the timeout period is 60 seconds.
Connection Retries	The number of times the connection is retried after timing out. The amount of time to wait (before retrying) is configurable by using the **set ipxwan** command. It defaults to 60 seconds.
Timer Requests Sent	The number of IPXWAN Timer Request packets sent.
Timer Requests Received	The number of IPXWAN Timer Request packets received.
Timer Responses Sent	The number of IPXWAN Timer Response packets sent.
Timer Responses Received	The number of IPXWAN Timer Response packets received.
Info Requests Sent	The number of IPXWAN Information Request packets sent.

Info Requests Received	The number of IPXWAN Information Request packets received.
Info Responses Sent	The number of IPXWAN Information Response packets sent.
Info Responses Received	The number of IPXWAN Information Response packets received.

summary

Lists a summary of the current configuration information for each serial interface on which IPXWAN is enabled. Summary information is displayed only for links that have successfully completed the IPXWAN protocol exchange.

Example: **`ipxwan summary`**

```
Net  Name   Common Net#  NodeID  Neighbor Name

2    PPP/1  132          727299  SKYSURF2
```

Net	Network interface number.
Name	Network interface name.
Common Net#	Network number agreed upon by both ends of the link. This number must be unique to the entire internetwork.
NodeID	Node ID (also known as the primary network number) of the neighbor. This is a IPX network number unique to the entire internetwork.
Neighbor Name	Router name of the neighbor as received in the RIP/SAP Information Request Packet.

Ping

Use the **ping** command to make the router send IPXPING packets to a given destination once a second (“pinging”) and watch for a response. This command can be used to isolate trouble in an internetwork environment.

This process is done continuously. Matching received responses are displayed with the sender's IPX network number and node number, the number of hops, and the round trip time in milliseconds.

To stop the pinging process, type any character at the console. At that time, a summary of packet loss, round trip time, and number of unreachable destinations will be displayed.

When a multicast address is given as destination, there may be multiple responses for each packet sent, one for each group member. Each returned response is displayed with the source address of the responder.

Note: Care should be taken when specifying the broadcast address (FFFF FFFF FFFF) as this could generate a large number of IPXPING response packets, which would degrade network and routing software performance.

Each packet contains 56 bytes of data that include a 4-byte time stamp (to determine the round-trip time) and 52 bytes of sequential numbers, which are checked when receiving IPXPING response packets.

Syntax: ping *ipx-net# ipx-node#*

ipx-net#
: Specifies the destination IPX network number. Valid values are in the range 0000 0001 - FFFF FFFE.

ipx-node#
: Specifies the destination IPX node number. Valid values are in the range 0000 0000 0001 - FFFF FFFF FFFF

Example: `ping 00000004 40000000409A`

```
IPXPING 00000004/40000000409A: 56 data bytes
56 data bytes from 00000004/40000000409A: hops=2, time=220 ms
56 data bytes from 00000004/40000000409A: hops=2, time=200 ms
56 data bytes from 00000004/40000000409A: hops=2, time=210 ms

3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms) min/ave/max = 200/210/220
```

Sizes

Use the **sizes** command to display the configured sizes of the local node and remote network caches, and the number of cache entries currently in use. (This command does not display the contents of the caches.)

Syntax: 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`

```
State Typ Service Name Hops Age(M:S)   Net/   Host     /Sock
 SAP 0004 PCS12         3   0:50         1/000000000048/0451
 SAP 0004 ACMPCS        3   0:50         1/00000000004A/0451
 SAP 0004 DEVEL2        1   0:50        11/0000000000B4/0451
 SAP 0004 PLANNING      2   0:50        BB/0000000000B7/0451
 SAP 0004 DEVEL         2   0:50        BB/0000000000EE/0451
 SAP 0004 SOFT2         1   0:30       704/000000000094/0451
 SAP 0004 SKYSURF1      2   0: 5  2C39ABE9/000000000001/0451
 SAP 0278 DIRTREE       2   0: 5  2C39ABE9/000000000001/4005
 SAP 026B DIRTREE       2   0: 5  2C39ABE9/000000000001/0045

9 services used out of 32
```

State	Specifies one of the following parameters: SAP - indicates that this service was obtained by the SAP routing protocol. Old - indicates that this service has timed out and is no longer being used. The service is kept briefly in the table to inform other routers that the service is no longer valid. After that, it is deleted and is no longer displayed.
Typ	Specifies the server type in hexadecimal. File servers are type 0004. Other type numbers are assigned by Novell.
Service name	Specifies the server's unique name for this type of server. Only the first 30 characters of the 47-character name are displayed to conserve space.
Hops	Specifies the number of router hops from this router to the server.
Age	Specifies the age of the service information. If an entry in the SAP table is not updated, the router takes the following actions: After 3 SAP update intervals have passed, the service is specified as Old and the router advertises that the service is no longer valid. The SAP update interval can be displayed using the IPX **config** command. After an additional 60 seconds, the service is deleted and does not appear in the **slist** display.
Net/Host/Sock	Specifies the address of the service. The address includes the following parameters: Network number Net host number (the address of the first interface on the network) Socket number at which the service can be reached

At the bottom of the display is the number of entries used and the total available. If all the entries are used, it is likely that the service table is not large enough. Use the IPX configuration **set maximum services** command to increase the size.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

IPX Interface Filter Monitoring Commands

Table 21-2 lists the commands available from the IPX *type*-Lists> prompt. Each of these commands is explained in detail in this section.

To access the IPX *type*-Lists> prompt, enter **filter-lists** *type* at the IPX> prompt. Valid types are router, rip, sap, and ipx.

Table 21-2. IPX Interface Filter Command Summary

Command	Function
Cache	Displays the contents of the filter cache for the specified interface. Only the IPX filter supports a filter cache.
Clear	Clears the counters of the specified filter, or clears the counters of all filters of the current type (ROUTER, RIP, SAP, or IPX).
Disable	Disables a specified filter, or all filters of the current type.
Enable	Enables a specified filter, or all filters of the current type.
List	Lists a specified filter, or all filters of the current type.
Exit	Returns to the previous prompt level.

Cache

Use the **cache** command to display the contents of the filter cache. Only the IPX filter supports a cache. ROUTER, RIP, and SAP filters do not support a filter cache.

Syntax: cache filter *filter#*

filter#

Specifies the number of the filter. The list command can be used to display a numbered list of configured filters.

Example: **cache filter 1**

```
IPX IPX-Lists>cache filter 1
Hops Type Dst Net  Address      Sock Src Net  Address      Sock Action
---- ---- -------- ------------ ---- -------- ------------ ---- -------
   4   00 04000000 400003900000  802 03000040 400003004400  966 EXCLUDE
   2   00 0004A300 400000233D00  952 0763A020 4000000DD100  920 INCLUDE
```

Clear

Use the **clear** command to clear the counters of the specified filter, or to clear the counters of all filters of the current type (ROUTER, RIP, SAP, or IPX).

Syntax: clear all
filter ...

all

Clears the counters of all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: **clear all**

filter *filter#*

Clears the counters of the specified filter number. The list command can be used to display a numbered list of configured filters.

Example: `clear filter 1`

Disable

Use the **disable** command to disable specific filters or to disable all filters of the current type (ROUTER, RIP, SAP, or IPX).

Syntax: disable all
filter *filter#*

all

Disables all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: `disable all`

filter *filter#*

Disables the specified filter number. The list command can be used to display a numbered list of configured filters.

Example: `disable filter 1`

Enable

Use the **enable** command to enable specific filters or to enable all filters of the current type (ROUTER, RIP, SAP, or IPX).

Syntax: enable all
filter *filter#*

all

Enables all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: `enable all`

filter *filter#*

Enables the specified filter number. The list command can be used to display a numbered list of configured filters.

Example: `enable filter 1`

List

Use the **list** command to display information about specific filters, or about all filters of the current type (ROUTER, RIP, SAP, or IPX).

Syntax: list all
filter *filter#*

all

Lists the configuration of all filters of the current type (ROUTER, RIP, SAP, or IPX).

Example: `list all`

```
IPX IPX-Lists>list all
Filtering: ENABLED

Filter Lists:
Name                           Action
------------------------------ ----------
ipx01                          EXCLUDE
ipx02                          INCLUDE
ipx03                          EXCLUDE

Filters:
Id   Default    State      Ifc  Direction  Cache
---- ---------- ---------- ---- ---------- -----
1    INCLUDE    ENABLED    0    INPUT      10
2    INCLUDE    ENABLED    0    OUTPUT     10
3    INCLUDE    DISABLED   1    INPUT      10
4    INCLUDE    DISABLED   1    OUTPUT     10
```

filter *filter#*

Lists the configuration of the specified filter number. The list command can be used to display a numbered list of configured filters.

Example: `list filter 1`

```
IPX IPX-Lists>list filter 1

Filters:
Id   Default    State      Ifc  Direction  Cache
---- ---------- ---------- ---- ---------- -----
1    INCLUDE    ENABLED    0    INPUT      10

Filter Lists:
Name                           Action     Count
------------------------------ ---------- ----------
ipx01                          EXCLUDE    43
ipx02                          INCLUDE    23453
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 22. Using and Configuring ARP

This chapter describes how to use the Address Resolution Protocol (ARP), Inverse Address Resolution Protocol (Inverse ARP), and ARP Over ATM 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 physical medium access control (MAC) addresses. Given only the network layer address of the destination system, ARP locates the MAC address of the destination host within the same network segment.

For example, a router receives an IP packet destined for a host connected to one of its LANs. The packet contains only a 32-bit IP destination address. To construct the data link layer header, a router acquires the physical MAC address of the destination host. Then, the router maps that address to the 32-bit IP address. This function is called *address resolution*. Figure 22-1 on page 22-2 illustrates how ARP works.

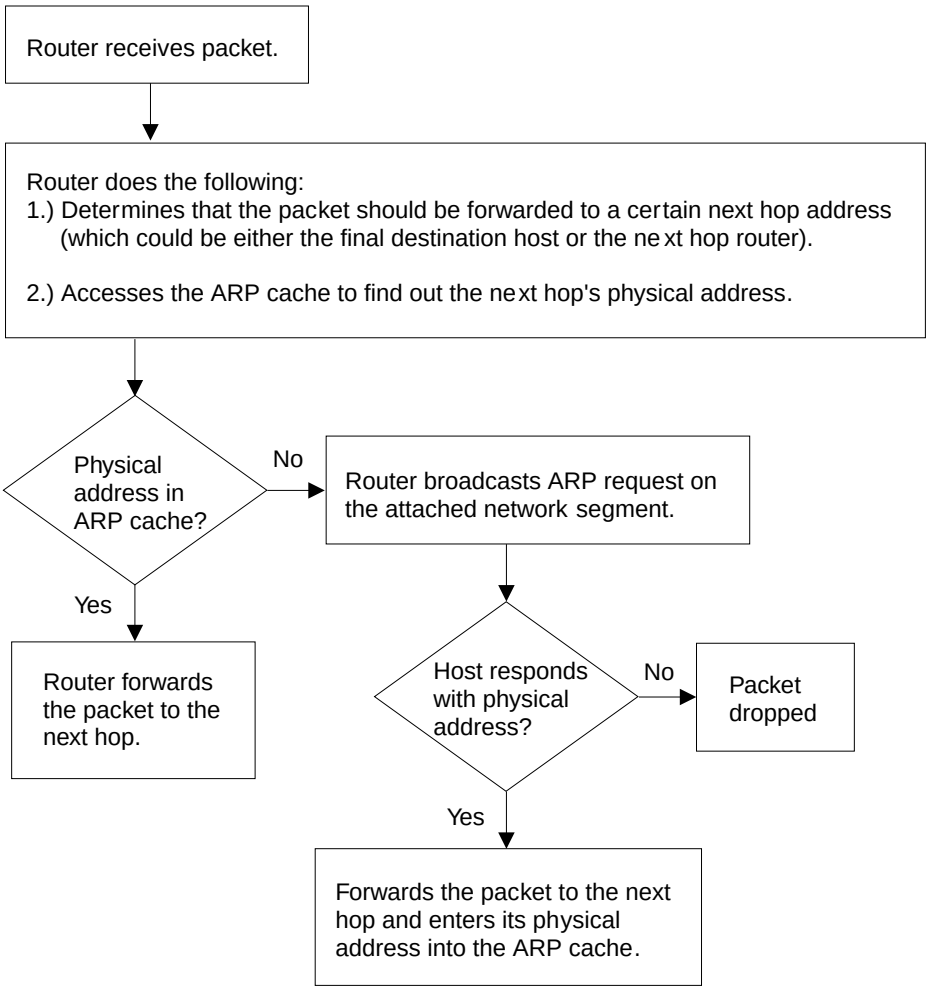


Figure 22-1. ARP Address Resolution Broadcast

When a router translates a network layer address to a physical address, the router accesses the ARP (translation) cache. The ARP cache contains the physical MAC address that corresponds to that network layer address. If the address is missing, the router broadcasts an ARP request to all hosts on the attached network segment to locate the correct physical MAC address. The node with the correct physical MAC address responds to the router. The router then sends the packet to the node and enters the physical MAC address into the translation cache for future use.

RFC 1577, Classical IP & ARP over ATM, extends the ARP protocol with a different packet format and with the addition of an entity known as the ARP server as described in “Classical IP and ARP Over ATM Overview (RFC 1577)” on page 22-4.

Inverse ARP Overview

Inverse ARP, described in RFC 1293, was created for Frame Relay networks. This protocol defines a method for routers on a Frame Relay network to learn the protocol addresses of other routers in a way that very efficiently reduces traffic by eliminating the need to use broadcast ARP packets for address resolution. Inverse ARP discovers a protocol address by sending Inverse ARP request packets to the hardware address (for Frame Relay circuits the circuit identifier is the Frame Relay equivalent of a hardware address), address; for ATM, an ATM address is exchanged), as soon as the circuit becomes active. The remote router responds with its protocol address and the resulting mapping is stored in the ARP cache.

In ATM, the inverse ARP packet has been extended to handle the variable-sized ATM addresses of the source and destination. Addresses learned by inverse ARP are aged out in the same way as those learned by ARP.

The protocol address-to-hardware address entries learned by Inverse ARP do not time out when the ARP refresh timer expires. The mappings do not age at all except when the Frame Relay circuit goes down. This means that the router does not need to transmit any ARP broadcasts to update the ARP cache. However, the router permits updates to an entry when the other (remote) router changes its protocol address.

Support for both ARP and Inverse ARP greatly enhances the router's interoperability with other vendors' routers over Frame Relay for dynamic mapping of protocol and hardware addresses. If other Frame Relay-attached routers support Inverse ARP, then the mappings are dynamically learned as described above. If the attached routers do not support Inverse ARP but support "traditional" ARP on Frame Relay, then the mappings still could be learned dynamically using ARP exchanges (see Figure 22-1 on page 22-2).

If needed, you can manually configure the protocol addresses of other routers using the Frame Relay configuration command **add protocol-address**. For additional information, see the chapter on configuring Frame Relay interfaces in the Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1.

Classical IP and ARP Over ATM Overview (RFC 1577)

The Internet Engineering Task Force (IETF) has standardized its solution for sending IP traffic over an ATM interface in RFC 1577, "Classical IP & ARP over ATM." This document, created by the IP over ATM working group of the IETF, strives to keep the ATM infrastructure transparent to IP. Most applications that run today in a LAN or WAN environment will see no difference in function; however, their performance and throughput gains may be substantial, as described in "Advantages of Classical IP."

For additional information on Classical IP & ARP over ATM, and for illustrations showing logical and physical network configurations, refer to *Multiprotocol Switched Services (MSS) Server Configuration and Operations 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 *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

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 *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

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 *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

  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 MSS Server 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 MSS Server allows you to place that equipment where it is most effective.

Classical IP Components

The Logical IP Subnet contains all of the properties of a normal IP subnet whether it is Ethernet, Token-Ring, or Frame Relay. However, because ATM is a Non-Broadcast Multiple Access (NBMA) network, the existing broadcast method for resolving addresses cannot be performed. To solve the addressing problem, RFC 1577 describes a registration/request procedure and introduces the notion of an ARP Server and ARP clients.

One ARP Server is defined per LIS. The server maintains the translation of IP addresses to ATM addresses. It allows CIP Clients to register by receiving incoming VCCs and querying the client for the appropriate information. The ARP Server also responds to ATMARP requests for ATM addresses corresponding to IP addresses requested by the client. Finally, the ARP server manages and updates its tables through aging ARP entries and managing incoming VCCs.

The client is the entity that always places calls. A client, as it IMLs, will place a call to the ARP Server, and through the exchange of InATMARP requests and replies will register with the ARP Server. The call placed by the client to the server is called a control channel. When the client has traffic to transmit to another client on the LIS, the client sends an ARP request to the ARP Server with the target IP address. The server sends back either a reply (if the server has the information in its table) or a NAK (if no information is available). The client uses this ATM address to place a call to the target client (this call is referred to as a data channel). Once the call is established, IP datagrams may traverse the link at any time.

Within the CIP model, there are two forms of request/replies: ATM ARP request/replies (referred to as ARPs), and InATMARP request/replies. One could consider InATMARPs as gathering first-hand information. That is, InATMARP is used to query the other end of a VCC for its IP address and ATM address. InATMARP also informs the other end who it is (its IP address and ATM address). ATMARP could be considered surrogate information. A CIP client sends an ATMARP to the ARP Server to find the ATM address corresponding to the specified IP address. The server replies with the requested information, or with a NAK if the information is not available. However, the RFC requires all clients and servers to respond to ARPs and InATMARPs with the appropriate response.

A user may configure up to 32 LISs per ATM interface. For each LIS, the device can appear as a client only, or can appear as both a client and an ARP Server on that LIS. The device does not support an ARP Server only as this goes against the recommendation of RFC 1577 that each ARP Server should contain an IP address.

Note: Using ATM Virtual Interfaces allows more than 32 IP addresses per physical ATM interface since several ATM Virtual Interfaces can be configured on each physical ATM Interface and each AVI can be configured with up to 32 IP addresses.

Refer to the *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1* 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).

IP Addresses and CIP Components

IP addresses are key to IP routing. When configuring the device, the act of adding an IP address to an interface (ATM port), automatically creates a CIP client. The client is defined further by adding ATM ARP client information, but it is the adding of the IP address that creates the client.

Each server, since it contains an IP address, implicitly contains a client as well. When configuring the server, you must configure an IP address. Again, this automatically creates a client. The server is not created until you customize the client configuration. In particular, you must specify that this client is also the server for the LIS. The required databases are then created and maintained to service incoming requests.

The IP address configured does not necessarily imply that the device will act as a router. To act as a router, a higher level routing protocol such as OSPF must be configured. However, if the device is attached to multiple subnets, and if packets are sent to it from one subnet destined to a station on one of the other attached subnets, the device will forward that packet without having any routing protocol configured. Further, if a packet is sent to the device, but the destination of the packet is not the device, and the destination is on the same subnet as the source, the device will send an ICMP redirect message to the originator, and will forward the packet to the proper host.

Because CIP treats ATM as an NBMA network, there is no notion of broadcast. RIP, which is a routing protocol that assumes broadcast will not function in this environment. OSPF, which adapts itself to several types of networks, treats ATM as either a point-to-multipoint network or a Non-Broadcast Multi-Access (NBMA) network which it manages quite well.

ATM Addresses of CIP Components

Each client receives a unique ATM address. As described earlier, only NSAP addresses are supported. The End System Identifier (ESI) and the Selector can be chosen by the person configuring or it may be generated automatically during initialization time. If a device is configured as a client-only on a LIS, then configuring the ESI or Selector is not required (it is recommended that automatic generation be used). If a device is configured as a client/server pair, then it is strongly recommended that you do specify your own Selector, and if necessary, the ESI. (Note that the ESI will default to a burned-in 6-byte value that is unique.) A user will want to specify this information so that the specific ATM address comes up every time for that Server. Clients wishing to connect to this server can rely on the fact that the ATM address of the Server will not change.

If a server/client pair is configured for a specific LIS, then both the server and the client will use the same ATM address. The ATM addresses for each CIP client should be unique.

Virtual Channel Connection (VCC)

A Virtual Channel Connection (VCC) is the lowest common denominator for data transmission. It can either be dynamically created in which case a VCC is a Switched Virtual Circuit (SVC), or it may be configured in the ATM Switch and end stations as a Permanent Virtual Circuit (PVC).

SVCs require a call setup or signalling protocol to establish the connection. Setting up an SVC is similar to placing a phone call. The user dials a phone number and waits for the phone to be answered before communicating to the answering party. If either end hangs up the phone, then the caller must redial the number before talking again. The same is true for ATM SVCs. The host sends out a setup message with a 20-byte ATM address (similar to a phone number), and waits for the other end to connect. Either host can hang-up the channel.

PVCs, on the other hand require no signalling protocol. Nor do they require matching levels of UNI. They are static, and are available to the host from initialization time until power down. The host does not need to take any actions

to “set up” the connection. As such, PVCs are simpler and generally more reliable than SVCs.

The device's implementation of Classical IP supports both PVCs and SVCs. SVCs may be generated automatically through the address resolution process and subsequent call setup performed by the Classical IP code, or an SVC may be explicitly configured by the user. Automatic SVCs are brought up and torn down by the ARP subsystem as required for sending IP traffic. A configured SVC is brought up during initialization, and is kept up indefinitely. If the configured SVC does not connect, the device continues to retry the connection periodically until power is turned off.

PVCs and configured SVCs require no ARP Server definition. That is, a LIS could consist of hosts that were interconnected only by configured information. Optionally, the destination IP address of a configured PVC or SVC can be configured as well. If the IP address is not configured, InATMARP packets are used to determine what IP address sits at the opposite end of a VCC. For a network of any size, the amount of manual configuration would become prohibitive. Automatically generated SVCs drastically reduce the amount of configured information, and provide maximum flexibility for adding and moving hosts.

Automatically generated VCCs can only exist with the assistance of an ARP Server. Each client must be configured with the ARP Server's ATM address. Immediately after initialization, the client will attempt to connect to the server. This connection is referred to as a control channel. The principal use of a control channel is for sending ATMARP and InATMARP requests and replies, although if the ARP Server is also a client, the control channel also can be used for sending IP data. Automatic VCCs generated to send data from one host to another are referred to as data channels.

The attributes of both control and data channels can be tailored to the user's needs. The CIP configuration of the device allows for configuration of the Peak Cell Rate, Sustained Cell Rate, maximum SDU sizes and other characteristics of the control and data channels set up by the device. A user also can choose to limit the cell rates of incoming calls to avoid the problems caused by mismatches in bandwidths of the various ATM attachments.

Key Configuration Parameters for Classical IP

The simplicity of CIP is that very few configuration parameters are required. For a client-only, two pieces of information are required:

1. The IP address and Subnet mask. (add address)
2. The ATM address of the ARP Server. (add arp-server)

The IP address and subnet mask are required to give the client its unique IP identity so that it can send and receive IP datagrams. It also defines the subnet to which this CIP client belongs. The ATM address of the ARP Server is used by the client during initialization to establish a control channel with the ARP Server.

The configuration of the server is similarly simple. Essentially, the server needs to be defined with a fixed, well-known ATM address, and it needs to know which LIS it is serving. The server configuration requires the following:

1. The IP address and Subnet mask. (add address)
2. Answering “Yes” to the question about whether this client is also a server. (add atm-client-configuration)
3. Specifying an explicit selector for the server's ATM address (answering “no” when asked if you wish to use the internally assigned selector). (add atm-client-configuration)

The IP address and Subnet mask tell the server which LIS it is serving. The IP address also gives IP access to the server and routing function if desired (through the implicit client). Questions 2 and 3 are asked, among others, in the “add atm-client-configuration” Question 2 is required to enable the server function for that LIS. Question 3 is used to give the server a predictable ATM address.

How to Enter Addresses

Addresses are entered in two ways, depending on whether the address represents (1) an IP address, or (2) an ATM address, MAC address, or route descriptor, as follows:

1. IP address

   IP addresses are entered in dotted decimal format, a four-byte field represented by four decimal numbers (0 to 255) separated by periods (.).

   Example of IP Address:

   ```
   01.255.01.00
   ```

2. ATM or MAC address or route descriptor

   ATM addresses, MAC addresses, and route descriptors are entered as strings of hexadecimal characters with or without optional separator characters between bytes. Valid separator characters are dashes (–), periods (.), or colons (:).

   Examples of ATM address, MAC address or route descriptor:

   ```
   A1FF010203
         or
   A1–FF–01–02–03
         or
   A1.FF.01.02.03
         or
   39.84.0F.00.00.00.00.00.00.00.00.00.03.10.00.5A.00.DE.AD.C8
         or
   A1:FF:01:02:03
         or even
   A1–FF.01:0203
   ```

   This applies to addresses entered for ATM, LAN emulation, and Classical IP & ARP over ATM.

IPX and ARP Over ATM Overview (RFC 1483)

The MSS Server uses LLC/SNAP encapsulation as specified by RFC 1483 to carry IPX traffic over ATM. MSS Servers (and other routers that support RFC 1483 LLC/SNAP encapsulation on ATM) can be interconnected in full or partial meshes via manually-configured RFC 1483 connections. Both PVCs and configured SVCs are supported. However, VCCs to IPX routers must be dedicated to IPX; they cannot be shared with other protocols, such as IP. 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 may be configured on a single ATM interface.

The MSS Server 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 MSS Server 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 MSS Server) 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 MSS Server'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 *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1* for additional information about ATM Virtual Interfaces.

Note: Although the combination of RFC 1483 encapsulation and InATMARP has not been has not yet been standardized, the combination is specified for IPX over Frame Relay in RFC 1490 [9].

Bridging over ATM Overview (RFC 1483)

While 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 refer to "RFC 1483 Support for Bridging" on page 5-5.

Classical IP Redundancy Overview

The ARP server redundancy has two MSSs that are acting as back-up for each other and performing routing from one subnet to another. This feature allows you to specify in your configuration which MSS will act as the primary server, and which MSS 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. The backup is configured to act as a client on this LIS. When the primary fails, the backup de-registers its ATM address and re-registers using the primary's ATM address and takes over as the ARP Server. It also acts as the redundancy 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 2 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).

If both MSS servers are operational, the one specified as primary will service incoming calls. The one specified as secondary will be inactive as an ARP server but operating as an ARP Client on the LIS with the primary as its ARP server until it detects that the primary server has failed. When the primary server fails, the secondary server will service incoming calls until the primary server is active again. When the primary is active again, the secondary relinquishes the ARP server responsibility (deregisters the primary's ATM address) but becomes active as a client (re-registers with its ATM address) on the LIS and the primary server services the incoming calls again.

ARP Server redundancy configuration will give you the capability to control which MSS acts as primary, and which one acts as the secondary. This allows you to effectively load balance your ARP Servers while providing backup. For example, you may want an MSS to be the primary ARP Server for 6 LISs and to be the secondary for 6 other LISs. And you may want a second MSS to be the secondary for the first 6 LISs and the primary for the other 6 LISs. The resulting configuration will have 12 LISs, 6 being served by one MSS, and 6

being served by the other. If either MSS goes down, the other MSS 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, 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 only used for redundancy activity. 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.

Note: The ARP server redundancy requires that clients on the LIS be able to associate more than one IP address with a single VCC.

ARP Server redundancy is configured by:

1. Configuring an APR Client/Server pair on one MSS Server and designating this as the primary ARP Server
2. Configuring an APR Client on the other MSS Server and designating that it should provide the backup ARP Server function, and
3. Using 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 for more detail.

Accessing the ARP Configuration Environment

For information on how to access the ARP configuration environment, see "Getting Started (Introduction to the User Interface)" in the Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1.

Use the following procedure to access the ARP *configuration* process.

1. At the OPCON prompt, enter **talk 6**. (For more detail on this command, refer to *The OPCON Process and Commands* in the Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1.) For example:

   ```
   * talk 6
   Config>
   ```

   After you enter the **talk 6** command, the CONFIG prompt (`Config>`) displays on the console. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the CONFIG prompt, enter the **prot arp** command to get to the `ARP Config>` prompt.

ARP and Inverse ARP Configuration Commands

This section summarizes and then explains all the ARP configuration commands. Table 22-1 lists the ARP configuration commands. You can access ARP configuration commands at the `ARP config>` prompt.

Note: These commands are used to manage the ARP table for emulated LANs. They have no effect on the Classical IP ARP table that is associated with the ATM physical interface.

Table 22-1. ARP Configuration Commands Summary

Command	Function
? (Help)	List the ARP configuration commands or list the options associated with specific commands.
Add Entry	Add a MAC address translation entry.
Change Entry	Change a MAC address translation entry.
Delete Entry	Deletes a MAC address translation entry.
Disable Auto-refresh	Disable ARP auto-refresh.
Enable Auto-refresh	Enable ARP auto-refresh.
List	List ARP configuration data in SRAM.
Set	Set the usage and refreshes timeout values.
Exit	Exit the ARP configuration process.

? (Help)

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

Syntax: ?

Example: **?**

Example:

```
ARP Config> ?
LIST
ADD
CHANGE
DELETE
DISABLE
ENABLE
SET
EXIT
```

or

`set ?`

Add Entry

Use the **add entry** command to add a “static protocol-to-hardware address mapping” entry. This command is currently supported for IP addresses only.

Syntax: add entry *ifc# prot-type prot-addr MAC-addr*

ifc# **Valid values**: any defined interface

Default value: 0

prot-type

Valid values: any protocol that ARP supports.

Default value: IP

prot-addr

Valid Values: any valid IP address

Default Value: 0

MAC-addr

Valid Values: any valid MAC address

Default Value: none

Example: **`add entry`**

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
Mac Address []?
```

Change Entry

Use the **change entry** command to change a “static protocol-to-hardware address mapping” entry. This command is currently supported for IP addresses only. The hardware address parameter (MAC-addr) should be the address of the node being changed.

ifc# **Valid values**: any defined interface

Default value: 0

prot-type

Valid values: any protocol that ARP supports.

Default value: IP

prot-addr

Valid Values: any valid IP mask

Default Value: none

MAC-addr

Valid Values: any valid MAC address

Default Value: none

Syntax: change entry *ifc# prot-type prot-addr MAC-addr*

Example: **`change entry`**

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
Mac Address []?
```

Delete Entry

Use the **delete entry** command to delete a “static protocol-to-hardware address mapping” entry. This command is currently supported for IP addresses only.

ifc# **Valid values**: any defined interface

Default value: 0

prot-type

Valid values: *IP* or *IPX*

Default value: IP

prot-addr

Valid Values: any valid IP address

Default Value: 0.0.0.0

Syntax: delete entry *ifc# prot-type prot-addr*

Example: **`delete entry`**

```
Interface Number [0]?
Protocol [IP]?
IP Address [0.0.0.0]?
```

Disable Auto-Refresh

Use the **disable auto-refresh** command to disable the auto-refresh function. The auto-refresh function is the router’s capability to send an ARP request based on the entry in the translation cache before the refresh timer expires. The request is sent directly to the hardware address in the current translation instead of a broadcast. If auto-refresh is disabled, no ‘preemptive’ ARP request is made, the refresh timer is allowed to expire, and the ARP translation is purged from the table. The next protocol packet to the destination protocol address will then cause a new ARP request to be broadcast on the network.

Syntax: disable auto-refresh

Example: **`disable auto-refresh`**

Enable Auto-Refresh

Use the **enable auto-refresh** command to enable the auto-refresh function. The auto-refresh function is the router’s capability to send an ARP request based on the entry in the translation cache before the refresh timer expires. The request is sent directly to the hardware address in the current translation instead of a broadcast.

Enabling auto-refresh could cause entries to be retained in the cache regardless of their usage. On networks with a large number of nodes, this can lead to an excessive number of entries in the cache, which might adversely affect router performance. However, on networks with a small number of nodes, this option is useful in reducing broadcast ARP traffic.

Syntax: enable auto-refresh

Example: **`enable auto-refresh`**

List

Use the **list** command to display the contents of the router's ARP configuration as stored in SRAM. The list command displays the current timeout settings for the refresh and usage timer.

Syntax: list all
config
entry

all

Lists the ARP configuration followed by all of the ARP entries.

Example: `list all`

```
ARP configuration:

Refresh Timeout: 5 minutes
Auto Refresh: disabled

Mac address translation configuration
IF #           Prot #         Protocol --> Mac Address
 0               0           2.2.2.1 --> 0000C90932EF
```

config

Lists the configuration. for the different ARP parameters.

Example: `list config`

```
ARP configuration:

Refresh Timeout: 5 minutes
Auto refresh: disabled
```

entry

Lists the ARP entries in SRAM.

Example: `list entry`

```
Mac address translation configuration

IF #           Prot #         Protocol --> Mac Address
 0               0           2.2.2.1 --> 0000C90932EF
```

Set

Use the **set** command to set an ARP configuration parameter.

Syntax: set refresh-timer

refresh-timer *minutes*

Changes the timeout value for the refresh timer. To change the timeout value for the refresh timer, enter the timeout value in minutes. A setting of zero (0) turns off (disables) the refresh timer.

This timer is used in determining when an ARP translation cache entry is to be refreshed while auto-refresh is enabled, or purged while auto-refresh is disabled. Disabling the timer causes entries to be retained until a newly learned address translation causes entries to be removed, until entries are cleared manually with the ARP **clear** console command, or until the router is restarted.

Valid Values: an integer number of minutes in the range of 0 - 65535

Default Value: 5 minutes

Example: `set refresh-timer 3`

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

ARP Over ATM Configuration Commands

This section summarizes and explains the ARP Over ATM configuration commands. These commands apply to:

- Classical IP & ARP over ATM
- IPX over ATM
- Bridging over ATM

Enter the commands at the `ARP Config>` prompt.

Differences for IP, IPX and Bridging

Configuring IPX over ATM (using RFC 1483) is similar to configuring Classical IP (RFC 1577).

Once you enter "IPX" as the protocol, some subsequent questions are different than those for protocol "IP." Since IPX over ATM does not use ARP servers, questions relating to ARP servers are not asked.

Also, IPX over ATM requires fewer parameters to be configured than Classical IP. The IPX network number and the IPX host number (IPX ATM-ARP-client) are the only required parameters for IPX over ATM. If you need to open a connection to a remote IPX router, you must additionally configure the desired channels (VCCs).

For bridging over ATM, no configuration is required here. However, channel and client records may be displayed here. Also, since default values are used for traffic parameters when creating the ATM client records, the user may wish to modify those here.

Effect on ARP Table Entries

These commands apply only to the physical ATM interface where the ARP entries reside for ARP over ATM. These commands will have no effect on a non-ATM interface such as an Emulated LAN.

Table 22-2. ARP Over ATM Configuration Command Summary

Command	Function
? (Help)	Lists all of the ARP over ATM configuration commands, or lists the options associated with specific commands. ?
Add	Adds an arp-server, atm-arp-client-configuration, pvc-atm-arp-entry, 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.
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 and Redundancy.
Exit	Exits the ARP over ATM configuration process and returns to the `Config>` prompt.

? (Help)

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

Syntax: list output

Example: `ARP Config>` **?**list output

Add

Use the **add** command to add an arp-server, atm-arp-client-configuration, 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

Adds an arp-server to the client specified. Only one ARP Server is allowed per client. During initialization, the specified Classical IP client will place a call to the ARP server, and will use it as the mechanism for resolving IP addresses to ATM addresses. If a CIP client is configured to also be a server, then this command will override the client configuration and the client goes to the remote ARP Server to resolve all addresses.

local client IP address

Valid Values: any valid IP mask

Default Value: none

The *private-nsapa* field is the Private Network Specified Access Point Address that is the addressing format specified in the UNI versions 3.0 and 3.1. The first byte of the *nsapa* defines the addressing format, as follows:

First Byte	*NSAP Address Format Specification*
0x39	DCC ATM Format
0x47	ICD ATM Format
0x45	E.164 ATM Format

Note: This setting corresponds to a client's (IP address/port number) pair.

Default value: none

Example:

```
ARP config> add arp-server private-nsapa
Local Client IP Address [0.0.0.0]? 2.2.3.100
Private NSAP Address: Specify 40 digits
ATM Address []? 39840f000000000000000000000410005a3345f3a0
```

Local Client IP Address Address of this client.

ATM NSAP Address Address of the Remote ARP Server

atm-arp-client-configuration

Adds atm-arp-client-configuration.

You will be prompted to provide information about the characteristics of the VCCs that will be set up and received by this client, the refresh timeout and auto-refresh settings, how the ATM address for this client is determined, and the frame size that this client can handle.

Note: Any bandwidth or cell parameter that equals zero will be treated as the line speed of the ATM interface.

Example for IP:

```
ARP config> add atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]?
Client IP Address [0.0.0.0]? 2.2.3.100
This client is also a server? [Yes]: no
Refresh timeout (in minutes) [5]?
Enable auto-refresh? [No]:
Refresh by InAtmArp? [Yes]:
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [Yes]:
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Service for Control VCCs? [Yes]:
Peak Cell Rate of outbound control VCCs (Kbps) [0]?
Sustained Cell Rate of outbound control VCCs (Kbps) [0]?
Use Best Effort Service for Data VCCs? [Yes]:
Peak Cell Rate of outbound Data VCCs (Kbps) [0]?
Sustained Cell Rate of outbound Data VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
```

Interface Number — Interface number assigned.

Valid values: any interface on the device

Default value: 0

Protocol — **Valid values**: *IP*, *IPX*, or *ASRT*

Default value: IP

Client IP Address — Client IP Address (IP only)

Valid Values: any valid IP address

Default value: 0.0.0.0

This client is also a server — Yes or No. If no, client is not a server. (IP only)

Refresh timeout (in minutes) — Refresh timeout value in minutes.

Valid Values: an integer number of minutes in the range of 0 - 65535

Default Value: 5 minutes

Enable auto-refresh — YES or NO.

Refresh by InAtmArp — YES or NO. If YES, and if auto-refresh is enabled, then InAtmArp requests will be periodically transmitted to confirm the existence of the remote host. If NO, then AtmArp

requests will be transmitted to the ARP Server to reconfirm the ARP entry.

Use burned-in ESI as part of the ATM address. You might be given other choices depending on your configuration.

Select ESI
Specify the ESI index number from the the number of any defined ESI

Default Value: 1 which specifies to used the burned in address “pick list” (the list of configured ESIs that you previously configured using the **add esi** network configuration command). The default item in the pick list is to use the ATM interface's burned in ESI.

Use internally assigned selector
Use internally assigned selector.

Valid Values: any single octet value that has not been previously used and is within the range defined for the device.

Default Value: none

Validate PCR for best effort VCCs
TRUE or FALSE. When true, Best-Effort VCCs will be rejected if the signaled forward PCR exceeds the Maximum Reserved Bandwidth or the speed of the adapter. If false, Best-Effort PCRs will be rejected without regard to the signaled Peak Cell Rate.

Maximum Reserved Bandwidth for incoming VCCs (Kbps)
Defines the maximum acceptable Sustained Cell Rate (SCR) for an incoming VCC. If SCR is not specified on the incoming call, then this parameter defines the maximum acceptable Peak Cell Rate (PCR). Calls received with traffic parameters specifying higher rates will be released. This parameter is applied to both forward and backward Cell Rate parameters. The constraint imposed by this parameter is applicable to best effort connections (if “validate PCR” is yes) and is compared to the PCR on the incoming call.

Valid Values: any single octet value that has not been previously used and is within the range defined for the device.

Default Value: none

Use Best Effort Service for Control VCCs
Specifies the type of traffic characteristics to be associated with Control VCCs. Bandwidth is not reserved for best effort traffic. **Valid Values**: *Best Effort* or *Reserved Bandwidth*

Default Value: Best Effort

Peak Cell Rate of outbound control VCCs (Kbps)
Specifies the Peak Cell Rate (PCR) traffic parameter for the Control VCC. This PCR value is used for both the forward and backward PCR values of both best effort and reserved bandwidth VCCs.

Valid Values: an integer Kbps in the range of 0 - line speed of the ATM device

Default Value: 0

Sustained Cell Rate of outbound control VCCs (Kbps)
Specifies the bandwidth reserved by all VCCs on a given ATM device. (Sustained Cell Rate can be considered to reserved bandwidth.) This parameter is applicable only when Best Effort Service is not selected for Control VCCs.

Valid Values: an integer Kbps in the range of 0 - control VCC PCR

Default Value: 0

Use Best Effort Service for Data VCCs
Yes or No. Specifies the type of traffic characteristics to be associated with Data VCCs. Bandwidth is not reserved for best effort traffic.

Peak Cell Rate of outbound Data VCCs (Kbps)
Specifies the Peak Cell Rate (PCR) traffic parameter for the Data VCC. This PCR value is used for both the forward and backward PCR values of both best effort and reserved bandwidth VCCs.

Valid Values: an integer Kbps in the range of 0 - control VCC PCR

Default Value: 0

Sustained Cell Rate of outbound Data VCCs (Kbps)
Specifies the Sustained Cell Rate (SCR) traffic parameter for the Data VCC. (Sustained Cell Rate can be considered to reserved bandwidth.) This parameter is applicable only when Best Effort Service is not selected for Data VCCs.

Valid Values: an integer Kbps in the range of 0 - PCR value for Data VCC

Default Value: 0

Max SDU size (bytes) Specifies the Maximum SDU size that will be specified when calls are placed from this client address. It also is used to verify incoming calls. This parameter cannot be set to a value

greater than the Maximum SDU size for the physical ATM interface (port).

Valid Values: an integer in the range of 72 - Maximum interface SDU

Default Value: 9188

Example for IPX:

```
ARP config> add atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]? IPX
Refresh timeout (in minutes) [5]?
Enable auto-refresh? [Yes]:
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [Yes]:
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Service for Control VCCs? [Yes]:
Peak Cell Rate of outbound control VCCs (Kbps) [0]?
Sustained Cell Rate of outbound control VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
```

For field descriptions, refer to the preceding example for IP.

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.
```

pvc-atm-arp-entry

Adds a PVC and optionally creates permanent ARP Entry if the destination protocol address is specified. For virtual ATM interfaces, you should check the configuration of the real ATM interface where the AVI sits and all the other AVIs configured on the real ATM interface. A new VPI/VCI pair is needed for a new PVC unless you specifically want to share the new PVC traffic with the traffic of an existing PVC.

interface number

Valid values: the number of the interface over which the device will boot or dump.

Default value: 0

protocol

Valid values: *IP*, *IPX*, *ASRT*

Default value: IP

local client IP address

Required for IP

Valid Values: any valid IP address

Default Value: 0.0.0.0

destination protocol address

Valid Values: any valid IP address

Default Value: 0.0.0.0

destination ATM address

Valid Values: any valid IP address

Default Value: none

permanent virtual circuit VPI

Valid Values: any valid value in the range of 0 - 255

Default Value: 0

permanent virtual circuit VCI

Valid Values: any value in the range of 0 - 65535

Default Value: 0

Example for IP:

```
ARP config> add pvc-atm-arp-entry
Interface Number [0]?
Protocol [IP]?
Local client IP address [0.0.0.0]? 2.2.3.100
Specify destination protocol address? [Yes]: no
Permanent Virtual Circuit VPI, Range 00..FF [00]?
Permanent Virtual Circuit VCI, Range 0000..FFFF [0000]? 0029
```

Interface Number Interface number assigned.

Protocol IP ,IPX, ASRT

Specify destination protocol address
If yes, you will be required to enter the destination protocol address.

Example for IPX:

```
ARP config> add pvc-atm-arp-entry
Interface Number [0]?
Protocol [IP]? IPX
Specify destination protocol address? [Yes]: no
Permanent Virtual Circuit VPI, Range 00..FF [00]?
Permanent Virtual Circuit VCI, Range 0000..FFFF [0000]? 0037
```

If you choose to specify a destination protocol address, enter a valid 6-byte IPX host address.

For field descriptions, refer to the preceding example for IP.

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.
```

Redundancy

Use this command to add a redundancy configuration.

ARP Server Redundancy Example

```
*********************************************************
* Configuring the Primary for Redundancy               *
*********************************************************
ARP config>add red
Interface Number [0]?
Protocol [IP]?
Local client IP Address [0.0.0.0]? 1.1.1.1
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) AAAAAAAAAAA
Select ESI [1]? 3
Redundancy selector, Range 00..FF [00]? aa
Is this client the primary ? [Yes]:
Partner's (Redundancy) ATM Address, 40 Hex digits
[00000000000000000000000000000000000000000000]?
39.84.0F.00.00.00.00.00.00.00.00.01.bb.bb.bb.bb.bb.bb.bb  1
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
*********************************************************
* Configuring the Secondary for Redundancy             *
*********************************************************
ARP config>add red
Interface Number [0]?
Protocol [IP]?
Local client IP Address [0.0.0.0]? 1.1.1.2
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) BBBBBBBBBBB
Select ESI [1]? 3
Redundancy selector, Range 00..FF [00]? bb
Is this client the primary ? [No]:
Partner's (Redundancy) ATM Address, 40 Hex digits
[00000000000000000000000000000000000000000000]?
39.84.0F.00.00.00.00.00.00.00.00.01.aa.aa.aa.aa.aa.aa.aa  1
If Secondary, Primary Server ESI [1]? 2
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) BBBBBBBBBBB
Primary Server 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
```

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

Valid Values: Any IP address

Default Values: 0.0.0.0

Select Redundancy ESI

Specifies whether a MAC address burned into the ATM device should be used as the End System Identifier component of the CIPC or CIPS ATM address.

Valid Values: Any of the values listed in the menu preceding this question.

Default Values: 1

Choose Redundancy Selector

Identifies the selector byte for the Redundancy ATM address.

Valid Values: any single octet value that has not been previously used and is within the range defined for the device.

Default Values: 00

Is This Client Acting As The Primary

If Yes, this client will place a call from its redundancy ATM address to the secondary's redundancy address. If No, this client will receive calls only from the Primary's redundancy ATM address

Valid Values: Yes or No

Default Values: No

Partner's (Redundancy) ATM Address

Specifies whether a burned in ESI or an ESI configured under ATM interface configuration should be used as the ESI for the redundancy ATM address. This question is preceded by a list of valid ESIs from which the selection is to be made.

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

Primary Server ESI

Specifies the ESI component of the primary's real ATM address. The backup uses this value as the ESI component of it's real ATM address if the primary fails.

Default Value: none

Primary Server Selector

Specifies the selector component of the primary real ATM address. The backup uses this value as the selector component of it's real ATM address if the primary fails.

Valid Values: value defined for primary server selector

Default Value: 00

Redundancy's default IP gateway

Specifies whether this ARP entity will participate in the provision of default gateway redundancy support for the LIS.

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 MSS Server as their default router.

Default Value: 0.0.0.0

svc-atm-arp-entry
Adds an SVC and optionally creates a permanent ARP Entry.

interface number
Valid values: the number of the interface over which the device will boot or dump.

Default value: 0

protocol
Valid values: *IP*, *IPX*, or *ASRT*

Default value: IP

local client IP address
Required for IP

Valid Values: any valid IP address

Default Value: 0.0.0.0

destination protocol address
Valid Values: any valid IP address

Default Value: 0.0.0.0

destination ATM address
Valid Values: any valid IP address

Default Value: none

Example for IP:

```
ARP config> add svc-atm-arp-entry
Interface Number [0]?
Protocol [IP]?
Local client IP address [0.0.0.0]? 2.2.3.100
Specify destination protocol address? [Yes]: no
Destination ATM Address []? 39840f000000000000000000000210005a00dead03
```

Interface Number Interface number assigned.

Protocol IP or IPX.

Specify destination protocol address
If no, it is resolved by InATMARP requests/replies.

Destination ATM Address Destination ATM address.

Example for IPX:

```
ARP config> add svc-atm-arp-entry
Interface Number [0]?
Protocol [IP]? IPX
Specify destination protocol address? [Yes]: no
Destination ATM Address []? 39840f0000000000000000000210005a00dead03
```

If you choose to specify a destination protocol address, enter a valid 6-byte IPX host address.

For field descriptions, refer to the preceding example for IP.

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.
```

Change

Use the **change** command to change the ATM-ARP configuration.

Syntax: change
entry
atm-arp-client-configuration
redundancy

atm-arp-client-configuration
Changes the atm-arp-client-configuration.

interface number
Valid values: any interface on the device

Default value: 0

protocol
Valid values: *IP*, *IPX*, or *ASRT*

Default value: IP

If you enter IP, you must enter the client IP addresss.

client IP address
Valid Values: any valid IP address

Default value: 0.0.0.0

refresh timeout
Valid Values: an integer number of minutes in the range of 0 - 65535

Default Value: 5 minutes

ESI a defined ESI index number.

maximum reserved bandwidth for incoming VCCs
Valid values: the number of any defined ESI

Default Value: 1 which specifies to used the burned in address

selector value
You enter this if you respond **no** to the “Use internally assigned selector” question.

Valid Values: any single octet value that has not been previously used and is within the range defined for the device.

Default Value: none

maximum reserved bandwidth for incoming VCCs

Valid Values: an integer in the range of 0 to the line speed of the ATM device.

Default Value: 0

use best effort service for control VCCs

Valid Values: *Best Effort* or *Reserved Bandwidth*

Default Value: Best Effort

peak cell rate of outbound control VCCs

Valid Values: an integer Kbps in the range of 0 - line speed of the ATM device

Default Value: 0

sustained cell rate of outbound control VCCs

Valid Values: an integer Kbps in the range of 0 - control VCC PCR

Default Value: 0

peak cell rate of outbound data VCCs

Valid Values: an integer Kbps in the range of 0 - control VCC PCR

Default Value: 0

sustained cell rate of outbound data VCCs

Valid Values: an integer Kbps in the range of 0 - PCR value for Data VCC

Default Value: 0

max service data unit (SDU) size

Valid Values: an integer in the range of 72 - Maximum interface SDU

Default Value: 9188

Specify the IP address of the atm-arp-client.

Example for IP:

```
ARP config> change atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]?
Client IP Address [0.0.0.0]? 1.1.1.100
This client is also a server? [No]:
Refresh timeout (in minutes) [5]?
Enable auto-refresh? [No]: yes
Refresh by InAtmArp? [Yes]:
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [Yes]:
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Service for Control VCCs? [Yes]:
Peak Cell Rate of outbound control VCCs (Kbps) [0]?
Sustained Cell Rate of outbound control VCCs (Kbps) [0]?
Use Best Effort Service for Data VCCs? [Yes]:
Peak Cell Rate of outbound Data VCCs (Kbps) [0]?
Sustained Cell Rate of outbound Data VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
```

For field descriptions, refer to the example for **add atm-arp-client-configuration**.

Example for IPX:

```
ARP config> change atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]? IPX
Refresh timeout (in minutes) [5]?
Enable auto-refresh? [Yes]:
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [No]:
Selector Only, Range 00..FF [00]? 20
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Service for Data VCCs? [Yes]:
Peak Cell Rate of outbound data VCCs (Kbps) [0]?
Sustained Cell Rate of outbound Data VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
```

Since only one IPX ATM-ARP client configuration record exists for an ATM interface, you are not prompted to enter a protocol address.

For field descriptions, refer to the preceding example for IP.

Example for Bridging:

```
ARP config> change atm-arp-client-configuration
Interface Number [0]?
Protocol [IP]? ASRT
Client Address (Port Number) [0]? 1      1
  (  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [Yes]:
Validate PCR for best effort VCCs? [No]:
Maximum Reserved Bandwidth for incoming VCCs (Kbps) [0]?
Use Best Effort Service for 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.

Use of this command is required only if you wish to use values other than the defaults for the traffic parameters.

For field descriptions, refer to the preceding example for IP.

redundancy
Changes the redundancy configuration.

Delete

Use the **delete** command to delete an arp-server, atm-arp-client-configuration, pvc-atm-arp-entry, or svc-atm-arp-entry.

Syntax: delete entry
arp-server
atm-arp-client-configuration
pvc-atm-arp-entry
svc-atm-arp-entry
redundancy

arp-server
Deletes an arp-server.

Specify the address of the arp-server.

Valid Values: any valid IP address

Default Value: 1.1.1.100

Example for IP:

```
ARP config> del arp-server
IP Address [1.1.1.100]? 2.2.3.100
Arp Server entry found and deleted
```

atm-arp-client-configuration
Deletes an atm-arp-client-configuration.

Specify the interface number, protocol, and Client IP address.

interface number
Valid values: any defined interface

Default value: 0

protocol
Valid values: *IP*, *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.
```

redundancy
: Deletes the redundancy configuration.

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.

List

Use the **list** command to display the contents of the router's ARP configuration as stored in SRAM. The list command also displays the current settings for the refresh and usage timer.

Syntax: list
- entry
- all
- arp-servers
- atm-arp-client-configuration
- pvc-atm-arp-entry
- svc-atm-arp-entry
- redundancy

all Lists the ARP configuration followed by all of the ARP entries.

Example: `list all`

```
ARP config> list all
ARP configuration:

Refresh timeout: 5 minutes
Auto refresh: disabled

Mac address translation configuration

No arp entries defined

ATM Arp Server List:
   IP Address   L/R    Address / Sub Address
   1.1.1.100  R  39.84.0F.00.00.00.00.00.00.00.00.00.02.10.00.5A.00.DE.AD.02
```

arp-servers
Lists arp-servers.

```
ARP config> list arp-servers

ATM Arp Server List:
IP Address   L/R    Address / Sub Address
   1.1.1.100  R  39.84.0F.00.00.00.00.00.00.00.00.00.02.10.00.5A.00.DE.AD.02
```

atm-arp-client-configuration
Lists the atm-arp-client-configuration.

```
ARP config> list atm-arp-client-configuration

ATM Arp Clients:
-------------------------------------------------
If: 0  Prot: 0  Addr: 1.1.1.100        ESI: burned-in          Sel: auto
Server: no   Refresh T/O: 5    AutoRefr: no    By InArp: yes  Validate PCR: no
Use Best Effort: yes/yes  (Control/Data)   Max B/W(kbps):      0
Cell Rate(kbps):  Peak:      0/     0    Sustained:      0/     0
Max SDU(bytes):   9188
-------------------------------------------------
If: 0  Prot: 0  Addr: 3.3.3.3          ESI: burned-in          Sel: auto
Server: yes  Refresh T/O: 5    AutoRefr: no    By InArp: yes  Validate PCR: no
Use Best Effort: yes/yes  (Control/Data)   Max B/W(kbps):      0
Cell Rate(kbps):  Peak:      0/     0    Sustained:      0/     0
Max SDU(bytes):   9188
-------------------------------------------------
If: 0  Prot: 0  Addr: 4.4.4.4          ESI: burned-in          Sel: auto
Server: yes  Refresh T/O: 5    AutoRefr: no    By InArp: yes  Validate PCR: no
Use Best Effort: yes/yes  (Control/Data)   Max B/W(kbps):      0
Cell Rate(kbps):  Peak:      0/     0    Sustained:      0/     0
Max SDU(bytes):   9188
-------------------------------------------------
If: 0  Prot:23  Port: 1    ESI: burned-in    Sel: auto  Validate PCR: no
Use Best Effort: yes  (Data)              Max B/W(kbps):      0
Cell Rate(kbps):  Peak:      0            Sustained:      0
Max SDU(bytes):   9188
```

If: Interface Number

Prot: 0 = IP, 7 = IPX, 23 = ASRT

Addr: IP Address

Port: Bridge port number if Prot = ASRT

ESI: End System Identifier

Sel: Selector (the last byte in the ATM address, following the ESI). If AUTO, the selector is generated at run-time.

Server: YES, this client is also a server; NO, this client is not a server.

Refresh T/O: Refresh Timeout value in minutes.

AutoRefr YES or NO.

By InArp: YES or NO. If YES, and if auto-refresh is enabled, then InAtmArp requests will be periodically transmitted to confirm the existence of the remote host. If NO, then AtmArp requests will be transmitted to the ARP Server to reconfirm the ARP entry.

Validate PCR: TRUE or FALSE. When true, Best-Effort VCCs will be rejected if the signaled forward PCR exceeds the Maximum Reserved Bandwidth or the speed of the adapter. If false, Best-Effort PCRs will be rejected without regard to the signaled Peak Cell Rate.

Use Best Effort: Specifies the type of traffic characteristics to be associated with the Control or Data VCCs.

Max B/W(kbps): Maximum bandwidth (kbps).

Cell Rate(kbps): Peak Cell Rate for the Control or Data VCC.

Max SDU(bytes): The maximum SDU size that is specified.

redundancy

Lists the redundancy configurations.

```
/************************************************************/
/* List ARP Server Redundancy under ATM ARP Configuration   */
/* List Redundancy on the Secondary                         */
/************************************************************/

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
Primary Server ESI: 11.11.11.11.11.11  Primary Server SEL: 11   Redundancy Default IP Gateway A
ddress: 1.1.1.3
-----------------------------------------

/************************************************************/
/* List ARP Server Redundancy under ATM ARP Configuration   */
/* List Redundancy on the Primary                           */
/************************************************************/

ARP config>list red


ATMARP Clients with Redundancy Configured
-----------------------------------------

If:  0    Prot: IP    Addr: 1.1.1.1
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
Redundancy Default IP Gateway Address: 1.1.1.3
-----------------------------------------
```

If: Interface Number

Prot: IP

Addr: IP Address

Red. ESI:

Burned In - Specifies whether the MAC address burned into 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.

Pri/Secy: 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.

Partner's (Red.) ATM Address:
Specifies the redundancy ATM address for the partner client/server.

Primary Server ESI: Identifies a locally administered ESI that is to be used as the ESI component of the secondary ARP server ATM address when the primary ARP server is down.

Primary Server SEL: Identifies the Selector portion of the primary server ESI and selector configured on the partner server

Redundancy Default IP Gateway Address: Specifies the default IP gateway address. This is the default gateway address configured in the client's 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
```

ATM No. ATM interface number

ARP IF#. ARP Interface Number

Prot# Protocol number (Prot# 0 = IP, 7 = IPX)

P/S: P for PVC, S for SVC

Protocol IP Address

VPI/VCI The decimal value of the defined channel.

svc-atm-arp-entry
Lists ARP SVCs.

```
ARP config> list svc

ATM Arp Switched Virtual Circuit Definitions
No. IF# Prot# P/S     Protocol     -> Destination ATM Address
2    0    0    S         0.0.0.0   ->
                    39.84.0F.00.00.00.00.00.00.00.00.00.02.10.00.5A.00.DE.AD.03
```

ATM No. ATM interface number

ARP IF#. ARP Interface Number

Prot# Protocol number (Prot# 0 = IP, 7 = IPX)

P/S: P for PVC, S for SVC

Protocol IP Address

Destination ATM Address Destination ATM Address

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Sample ARP Configurations

Sample configurations show the complete sequence of events for that configuration.

Configuring MSS for ARP Server Redundancy

Configuring the Primary

```
/*******************************************************/
/* Sample configuration for the ARP Server Redundancy    */
/* Configuring the Primary ARP Server                    */
/*******************************************************/
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 Service for Control VCCs? [Yes]:
Peak Cell Rate of outbound control VCCs (Kbps) [0]?
Sustained Cell Rate of outbound control VCCs (Kbps) [0]?
Use Best Effort Service for Data VCCs? [Yes]:
Peak Cell Rate of outbound Data VCCs (Kbps) [0]?
Sustained Cell Rate of outbound Data VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
ARP config>add red      8
Interface Number [0]?
Protocol [IP]?
Local client IP Address [0.0.0.0]? 1.1.1.1
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) AAAAAAAAAAAA
Select ESI [1]? 3
Redundancy selector, Range 00..FF [00]? aa
Is this client the primary ? [Yes]:
Partner's (Redundancy) ATM Address, 40 Hex digits
[00000000000000000000000000000000000000000000]? 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
```

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

Configuring the Secondary

```
/*********************************************************/
/* Sample configuration for the  ARP Server Redundancy     */
/* Configuring 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 Service for Control VCCs? [Yes]:
Peak Cell Rate of outbound control VCCs (Kbps) [0]?
Sustained Cell Rate of outbound control VCCs (Kbps) [0]?
Use Best Effort Service for Data VCCs? [Yes]:
Peak Cell Rate of outbound Data VCCs (Kbps) [0]?
Sustained Cell Rate of outbound Data VCCs (Kbps) [0]?
Max SDU size (bytes) [9188]?
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 []? 39840F000000000000000000111111111111111
ARP config>add red  8
Interface Number [0]?
Protocol [IP]?
Local client IP Address [0.0.0.0]? 1.1.1.2
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) BBBBBBBBBBBB
Select ESI [1]? 3
Redundancy selector, Range 00..FF [00]? bb
Is this client the primary ? [No]:  9
Partner's (Redundancy) ATM Address, 40 Hex digits
[0000000000000000000000000000000000000000]? 39.84.0F.00.00.00.00.00.00.00.00.01.aa.aa.aa.aa.aa.aa.aa
If Secondary, Primary Server ESI [1]? 2
  (  1) Use burned in ESI
  (  2) 111111111111
  (  3) BBBBBBBBBBBB
Primary Server 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 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

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

Chapter 23. Monitoring ARP

This chapter describes how to monitor ARP protocol activity and how to use the ARP console commands and 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.

Accessing the ARP Console Environment

Use the following procedure to access the ARP console commands. This process gives you access to the ARP *monitoring* process.

1. At the OPCON prompt, enter **talk 5**. (For more detail on this command, refer to *The OPCON Process and Commands* in the Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1.) For example:

   ```
   * talk 5
   +
   ```

   After you enter the **talk 5** command, the GWCON prompt (+) displays on the console. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the + prompt, enter the **protocol arp** command to get you to the ARP> prompt.

   Example:

   ```
   + prot arp
   ARP>
   ```

ARP Console Commands

This section summarizes and then explains the ARP console commands. You can access ARP console commands at the ARP> prompt. Table 23-1 shows the commands.

Table 23-1. ARP Console Command Summary

Command	Function
? (Help)	List the ARP console commands or list the options associated with specific commands.
Clear	Clear the cache for a specified interface.
Dump	Display the cache for a specified interface.
Hardware	List each ARP-configured network.
Protocol	List each ARP-configured protocol.
Statistics	Display ARP information.
Exit	Exit the ARP console process.

? (Help)

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

Syntax: ?

Example: **?**

Clear

Use the **clear** command to flush the ARP cache for a given network interface. The **clear** command can be used to force the deletion of bad transactions.

To clear a particular interface, enter the interface or network number as part of the command. To obtain the interface number, use the CONFIG **list devices** command.

Syntax: clear *interface#*

Example: **clear 1**

Dump

Use the **dump** command to display the ARP cache for a given network/protocol combination. To display the ARP cache for a particular interface, enter the interface or network number as part of the command. To obtain the interface number, use the CONFIG **list devices** command.

If there is more than one protocol on that network, the protocol number must also be given. This causes the console to display the hardware address-to-protocol mappings stored in that database. If ARP is in use by only one protocol on the specified interface, then the protocol number is optional. To obtain the protocol number, use the CONFIG **protocol** command.

The **dump** command display shows the hardware address, the protocol address, and the refresh timer parameter for each mapping.

Syntax: dump *interface# protocol#*

Example: **dump 2 ip**

```
Hardware Address    IP Address      Refresh
02-07-01-00-00-01   192.9.1.2       Permanent
a1-b2-c3-4d-5e-6f   128.185.214.36  5
100                 128.185.123.51  Not Aging
16                  128.185.214.38  Not Aging
```

Valid refresh timer parameters are:

Permanent — A statically configured mapping between hardware address and protocol address (entered using the ARP **add entry** command, or the frame-relay **add protocol** command, or the X25 **add address** command). These entries do not age and are not overwritten by dynamically learned mappings.

minutes to expire — The number of minutes until this mapping expires due to aging or until this mapping is refreshed (if auto-refresh is enabled). This parameter is expressed as a numeric value.

Not Aging — A fixed SVC or PVC mapping learned through Inverse ARP. It begins to age only when the circuit goes down. The mapping can be overwritten by a newer learned address and can be cleared by the ARP **clear** console command.

Hardware

Use the **hardware** command to display the networks registered with ARP. The **hardware** command lists each ARP-registered network, and displays each network's hardware address space (Hardware AS) and local hardware address.

Syntax: hardware

Example: **hardware**

```
Network       Hardware AS    Hardware Address

1 FR/0          000F            1023
5 TKR/0         0006            00:00:C9:09:32:EF
8 Eth/0         0001            AA-00-04-00-26-14
9 IPPN/0        2048            128.185.214.38
10 BDG/0        0001            00-00-93-90-4C-F7
```

Note: The IPPN entry refers to IP Tunneling where the hardware address field indicates the IP address of the IP Tunnel.

Ping

Use the **ping** command to have the router send ICMP Echo Requests to a given destination. For more information on the **ping** command, see "Ping" on page 15-6.

Protocol

Use the **protocol** command to display (by network) the protocols that have addresses registered with ARP. This command displays the network, protocol name, protocol number, protocol address space (in hexadecimal), and local protocol addresses.

Syntax: protocol

Example: **protocol**

```
Network  Protocol (num)  AS    Protocol Address(es)
5 TKR/0  IP       (00)   800    128.185.209.38
6 TKR/1  IP       (00)   800    10.1.181.38
8 Eth/0  IP       (00)   800    128.185.221.38
8 Eth/0  AP2      (22)   80F3   221/38
```

Note: SR entries refer to Source Routing - the protocol address is used to indicate the MAC address. Use the token-ring **dump** command to view actual RIF entries.

Statistics

Use the **statistics** command to display a variety of statistics about the operation of the ARP module.

Syntax: statistics

Example: **statistics**

```
ARP input packet overflows
Net   Count
 PPP/0  0
 PPP/1  0
 TKR/0  0
IPPN/0  0
 BDG/0  0ARP

ARP cache meters
Net Prot  Max Cur Cnt    Alloc  Refresh: Tot  Failure  TMOs: Refresh
0   0     1   1   1       17             0        0               13
0  22     1   0   0        6             0        0                6
1   0     1   1   2       27             0        0               25
1  16     3   3   7      291             0        0                0
2   0     1   0   0        2             0        0                2
2  16     1   0   0        1             0        0                0
8   0     1   1   1       11             0        0               10
```

ARP input packet overflows	Displays counters that represent the number of ARP packets discarded on input because the ARP layer was too busy. The counts shown are per network interface.
ARP cache meters	Consists of a variety of meters on the operation of the ARP cache. The counts shown are all per protocol, per interface.
Net	Displays the interface numbers.
Prot	Displays the protocol numbers.
Max	Displays the all-time maximum length hash chain.
Cur	Displays the current maximum length hash chain.
Cnt	Displays the count of entries currently active.
Alloc	Displays the count of entries created.

Rfrsh:Tot	Displays the number of refresh requests sent for this network interface and protocol.
Fail	Displays the number of auto-refresh attempt failures due to unavailability of internal resources. This count is not related to whether or not an entry was refreshed.
TMOs:Rfrsh	Displays the count of entries deleted due to a timeout of the refresh timer.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

ARP Over ATM Console Commands

This section summarizes and then explains the ARP over ATM (CIP) console commands. It describes the console commands for:

- Classical IP & ARP over ATM
- IPX over ATM
- Bridging over ATM

The console commands for IPX and ARP over ATM are essentially the same as those for Classical IP and ARP. The main difference is the format of protocol addresses:

- Protocol addresses for IP are specified as 4-byte fields in dotted decimal notation.
- Protocol addresses for IPX are specified as 6-byte fields in hexadecimal characters.

Note: The **ping** command for IPX over ATM is different from that used for Classical IP and ARP. The IPX version of the **ping** command is available at the IPX console. You can access ARP console commands at the ARP> prompt. Table 23-2 on page 23-7 shows the commands.

For further information, refer to "Classical IP and ARP Over ATM Overview (RFC 1577)" on page 22-4 and "IPX and ARP Over ATM Overview (RFC 1483)" on page 22-10. For additional information on ARP over ATM, and for illustrations showing logical and physical network configurations, refer to *Multiprotocol Switched Services (MSS) Server Configuration and Operations Guide*.

Since bridging does not use ARP, the console may only be used to check the status of a channel associated with a bridge port. Also, since bridging does not use protocol addresses, only the port number for the associated (local) port is displayed with a channel.

Table 23-2. ARP Over ATM Console Command Summary

Command	Function
? (Help)	List the ARP console commands or list the options associated with specific commands.
Delete	Immediately bring down an active channel. A new channel may or may not be brought up to replace the old one depending on the conditions.
Display	Display all of the channels (VCCs) associated with a single ATM interface.
Dump	Show which ATM channels are being used for sending datagrams and show their corresponding IP addresses.
Hardware	List each ARP-configured network.
Ping	Verify connectivity between the device and the specified end station.
Protocol	List each ARP-configured protocol.
Redundancy-State	Display IP clients configured with Redundancy
Statistics	Display statistics of the ARP code over all of the network interfaces.
Exit	Exit the ARP console process.

? (Help)

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

Syntax: List output

Example: **`redundancy`**

Example of IP configured with Redundancy

```
*t 5
CGW Operator Console
+p arp
ARP>red
Network number  [0]?
Protocol [IP]?
If: 0 Prot: IP  Clients configured with Redundancy
------------------------------------------------------------
Addr: 1.1.1.1   Pri/Secy: Pri  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.00.00.00.00.00.00.00.00.00.02.AA.AA.AA.AA.AA.AA.AA
                    Red. Channel: Target ATM address
39.84.0F.00.00.00.00.00.00.00.00.00.03.BB.BB.BB.BB.BB.BB.BB
                    Redundancy Status: Active

------------------------------------------
```

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   01    00/0068   155000000 155000000      9188      F     T
        Tgt Addr.:
        Client Address (owner): Port No. 1
  2) P  80   03    00/0048   155000000 155000000      9188      F     T
        Tgt Addr.:
        Client Address (owner): Port No. 1
```

P/S — P means that this channel is a PVC. S means that this channel is an SVC. In the example, PVC #2 is for bridging.

List — For internal use.

Flags — For internal use.

VPI/VCI — Virtual Path Identifier and Virtual Channel Identifier of the channel in use.

FwdPcr — The Peak Cell Rate in bits per second.

FwdScr — The Sustained Cell Rate in bits per second.

MaxSDUsz	The maximum SDU size for this channel. All packets transmitted or received on this interface must be less than or equal to this size less the 8-byte header prefix used by RFC 1483.
Control	T if this is a control channel (channel to the ARP server). F if this is a data channel (channel to another client).
P2P	T if this channel is point-to-point. F if this channel is point-to-multipoint.
Active Channel List	These channels are true connections with the remote party. Data can flow over these connections with the traffic parameters shown.
New Channel List	These channels are in the process of being connected with the other end. No data may flow over them until they are moved to the active list.
PVC Channel List	These are channels which have been specifically configured as PVCs. They take on the client characteristics for Data Channels as defined in the client configuration.

Dump

Use the **dump** command to show which ATM channels are being used for sending datagrams, and their corresponding IP addresses.

This table represents the entire ARP table for a physical ATM network running Classical IP. The hardware address is the resultant VCC identifier (VPI/VCI) for an active channel. That is, all traffic that is to be sent to the IP address will be transmitted out on the associated channel (listed under Hardware Address).

Note: If the host on the other end of the channel sends either a request or reply with its own address, we will automatically reset the refresh time to its maximum value.

Syntax: dump

Example: **dump**

```
ARP> dump 0
Hardware Address          IP Address                Refresh
0x00/0x0020               1.1.1.2                   not aging
```

Under Refresh, the time specified is the approximate time before the ARP entry is aged out (in minutes). If autorefresh is turned on, then an ARP request or an InATMARP request will be sent out 30 seconds before the expiration. If a reply is received before expiration, the Refresh time is reset, and the ARP entry remains. If no reply is received, or if autorefresh is turned off, the ARP entry will be deleted when it expires. It will be recreated as required.

If “not aging” appears under Refresh, that entry will remain indefinitely.

Hardware

Use the **hardware** command to list all of the ATM addresses associated with each configured IP client.

Syntax: hardware

Example: **hardware**

```
ARP> hardware
    Network         Hardware AS     Hardware Address
    0 ATM/0         0013            39.84.0F.00.00.00.00.00.00.00.00.01.
                                    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.00.00.00.00.00.00.00.00.01.10.00.5A.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

```
ARP Server Redundancy Console screens:
/*********************************************************/
/* Redundancy console section on the Primary ARP Server    */
/* When Redundancy Channel is inactive                     */
/*********************************************************/

CGW Operator Console

+p arp

ARP>red

Network number  [0]?

Protocol [IP]

If: 0 Prot: IP  Clients configured with Redundancy

---------------------------------------------------------

Addr: 1.1.1.1   Pri/Secy: Pri  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
```

```
/*********************************************************/
/* Redundancy console section on the Primary Server      */
/* When Redundancy Channel is active                     */
/*********************************************************/

CGW Operator Console

+p arp


ARP>red

Network number  [0]?

Protocol [IP]?

If: 0 Prot: IP  Clients configured with Redundancy

---------------------------------------------------------

Addr: 1.1.1.1   Pri/Secy: Pri  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


-----------------------------------------
```

```
/************************************************************/
/* Redundancy console section on the Secondary              */
/* When the Redundancy channel is inactive and the Secondary */
/* 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   Pri/Secy: Secy  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

Primary Server ESI: 11.11.11.11.11.11, Primary Server SEL: 11, In backup Server mode

Redundancy default IP Gateway Protocol address: 1.1.1.3

-------------------------------------------
```

```
/************************************************************/
/* Redundancy console 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   Pri/Secy: Secy  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 ...

Primary Server ESI: 11.11.11.11.11.11, Primary Server SEL: 11, Acting as a Client

Redundancy default IP Gateway Protocol address: 1.1.1.3

-------------------------------------------

ARP>exit
```

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 23-4.

Syntax: statistics

Example: **statistics**

```
ARP> statistics

    ARP input packet overflows
        Net   Count
       ATM/0  0
      IPPN/0  0
       BDG/0  0

    ARP cache meters
    Net Prot  Max Cur Cnt    Alloc  Refresh: Tot  Failure  TMOs: Refresh
      0   0     1   1   1        1             0        0              0
```

Chapter 24. Using and Configuring BGP4

This chapter describes how to configure 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 endnodes that operate under a single administrative organization. Within each AS, routers and endnodes 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.

Connections between BGP speakers in the same AS are called internal BGP (IBGP) connections, while connections between BGP speakers in different autonomous systems are external BGP (EBGP) connections.

A single AS may have one or many BGP connections to outside autonomous systems. Figure 24-1 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.

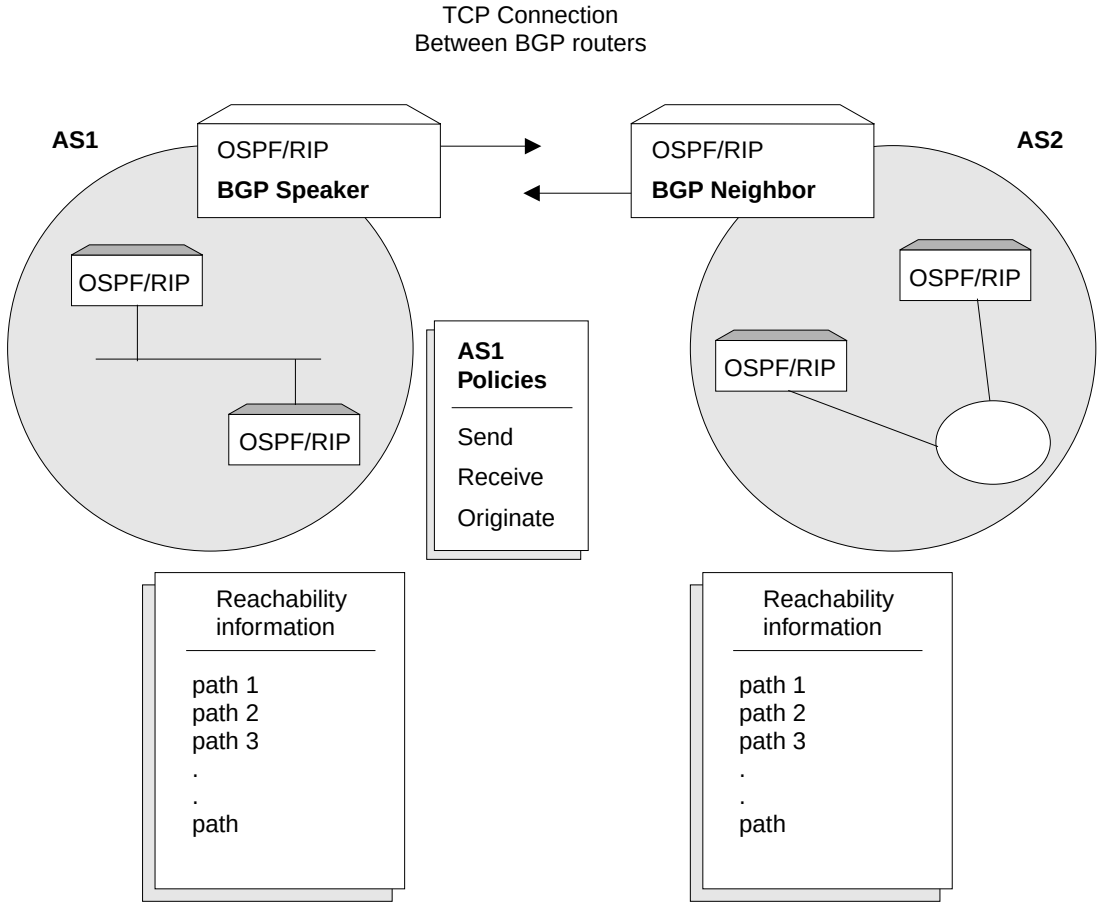


Figure 24-1. BGP Connections between Two Autonomous Systems. Once the BGP speaker in AS1 establishes a TCP connection with its BGP neighbor in AS2, the two routers can selectively exchange reachability information. The information each router sends or accepts is determined by policies defined for each router.

While the autonomous systems shown in Figure 24-1 have only one BGP router, each could have multiple connections to other autonomous systems. As an example of this, Figure 24-2 on page 24-3 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.

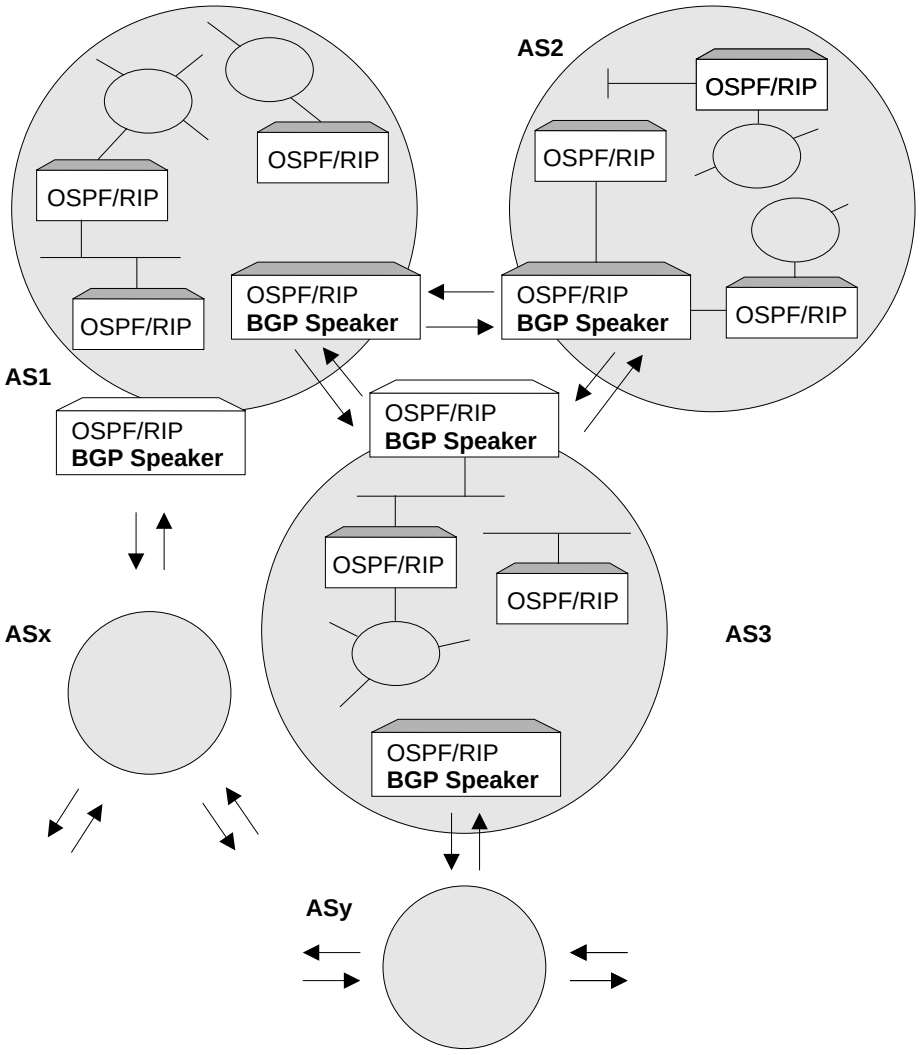


Figure 24-2. BGP Connections among Three Autonomous Systems. Note that AS1 and AS3 have two BGP speakers.

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
- Receive Policies

Once a TCP connection is established, the BGP speaker shown in Figure 24-1 on page 24-2 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.

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.

   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.

   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 24-5 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.

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
```

Adding a BGP neighbor automatically enables it, causing the BGP speaker to send out a connection request to the neighbor.

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.

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 24-8 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 BGP 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 “Using and Configuring IP” in *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1* for information on setting the tag value.

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 a Transit 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 AS-path.

This example will prevent the BGP speaker from importing any route that has AS 175 in its AS-path list.

```
BGP Config> add no-receive
Enter AS: [0]? 175
```

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]?
```

Accessing the BGP4 Console Environment

For information on how to access the BGP console environment, see “Getting Started (Introduction to the User Interface)” in the Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1.

BGP4 Configuration Commands

This section summarizes and then explains all 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 24-1. BGP Command Summary

Command	Function
? (Help)	Lists the configuration commands or lists the actions associated with specific commands.
Add	Add BGP neighbors.
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 or BGP neighbors.
List	Displays BGP configuration items.
Move	Changes the order in which policies and aggregates are defined.
Exit	Exits the process.

? (Help)

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

Syntax: ?
Example: **?**

```
ADD
CHANGE
DELETE
DISABLE
ENABLE
LIST
MOVE
EXIT
```

Add

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

Syntax: add aggregate . . .
neighbor . . .
no-receive asnum . . .
originate-policy . . .
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 support both the individual routes and the aggregate you have defined.

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. An internal neighbor must exist on the same network as the speaker.

1. The IP *address of the neighbor* has:

   Valid Values: Any valid IP address.

   Default Value: none

2. 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 transitioned 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 Keep-alive 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 Keep-Alives 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.

   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]?
```

Neighbor address
: 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.

AS
: Your own autonomous system number for internal neighbor or neighbor's autonomous system number.

Init Timer
: 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 transitioned to IDLE state due to an error. If the error persists, this timer increases exponentially. The default is 12 seconds.

Connect Timer
: 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 meantime, the BGP speaker continues to listen for any connection that may be initiated by its neighbor. The default is 120 seconds.

Hold Timer
: 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. The default is 90 seconds. Once neighbors have established BGP connection, they exchange Keep-alive 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 3 seconds. Note that, on switched lines, you may wish to have the Hold Timer value of zero to save bandwidth by not sending Keep-Alive messages at frequent intervals.

TCP Segment Size
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. It defaults to 1024, but can be set up in the range 1 to 65535.

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.
Address match	The address, or range of addresses, that will be affected by the policy statement.
Tag	The value that has been set for a particular AS. All tag values match that of the AS from which they were learned.

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. 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

3. Select whether the *Address match* is to be 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

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 24-5 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 "Receive Policy Examples" on page 24-6 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 "Send Policy Examples" on page 24-7 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 . . .
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

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]?
```

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 . . .
receive-policy . . .
send-policy. . .

aggregate *index#*

You must specify the index number of the aggregate you want to delete. The index number is equivalent to the AS number.

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

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**

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
neighbor . . .

disable bgp speaker

Example: `disable bgp speaker`

disable 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
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]?
```

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
- 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
```

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]?
```

Exit

Use the **exit** command to leave the BGP configuration module and return to the `Config>` prompt.

Syntax: exit

Example: **`exit`**

Chapter 25. Monitoring BGP4

This chapter describes the BGP console commands and includes the following sections:

- "Accessing the BGP Console Environment"
- "BGP4 Console Commands"

Accessing the BGP Console Environment

For information on how to access the BGP console environment, see "Getting Started (Introduction to the User Interface)" in the Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1.

BGP4 Console Commands

This section summarizes and then explains all 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 25-1. BGP Command Summary

Command	Function
? (Help)	Lists the monitoring commands or lists the actions associated with specific commands.
Destinations	Displays all entries in the BGP routing table.
Dump routing tables	Lists the contents of the IP routing table.
Neighbors	Displays currently active neighbors.
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.
Sizes	Displays the number of entries in various databases.
Traceroute	Displays the complete path (hop-by-hop) to a particular destination.
Exit	Exits the process.

? (Help)

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

Syntax: ?
Example: **?**

```
DESTINATIONS
NEIGHBORS
PATHS
SIZES
EXIT
```

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: `destinations`

```
Network         Mask       NextHop         MED AAG AGRAS ORG ASPath
128.185.0.0     FFFF0000   192.0.251.165   0   No  0     IGP
142.4.0.0       FFFF0000   192.0.190.178   0   No  0     IGP seq[178]
143.116.0.0     FFFF0000   128.185.252.168 0   No  0     IGP seq[168]
192.0.190.0     FFFFFF00   192.0.251.165   0   No  0     IGP
192.0.251.0     FFFFFF00   192.0.251.165   0   No  0     IGP
194.10.16.0     FFFFF000   192.0.251.167   0   No  167   IGP seq[167]
```

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         Mask     NextHop         MED AAG AGRAS ORG ASPath
142.4.0.0       FFFF0000 192.0.251.165   0   No  0     IGP
seq[165-178]Dest:142.4.0.0, Mask:FFFF0000, 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: 0
        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, Mask:FFFFF000, 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       Mask     NextHop         MED AAG AGRAS ORG ASPath
194.10.16.0   FFFFF000 194.10.16.167   0   No  167   IGP
192.0.190.0   FFFFFF00 192.0.251.165   0   No  0     IGP seq [165]
142.4.0.0     FFFF0000 192.0.251.165   0   No  0     IGP seq [165-178]
143.116.0.0   FFFF0000 128.185.250.168 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     Mask     NextHop         MED AAG AGRAS ORG ASPath
194.10.16.0 FFFFF000 128.185.250.167 0   No  167   IGP seq[167]
192.0.190.0 FFFFFF00 128.185.250.167 0   No  0     IGP seq[167-165]
142.4.0.0   FFFF0000 128.185.250.167 0   No  0     IGP seq[167-165-178]
```

Dump Routing Tables

For a complete explanation of the **dump routing tables** command, refer to "Dump Routing Table" in the "Monitoring IP" chapter of *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

Neighbors

Use the **neighbors** command to display information on all active BGP neighbors.

Syntax: neighbors *internet address*

Example: **neighbors**

```
IP-Address       State        DAY-HH:MM:SS  BGPID           AS     Upd#

128.185.252.168  Established  00000:48:52  128.185.142.168  168     16
192.0.190.178    Established  00002:01:49  142.4.140.178    178     16
192.0.251.167    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
```

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 “Ping” on page 15-6.

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, see its description in the “Monitoring IP” chapter of Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 26. Using and Configuring AppleTalk Phase 2

This chapter describes the AppleTalk Phase 2 (AP2) configuration commands and includes the following sections:

Basic Configuration Procedures

This section outlines the initial steps required to get the AppleTalk Phase 2 protocol up and running. Information on how to make further configuration changes will be covered in the command sections of this chapter. For the new configuration changes to take effect, the router must be restarted.

Enabling Router Parameters

When you configure a router to forward AppleTalk Phase 2 packets, you must enable certain parameters regardless of the number or type of interfaces in the router. If you have multiple routers transferring AppleTalk Phase 2 packets, specify these parameters for each router.

- Globally Enable AppleTalk Phase 2 - To begin, you must globally enable the AppleTalk Phase 2 software using the AppleTalk Phase 2 configuration **enable ap2** command. If the router displays an error in this step, there is no AppleTalk Phase 2 software present in your load. If this is the case, contact your customer service representative.
- Enable Specific Interfaces - You must then enable the specific interfaces over which AppleTalk Phase 2 is to send the packets. Use the **enable interface** *interface number* command to do this.

  Note: When enabling AppleTalk over ATM, you must enable the specific emulated LAN interfaces over which AppleTalk is to send packets. You must not enable AppleTalk over the physical ATM interface. All further uses of the word "interface" in this chapter refer to the emulated LAN interface, not the ATM physical interface.
- Enable Checksumming - You can then determine whether the router will compute DDP checksums of packets it originates. Checksum software does not work correctly in some AppleTalk Phase 2 implementations, so you may not want to originate packets with checksums for compatibility with these implementations. Normally, however, you will want to enable the generation of checksums. Any packet forwarded with a checksum will have its checksum verified.

Setting Network Parameters

You must also specify certain parameters for each network and interface that sends and receives AppleTalk Phase 2 packets. After you have specified the parameters, use the AppleTalk Phase 2 list configuration command to view the results of the configuration.

- Set the Network Range for Seed Routers - Coordinating network ranges and zone lists for all routers on a network is simplified by having specific routers designated as seed routers. Seed routers are configured with the network range and zone list while all other routers are given null values. Null values indicate that the router should query the network for values from the seed routers. For every network (segment) of your interconnected AppleTalk internet, at least one router interface must be configured as the seed router for that network. There are usually several seed routers on a network in case one of them fails. Also, a router can be a seed router for some or all of its network interfaces. Use the **set net-range** command to assign the network range in seed routers.
- Set the Starting Node Number - Use the **set node** command to assign the starting node number for the router. The router will AARP for this node, but if it is already in use, a new node will be chosen.
- Add a Zone Name - You can add one or more zone names for each network in the internetwork. You can add a zone name for a given network in any router connected to that network; however, only the seed router needs to contain the zone name information for a connected network. Attached routers dynamically acquire the zone name from adjacent routers using the ZIP protocol. Apple recommends that, for a given network, you choose the same seed router for the network number and the zone name. The zone name cannot be configured for a network unless the network number is also configured. To add a zone name for each network number, use the AppleTalk Phase 2 configuration **add zone** *name* command.

AppleTalk 2 Zone Filters

ZoneName filtering, although not required for AppleTalk, is a very desirable feature for the security and administration of large AppleTalk Internetworks. There are also provisions for restricting access to networks by net numbers.

General Information

AppleTalk is structured so that every network is identified in two ways. The first is a network number or range of consecutive network numbers that must be unique throughout the internet. The network number combined with the node number uniquely identifies any end station in the internet.

The second identifier for the network is one or more ZoneNames. These ZoneName strings are not unique throughout the internet. The end station is uniquely identified by a combined **object:type:ZoneName-string**.

A router first learns about a network when the new net range appears in the RTMP routing update from a neighboring router. The router then queries the neighbor for the ZoneNames of the new network. Note that the net range is repeated in every new RTMP update but that the ZoneNames are requested only once.

The end stations obtain the network numbers from the broadcasted RTMP (routing information) packets and then choose a node number. This net/node pair is then AARP'd for (AARP Probe) to see if any other end station has already claimed its use. If another station responds, another net/node pair is chosen by the end station and the process repeated until no responses are received.

Why ZoneName Filters?

When the typical AppleTalk end station wants to use a service (printer, file server) on the Apple Internet, it first looks at all available Zones and selects one. It then chooses a service type and requests a list of all names advertising the type in the chosen Zone. Several problems arise from this mechanism.

- A large internet may have many Zones. Presenting the user with a long list to choose from obscures the needed ones (thereby inhibiting usability of the list).
- The server may not want to make itself available throughout the internet (for security reasons). If the Zone that the service is in is not visible to the client, security is enhanced.
- Restricting the Zones that are visible from a department to the rest of the internet will allow the internet administration to let the department control (or not) its own domain while not increasing the overhead for the rest of the internet (reducing administration).

The filtering of network numbers further enhances the security and administration of the internet. Network access is only indirectly controlled by Zone filtering. An unregulated department could add networks with the same Zone names but new net numbers that conflict with other departments. Network number filtering can be used to prevent these random additions of zone names and net numbers from impacting the rest of the network.

How Do You Add Filters?

The router is configured with an exclusive (meaning block the specified zones) or inclusive (meaning allow only these zones) list of Zones for each direction on each interface. The specified interface will not readvertise filtered Zone information in the defined direction. If all Zones in a network's Zonelist are filtered, network information will also be filtered across the interface.

- Use configuration commands **add** and **delete**, to create the filter list for an interface.
- Use configuration commands **enable** and **disable** to specify how the filter list is applied.

  Use similar commands to create network number filters.

Other Commands:

You can use the `AP2 CONFIG>` **list** command to display all filter information for the interfaces. In addition, the **list** command accepts an *interface#* as an argument so that you can list information for only an interface.

Sample Configuration Procedures

This section covers the steps required to get AP2 up and running. For information on how to make further configuration changes, see “AppleTalk Phase 2 Configuration Commands” on page 26-8. For the configuration changes to take effect, you must restart the router.

To access the AP2 configuration environment, enter **protocol ap2** at the `Config>` prompt.

Enabling AP2

When you configure a router to forward AP2 packets, you must enable certain parameters. If you have multiple routers transferring AP2 packets, specify these parameters for each router. To enable AP2:

1. Use the **enable ap2** command to globally enable AP2 on the router. For example:

   ```
   AP2 config>enable ap2
   ```

2. Enable the specific interfaces over which AP2 is to send packets. For example:

   ```
   AP2 config>enable interface 1
   ```

Setting Network Parameters

To set up your router as a seed router, you must set the network range, a starting node number, and at least one zone name. You can configure some interfaces on a router as seed routers and leave other interfaces as non-seed routers. You must have at least one seed router for each AppleTalk network, and you should configure several seed routers on a network in case one of them fails.

1. Use the **set net-range** command to set the Network Range. For example:

   ```
   AP2 config>set net-range
   Interface #  [0]? 1
   First Network range number (1-65279, or 0 to delete) []? 1
   Last Network range number (1-165279) []? 5
   ```

   Enter the same first and last values for a single-numbered network.

2. Use the **set node-number** command to set the Starting Node Number for the interface. The router will AARP for this node. If the number is already in use, the router will choose a new number. For example:

   ```
   AP2 config>set node-number
   Interface #  [0]? 1
   Node number (1-253, or 0 to delete) []? 1
   ```

3. Use the **add zone** command to add one or more zone names for the network attached to the interface. If you define a network range for an interface, you should also define the zone names for the interface. If you did not define a network number, do not define zone names. For example:

   ```
   AP2 config>add zone
   Interface # [0]? 1
   Zone name []? Finance
   ```

After you have specified the parameters, you can use the **list** command at the `AP2 config>` prompt to view your configuration.

Setting Up Zone Filters

Zone filtering lets you filter zones in each direction on each interface. To filter incoming packets, set up an input filter. To filter outgoing packets, set up an output filter. The interface will not readvertise filtered zone information in the direction that you define. Follow these steps to set up a zone filter:

1. Add zone filters to an interface. Use the **add zfilter in** command to add an input zone filter to an interface. Use the **add zfilter out** command to add an output zone filter to an interface. For example:

```
AP2 config>add zfilter in
Interface # [0]? 1
Zone name []? Admin
```

2. Enable the zone filters that you added. This turns on the filter and controls whether the filter is inclusive or exclusive. Inclusive filters forward only the zone information in that filter. Exclusive filters block only the zone information in that filter. For example:

```
AP2 config>enable zfilter in exc
Interface # [0]? 1
```

The following are some examples that explain how to set up zone filters in the internet shown in Figure 26-1.

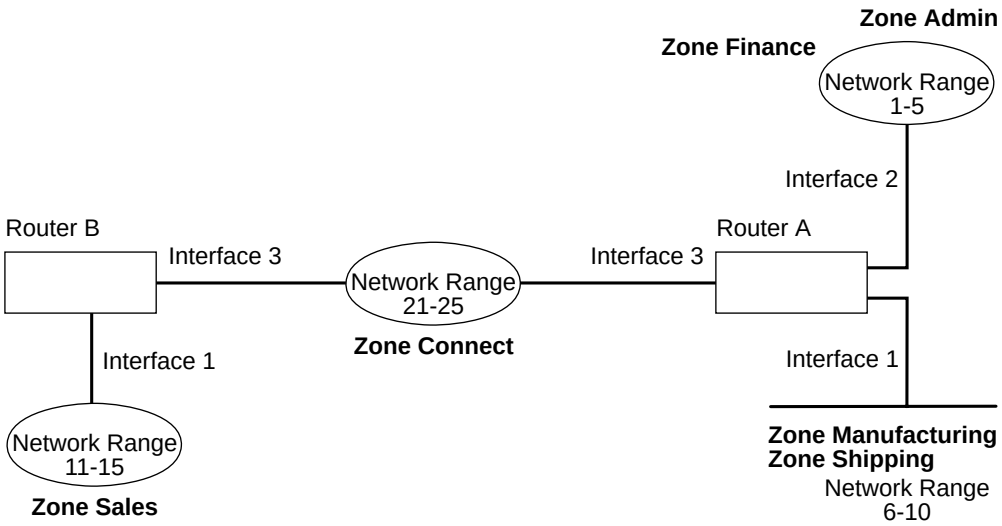


Figure 26-1. Example of Zone Filtering

Example 1

The following is an example of how to filter the Manufacturing zone from all other networks. To do this, you would set up an input filter on Interface 1 of Router A to exclude the Manufacturing zone.

1. On Router A, add an input zone filter to Interface 1.

```
AP2 config>add zfilter in
Interface # [0]? 1
Zone name []? Manufacturing
```

2. Enable the input zone filter and make the filter exclusive.

```
AP2 config>enable zfilter in exc
Interface # [0]? 1
```

This excludes Manufacturing zone information from entering Router A, thereby filtering the zone from the rest of the internet.

Example 2

The following example shows how to filter the Manufacturing zone from Network 11-15, but still allow the Manufacturing zone to be visible on Network 1-5. To do this, you would set up an output filter on Interface 3 of Router A to exclude Manufacturing zone information from being forwarded out of Interface 3. The interface will continue to advertise Manufacturing zone information over interfaces 1 and 2 on Router A, making it visible on Network 1-5.

1. Add an output zone filter to Interface 3.

```
AP2 config>add zfilter out
Interface # [0]? 3
Zone name []? Manufacturing
```

2. Enable the output zone filter and make the filter exclusive.

```
AP2 config>enable zfilter out exc
Interface # [0]? 3
```

   This filter excludes Manufacturing zone information from the output of Interface 3.

Example 3

The next example shows how to set up a filter so that the Admin zone is visible on all networks, but the Finance zone is not visible to the rest of the internet.

1. Add an input zone filter to Interface 2 on Router A.

```
AP2 config>add zfilter in
Interface # [0]? 2
Zone name []? Admin
```

2. Enable the input zone filter and make it inclusive.

```
AP2 config>enable zfilter in inc
Interface # [0]? 2
```

   By setting up this input filter as inclusive, only Admin zone information is forwarded through Interface 2 to the rest of the internet.

Setting Up Network Filters

Network filters are similar to zone filters, except they let you filter an entire network. To set up a network filter:

1. Add a network filter. Use the **add nfilter in** command to add an input network filter to an interface. Use the **add nfilter out** command to add an output network filter to an interface.
   For example:

```
AP2 config>add nfilter out
Interface # [0]? 2
First Network range number (decimal) [0]? 11
Last Network range number (decimal) [0]? 15
```

   The network range you enter here must match the range that you assigned to that network.

2. Enable the network filter that you added and make it either inclusive or exclusive. Inclusive filters forward only network information in that filter. Exclusive filters block only network information in a filter, and they allow all other network information to be forwarded.

```
AP2 config>enable nfilter in exc
Interface # [0]? 2
```

Following are some examples that explain how to set up network filters in the internet shown Figure 26-2 on page 26-7.

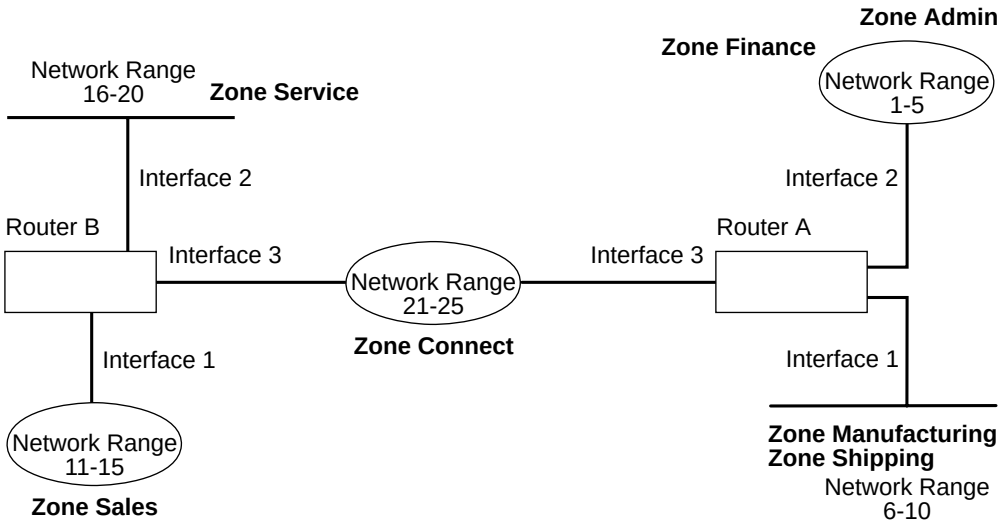


Figure 26-2. Example of Network Filtering.

Note: *Interfaces refer to emulated LAN interfaces, not ATM physical interfaces.*

Example

The following example shows how to filter Network 6-10 so that it is not visible to Network 16-20 as shown in Figure 26-2.

1. Add an output network filter for Network 6-10 to Interface 2 on Router B.

```
AP2 config>add nfilter out
Interface # [0]? 2
First Network range number (decimal) [0]? 6
Last Network range number (decimal) [0]? 10
```

2. Enable the output network filter as exclusive.

```
AP2 config>enable nfilter out exc
Interface # [0]? 2
```

   This filter excludes all information on Network 6-10 from being forwarded through Interface 2 to Network 16-20.

Accessing the AppleTalk Phase 2 Configuration Environment

To access the AppleTalk Phase 2 configuration environment, enter the following command at the `Config>` prompt:

```
Config> ap2
AP2 Protocol user configuration
AP2 Config>
```

AppleTalk Phase 2 Configuration Commands

This section summarizes and then explains the AppleTalk Phase 2 configuration commands.

The AppleTalk Phase 2 configuration commands allow you to specify network parameters for router interfaces that transmit AppleTalk Phase 2 packets. The information you specify with the configuration commands becomes activated when you restart the router.

Enter the AppleTalk Phase 2 configuration commands at the `AP2 config>` prompt. Table 26-1 shows the commands.

Table 26-1. AppleTalk Phase 2 Configuration Commands Summary

Command	Function
? (Help)	Lists the AppleTalk Phase 2 configuration commands or lists the options associated with specific commands.
Add	Adds zone names, network filters, and zone filters to an interface.
Delete	Deletes the zone names, interfaces, network filters, and zone filters.
Disable	Disables interfaces, checksumming, split-horizon routing, network filters, or zone filters, or globally disables AppleTalk Phase 2.
Enable	Enables interfaces, checksumming, split-horizon routing, network filters, zone filters, or globally enables AppleTalk Phase 2.
List	Displays the current AppleTalk Phase 2 configuration.
Set	Sets the cache size, network range, and node number.
Exit	Exits the AppleTalk Phase 2 configuration process and returns to the CONFIG environment.

? (Help)

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

Syntax: ?

Example: **`?`** OR **`disable ?`**

Add

Use the **add** command to add the zone name to the interface zone list or to add the zone name to the interface zone list as the default for the interface or to add network and zone filters.

Syntax: add zone . . .
defaultzone . . .
nfilter in . . .
nfilter out . . .
zfilter in . . .
zfilter out . . .

zone *interface# zonename*

Adds the zone name to the interface zone list. If you define a network number for an interface, you should also define the zone names for the interface. If you did not define a network number, do not define zone names.

Example: add zone

```
Interface # [0]? 0
Zone name []? Finance
```

defaultzone interface# zonename

Adds a default zone name for the interface. If a node on the network requests a zone name that is invalid, the router assigns the default zone name to the node until another zone name is chosen. If you add more than one default to an interface, the last one added overrides the previous default. If you do not add a default, the first zone name added using the **zone** command is the default.

Example: add defaultzone

```
Interface # [0]? 0
Zone name []? Headquarters
```

nfilter in *interface# first network# last network#*

Adds a network filter to the input of the interface. The network range that you enter must match the network range you set for that interface. You cannot filter only a portion of a network range. For example, if you set a network range of 1–10, and you set up a filter for 5–8, the router filters the full network range of 1–10.

Example: add nfilter in

```
Interface # [0]? 0
First Network range number (decimal) [0]? 1
Last Network range number (decimal) [0]? 10
```

nfilter out *interface# first network# last network#*

Adds a network filter to the output of the interface. The network range that you enter must match the network range you set for that interface. You cannot filter only a portion of a network range. For example, if you set a network range of 1–0, and you set up a filter for 5–8, the router filters the full network range of 1–10.

Example: add nfilter out

```
Interface # [0]? 0
First Network range number (decimal) [0]? 11
Last Network range number (decimal) [0]? 20
```

zfilter in *interface# zone name*

Adds a zone name filter to the input or output of the interface.

Example: add zfilter in

```
Interface # [0]? 1
Zone name []? Marketing
```

zfilter out *interface# zone name*

Adds a zone name filter to the output of the interface.

Example: add zfilter out

```
Interface # [0]? 0
Zone name []? Corporate
```

Delete

Use the **delete** command to delete a zone name from the interface zone list, network or zone name filters, or all AppleTalk Phase 2 information from an interface.

Syntax: delete
- zone . . .
- nfilter in . . .
- nfilter out . . .
- zfilter in . . .
- zfilter out . . .
- interface

zone *interface# zonename*

Deletes a zone name from the interface zone list.

Example: **`delete zone 2 newyork`**

nfilter in *interface# first network# last network#*

Deletes a network filter from the input of the interface. You must enter the same network range numbers you set using the **add nfilter in** command.

Example: **`delete nfilter in`**

```
Interface # [0]? 0
First Network range number (decimal) [0]? 1
Last Network range number (decimal) [0]? 12
```

nfilter out *interface#*

Deletes a network filter from the output of the interface. You must enter the same network range numbers you set using the **add nfilter out** command.

Example: **`delete nfilter out`**

```
Interface # [0]? 0
First Network range number (decimal) [0]? 11
Last Network range number (decimal) [0]? 20
```

zfilter in *interface# zone name*

Deletes a zone name filter from the input of the interface.

Example: **`delete zfilter in`**

```
Interface # [0]? 1
Zone name []? Marketing
```

zfilter out *interface# zone name*

Deletes a zone name filter from the output of the interface.

Example: **`delete zfilter out`**

```
Interface # [0]? 1
Zone name []? Marketing
```

interface

Use this command to delete an interface. This is the only way to delete zone names that have non-printing characters.

Example: **`delete interface 1`**

Disable

Use the **disable** command to disable AP2 on all interfaces or on a specified interface, checksumming, filtering, APL/AP2 translation, or split horizon routing.

Syntax: disable ap2
checksum
interface . . .
nfilter in . . .
nfilter out . . .
zfilter in . . .
zfilter out . . .
split-horizon-routing . . .

ap2
Disables the AppleTalk Phase 2 packet forwarder for all interfaces.

Example: `disable ap2`

checksum
Specifies that the router will not compute the checksum in packets it generates. The router usually checksums all packets it forwards. This is the default.

Example: `disable checksum`

interface *interface#*
Disables all AP2 functions on the specified network interface. The network continues to remain available for all other protocols.

Example: `disable interface 2`

nfilter in *interface#*
Disables, but does not delete, the input network filters on this interface.

Example: `disable nfilter in`

```
Interface # [0]? 2
```

nfilter out *interface#*
Disables, but does not delete, the output network filters on this interface.

Example: `disable nfilter out`

```
Interface # [0]? 2
```

zfilter in *interface#*
Disables, but does not delete, the input zone filters on this interface.

Example: `disable zfilter in`

```
Interface # [0]? 1
```

zfilter out *interface#*
Disables, but does not delete, the output zone filters on this interface.

Example: `disable zfilter out 0`

```
Interface # [0]? 1
```

split-horizon-routing *interface#*
Disables split-horizon-routing on this interface. You need to disable split-horizon routing only on Frame Relay interfaces that are on a hub in a partially-meshed Frame Relay network. Disabling split-horizon routing causes all of the routing tables to be propagated on this interface.

Example: `disable split-horizon-routing 0`

Enable

Use the **enable** command to enable the checksum function, to enable a specified interface, to enable AppleTalk 2 gateway function, or to globally enable the AppleTalk Phase 2 protocol.

Syntax: enable ap2
checksum
interface . . .
nfilter in . . .
nfilter oout . . .
split-horizon-routing . . .
zfilter . . .

ap2

Enables the AppleTalk Phase 2 packet forwarder over all of the interfaces.

Example: **`enable ap2`**

checksum

Specifies that the router will compute the checksum in packets it generates. The router checksums all AP2 packets it forwards.

Example: **`enable checksum`**

interface *interface#*

Enables the router to send AppleTalk Phase 2 packets over specific interfaces.

Example: **`enable interface 3`**

nfilter in *exclusive* or *exclusive interface#*

Enables network input filters and controls how the filter is applied to the interface. Inclusive forwards matches. Exclusive drops matches.

Example: **`enable nfilter in inc`**

```
Interface # [0]? 1
```

nfilter out *exclusive* or *exclusive interface#*

Enables network output filters and controls how the filter is applied to the interface. Inclusive forwards matches. Exclusive drops matches.

Example: **`enable nfilter out exc`**

```
Interface # [0]? 1
```

split-horizon-routing *interface #*

Enables split-horizon routing on the interface. The default is *enabled*.

Example: **`enable split-horizon-routing 1`**

zfilter

Enables zone filters assigned to an interface. Must specify if filter is “in” or “out” and if the filter is inclusive or exclusive. Inclusive means that only packets matching the filter will be routed. Exclusive means that all packets matching the filter will be discarded.

Example: **`enable zfilter in inc`**

```
Interface # [0]?
```

Example: **`enable zfilter out exc`**

```
Interface # [0]? 0
```

List

Use the **list** command to display the current AP2 configuration. In the example, the router is a seed router on interface 1 and an unseeded router on interface 2. Interface 2 will learn the network number and zone name from a seed router.

Note: The **list** command accepts an *interface#* as an argument.

Syntax: list

Example: `list`

```
APL2 globally  enabled
Checksumming  disabled
Cache size 500

List of configured interfaces:

Interface       netrange         /  node        Zone
1               10-19            /  52  "EtherTalk", "Sales"(Def)
 Input ZFilters disabled
 Input NFilters (inclusive)
 Output ZFilters disabled
 Output NFilters disabled
 Split-horizon-routing enabled
2               unseeded net     /  0
 Input ZFilters disabled
 Input NFilters (inclusive)
 Output ZFilters disabled
 Output NFilters disabled
 Split-horizon-routing disabled
```

APL2 globally	Indicates whether AppleTalk Phase 2 is globally enabled or disabled.
Checksumming	Indicates whether checksum is enabled or disabled.
Cache size	Number of fastpath cache entries.
List of configured interfaces	Lists each interface number and its network range, node number, and zone name(s) as well as the default zone. For each interface also lists whether or not input and output zone filters and network filters and enabled or disabled. If they are enabled, indicates whether or not they are inclusive or exclusive.
Input/output Zfilters	Indicates zone filters assigned to an interface. Inclusive means that only packets matching the filter will be routed. Exclusive means that all packets matching the filter will be discarded. The name of the zone filtered is displayed. Input means that the filter is applied to traffic coming into the interface. Output means that filter is applied to traffic going out to the interface.
Input/output Nfilters	Indicates net filters assigned to an interface. Inclusive means that only packets matching the filter will be routed. Exclusive means that all packets matching the filter will be discarded. The range of networks filtered is displayed. Input means that the filter is applied to traffic coming into the interface. Output means that filter is applied to traffic going out to the interface.
Split-horizon-routing	Shows whether or not split-horizon routing is enabled or disabled on each interface.

Set

Use the **set** command to define the cache-size of fastpath or specific AppleTalk Phase 2 parameters, including the network range in seed routers and the node number.

Syntax: set cache-size . . .
net-range . . .
node . . .

cache-size *value*

Cache-size corresponds to the total number of AppleTalk nodes that can simultaneously communicate through this router using the fastpath feature. (Fastpath is a method of precalculating MAC headers to forward packets more quickly.) The default is 500, which allows up to 500 nodes to simultaneously communicate through the router and still use fastpath. If the number of nodes becomes greater than the cache size, the router still forwards the packets, but it does not use fastpath. You can set cache size from 100 to 10,000. Although not recommended, setting the cache-size to zero disables the fastpath feature and no memory is used for the cache. You need to change this default only for very large networks. Each cache-size entry uses 36 bytes of memory.

Example: `set cache-size 700`

net-range *interface# first# last#*

Assigns the network range in seed routers using the following:

- interface# - Designates the router interface to operate on.
- first# - Assigns the lowest number of the network range. Legal values are 1 to 65279 (10xFEFF hexadecimal).
- last# - Sets the highest number of the network range. Legal values are *first#* to 65279.

A single numbered network has the same first and last values. A first value of zero deletes the netrange for the interface and turn the “seeded” interface into an “unseeded” interface. First# and last# are inclusive in the network range.

Example: `set Net-Range 2 43 45`

node *interface# node#*

Assigns the starting node number for the router. The router will AARP for this node but if it is already in use, a new node will be chosen. The following explains each argument that is entered after this command:

- interface# - Designates the router interface to operate on.
- node# - Designates the first attempted node number. Legal values are 1 to 253. A node# value of zero deletes the node number for the interface and forces the router to choose one at random.

Example: `set node 2 2`

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 27. Monitoring AppleTalk Phase 2

This chapter describes the AppleTalk Phase 2 (AP2) console commands and includes the following sections:

- "Accessing the AppleTalk Phase 2 Console Environment"
- "AppleTalk Phase 2 Monitoring Commands"

Accessing the AppleTalk Phase 2 Console Environment

To access the AppleTalk Phase 2 console environment, enter the following command at the + (GWCON) prompt:

```
+ protocol ap2
AP2>
```

AppleTalk Phase 2 Monitoring Commands

This section summarizes and then explains the AppleTalk Phase 2 console commands which allow you to view the parameters and statistics of the interfaces and networks that transmit AppleTalk Phase 2 packets. Console commands display configuration values for the physical, frame, and packet levels. You also have the option of viewing the values for all three protocol levels at once.

Enter the AppleTalk Phase 2 console commands at the AP2> prompt. Table 27-1 shows the commands.

Table 27-1. AppleTalk Phase 2 Console Command Summary

Command	Function
? (Help)	Lists all the AppleTalk Phase 2 console commands or lists the options associated with specific commands.
Atecho	Sends echo requests and watches for responses.
Cache	Displays the cache table entries.
Clear Counters	Clears all cache usage counters and packet overflow counters.
Counters	Displays the overflow count of AP2 packets for each interface.
Dump	Displays the current state of the routing table for all networks in the internet and their associated zone names.
Interface	Displays the current addresses of the interfaces.
Exit	Exits the AppleTalk Phase 2 console process and returns to the GWCON environment.

? (Help)

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

Syntax: ?

Example: ?

```
atecho
cache
clear-counters
counters
dump
interface
exit
```

Atecho

The **atecho** command sends AppleTalk Echo Requests to a specified destination and watches for a response. This command can be used to verify basic AppleTalk connectivity and to isolate trouble in the AppleTalk internetwork.

Syntax: atecho *dest_net dest_node*

Example: **atecho 1 27**

dest_net	Specifies the destination AppleTalk network number, in decimal. This is a required parameter.
dest_node	Specifies the destination AppleTalk node number, in decimal. This is a required parameter.

Note: For many AppleTalk nodes, the network address (network number and node number) is dynamically assigned and might not be readily available. However, there are still a number of ways to use the **atecho** command effectively:

1. The AppleTalk address for router nodes is statically configured in many cases. Connectivity between router nodes is critical to overall network connectivity.
2. By setting the atecho destination node number to 255, you can query all nodes on the specified network number on a directly attached AppleTalk network. The received responses will indicate the node's node number. These node numbers can then be used to echo these nodes from distant routers to verify connectivity.

src_net	Source AppleTalk network number. This is an optional parameter. If not specified, the router uses its interface network number on the outgoing interface leading to the destination network.
src_node	Source AppleTalk node number. This is an optional parameter. If not specified, the router uses its interface node number on the outgoing interface leading to the destination network.
size	Number of bytes to use in the AppleTalk echo requests. This is an optional parameter. Default is 56 bytes.
rate	Rate of sending AppleTalk echo requests. This is an optional parameter. Default is one second.

Note: If you enter **atecho** with no parameters, you are prompted for all the parameters. Enter values for the required parameters and either enter values for the optional parameters or accept defaults.

Cache

The **cache** command displays information about the cache-size entries.

Syntax: cache

Example: **cache**

```
Destination  Interface   Usage  Next Hop
122/22        1          1       27/5
138/51        0          1       27/5
23/7          1          1       Direct
```

Destination	AppleTalk node address (network number/node number).
Net	Number of the interface used to forward to the destination node.
Usage	Number of times this cache entry has been used in this aging period, which is five seconds. An unused entry is deleted after 10 seconds.
Next Hop	The AppleTalk address of the next hop router used to forward a packet to the destination node, or Direct if the destination node is directly connected to the interface.

Clear Counters

The clear-counters command clears all cache usage counters and packet overflow counters.

Syntax: clear-counters

Example: **clear-counters**

Counters

Use the **counters** command to display the number of packet overflows on each network that sends and receives AppleTalk Phase 2 packets. This command displays the number of times the AppleTalk Phase 2 forwarder input queue was full when packets were received from the specified network.

Syntax: counters

Example: **counters**

```
AP2 Input Packet Overflows

Net            Count
Eth/0            4
```

Dump

Use the **dump** command to obtain routing table information about the interfaces on the router that forwards AppleTalk Phase 2 packets.

Note: `dump` *interface#* displays the part of the overall network and zone information that is visible on that interface.

Syntax: dump

Example: dump

```
Dest Net    Cost     State   Next hop        Zone
  10-19       0       Dir    0/0       "Ethertalk", "Sales"
  40-49       1      Good    10/13     "Marketing", "CustomerSer",
                                       "TokenTalk"
  20-29       2      Sspct   10/13     "Fuchsia", "Backbone",
                                       "Engineering", "MKTING"

3 entries
```

You can also use the **dump** command with a specific interface to display the routes that are visible on that interface. You can use this feature to make sure filters are configured correctly because it shows whether or not filtered zones or networks are visible to an interface.

Example: dump 0

```
View for interface 0

Dest net   Cost  State   Next hop   Zone
 214-214   1     Good     152/152   "eth-214"
 153-153   0      Dir               "eth153"
 152-152   0      Dir               "ser152"

3 entries
```

Dest Net Specifies the destination network number, in decimal.

Cost Specifies the number of router hops to this destination network.

State Specifies the state of the entry in the routing table. It includes the following:

Dir - indicates that the interface is connected directly to the destination network, the interface is enabled, and the network number is known. (The interface is used to retrieve the routing table.)

Good - indicates that an RTMP packet containing a good tuple for this network was heard in the last 20 seconds.

Suspct - indicates that no RTMP tuple was received for this network in the last 20 seconds.

Bad - indicates that no RTMP tuple was received for this network in the last 40 seconds. RTMP packets with tuples listing this network as unreachable will be sent for 20 seconds, then the network will be deleted from the RTMP routing table. (For more information, refer to the RTMP chapter in *Inside AppleTalk Second Edition* by Gursharan S. Sidhu.)

Next hop Specifies the next hop for packets going to networks that are not directly connected. For directly-connected networks, this is node number 0.

Zone(s) Specifies the human-understandable name for that network. The zone name(s) is enclosed in double quotes in case there are embedded spaces or non-printing characters. If the zone name contains characters beyond the 7-bit ASCII character set (they are 8-bit), the zone name that displays will depend on the characteristics of your console terminal.

Interface

Use the **interface** command to display the addresses of all the interfaces in the router on which AppleTalk Phase 2 is enabled. If the interface is present in the router but is disabled, this command shows that status.

Note: `interface` *interface#* displays the active filtering for that interface. It displays net, node, default zone, and active filters for one interface.

Syntax: interface

Example: **`interface`**

```
Interface     Addresses
  Eth/0       10/52 on net 10-19    default zone  "Sales"
```

You can also enter the interface command followed by a specific interface number to view the AP2 configuration of that interface.

Example: **`interface 1`**

```
Eth/0   1/30 on net 1-5   default zone "marketing"

Input Net filters inclusive   1-5
Output Zone filters inclusive "finance"
Output Net filters exclusive  1-5
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: **`exit`**

Chapter 28. Using and Configuring NHRP

This chapter describes how to use and configure Next Hop Resolution Protocol (NHRP) as specified in Internet Draft Version 10 which has been submitted for RFC status.

This chapter contains the following sections:

Next Hop Resolution Protocol (NHRP) Overview

The Next Hop Resolution Protocol (NHRP) defines a method for a source station to determine the Non-Broadcast Multi-Access (NBMA) address of the "next hop" towards a destination. The NBMA next hop may be the destination itself or the egress router from the NBMA network that is "nearest" to the destination station. This "next hop" information is called a "cut-through" route or VC in the NHRP specification; the MSS uses the term "shortcut" instead of "cut-through." The source station can then establish an NBMA virtual circuit directly with the destination or the egress router and reduce the number of hops through the network.

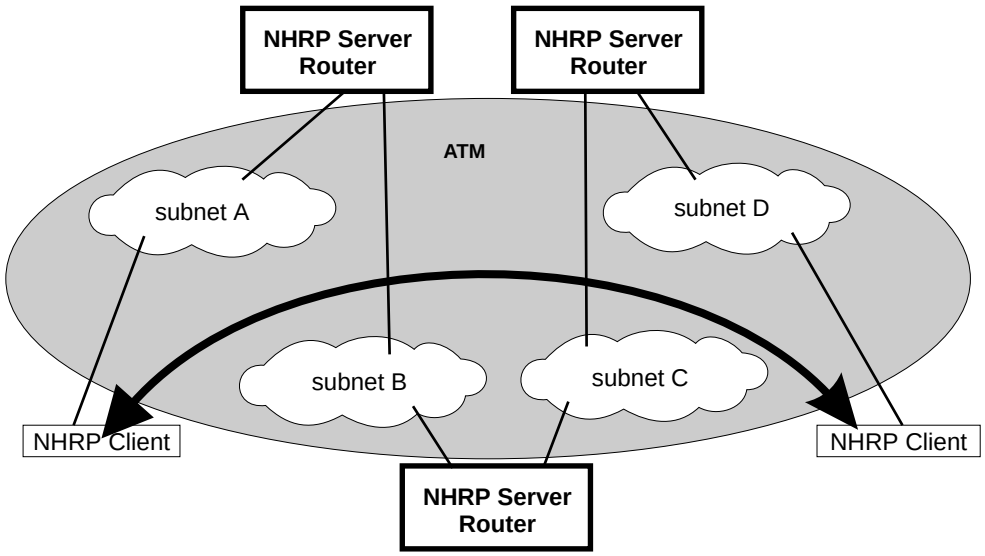


Figure 28-1. Next Hop Resolution Protocol (NHRP) Overview

The MSS Server can use NHRP to establish shortcuts for IP traffic over the ATM NBMA network for both RFC 1483 and Emulated LAN (ELAN) interfaces. The Internet draft does not address the use of NHRP in an ELAN environment, but the MSS Server includes enhancements to allow using LANs. These enhancements are currently implemented using the vendor-private extensions included in the NHRP protocol definition.

The NHRP draft describes the basic protocol flow as follows: NHRP clients register their protocol addresses and their NBMA addresses with one or more

NHRP servers. The servers are typically routers on the routed path through the NBMA network to the clients. When a client wants to establish a shortcut to a destination, it sends a Next Hop Resolution Request packet along the routed path. The request includes the destination protocol address. The routers (that are also NHRP servers) along the routed path first check to see if the destination protocol address is an address that it can serve.

If the router can satisfy the request, the router returns a Next Hop Resolution Reply with the NBMA address of the destination station. The originator can then establish a direct virtual circuit with the destination. If it cannot satisfy the request, the router forwards the request to the next-hop router. This forwarding continues until the request can be satisfied, or it is determined that the destination cannot be reached.

To use client/server terminology, a device may be both a client and a server. The client is the device that originates Next Hop Resolution Requests, and the server is the one that provides Next Hop Resolution Replies with NBMA address information. The MSS Server is such a device; the MSS client conceptually “registers” with the server function in the same machine, although no Registration Requests actually flow. The MSS server also supports NHRP Registrations from remote NHRP clients.

The information provided by clients to their server, and by servers to requestors, must be refreshed periodically and may be purged if conditions dictate. Clients and Servers maintain caches of resolution information that they have sent and received; holding times are used to age out the entries or force refreshes.

Benefits of NHRP and the MSS Implementation

In general, use of NHRP shortcuts can:

- Improve end-to-end performance, by eliminating hops between routers when the source and destination are on the same NBMA network and can communicate directly
- Reduce the load on network routers, since they are bypassed for traffic that, without NHRP, would be handled by the router. This can reduce overall costs as fewer routers or less bandwidth may be needed.

The MSS Implementation of NHRP provides these additional benefits:

- The NHRP draft does not address using the protocol in an Emulated LAN environment. However, the MSS implementation of NHRP includes considerations for such environments; NHRP packets can flow between MSS routers over ELAN connections, and shortcut VCs can be established.
- One-hop Routing: ATM devices that do not support NHRP can be the destination of shortcut paths, eliminating another router hop for traffic, by expanding the definition of the devices that are “served” to include devices that share a protocol subnetwork with the server. For example, all IP addresses on a classical IP subnet that a server is part of, are “served” by that server. The NHRP function interfaces with classical IP 1577 and Lan Emulation components to use their existing ATM address resolution capabilities and apply them to NHRP requests. This enhancement can even be used for traffic to legacy LAN-attached devices that connect to ATM through LAN switches; the NHRP server in the MSS replies to the

client with ATM addressing information for the LAN switch, allowing the client to shortcut directly to that switch. For examples of these “one-hop routing” cases, see Figure 28-1 on page 28-1 and Figure 28-2 on page 28-4

Note: A hop is an operation performed by a traditional router when forwarding packets from one subnet to another. In particular those operations are (1) doing a lookup on a layer 3 subnet identifier (2) determining the outbound “next hop” for the packet (3) stripping and replacing the layer 2 packet header, removing ingress link information and adding egress link information. So, for “one-hop” routing this operation happens once during transfer of a packet from its source to its destination.

- The MSS implementation can operate in networks where some routers do not support NHRP. If the next-hop router is not capable of providing NHRP support, shortcut VCs can be established to the “last” MSS server in the path. See “Disallowed Router-to-Router Shortcuts” on page 28-12 and “Exclude Lists” on page 28-10.
- The customer may configure the MSS Server to establish shortcuts only when traffic to a destination exceeds a given data rate. This can eliminate the creation of VCs for low volume or one-time traffic (for example, SNMP traps). See “data-rate parameter” on page 28-23 and “attempt shortcuts? parameter” on page 28-22.
- MSS provides solutions for the “domino” effect that is described in the NHRP draft. See “attempt shortcuts? parameter” on page 28-22.
- All ATM-attached routers on the routed path should support NHRP for the optimal benefit, although the MSS Server can still operate and provide shortcuts in a mixed network.

Performance Characteristics

NHRP is used during initial contact from a source device to a destination. Once a shortcut VC has been established, NHRP is not involved in actual data transfer. Safeguards ensure that NHRP traffic is not retried for every packet. Also, the MSS implementation provides an option for NHRP shortcuts to be requested only when traffic to a certain destination exceeds a configurable data rate threshold. This can prevent, for example, the establishment of virtual circuits that would only be used for one SNMP trap frame that is generated by an IP host.

NHRP operation does not affect the performance of the MSS router fastpath and will not significantly affect the slowpath. When shortcuts are available, the performance is improved by the elimination of extraneous hops over the ATM network. Also, the performance of intermediate routers that are bypassed by NHRP shortcuts should be improved, as they handle less traffic.

Note: If an MSS configuration does not include a 1577 interface (that is, the MSS is configured only for ELANs), shortcut VCs can be established to the MSS only from clients that support the IBM extensions. This limitation can be avoided simply by defining a 1577 interface on the MSS.

Examples of NHRP Configurations

The following paragraphs give examples of NHRP configurations.

NHRP in an RFC 1577 Classic IP Environment with All Devices NHRP-capable

In this picture, the NHRP clients use RFC 1577 connections to communicate with the router. They use NHRP protocol to learn from the NHRP server about each other's ATM addresses. Then they establish a direct virtual circuit between them for IP traffic.

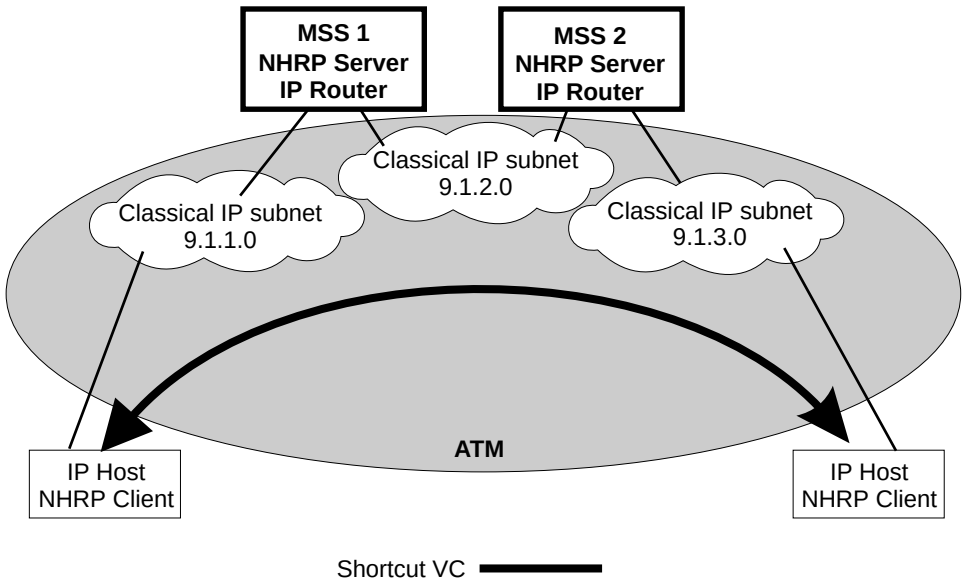


Figure 28-2. NHRP in a Classic IP Environment

NHRP in a Classic IP Environment with non-NHRP Device

This example shows how NHRP can be used between two 1577 devices. when one of them does not support NHRP. Here, MSS2 provides the NHRP client with the ATM address of the non-NHRP device and the client can establish a shortcut for traffic to the non-NHRP host. However, when traffic flows from the non-NHRP device, it flows on the routed path to MSS2; then MSS2 acts as an NHRP client and establishes a shortcut to the destination.

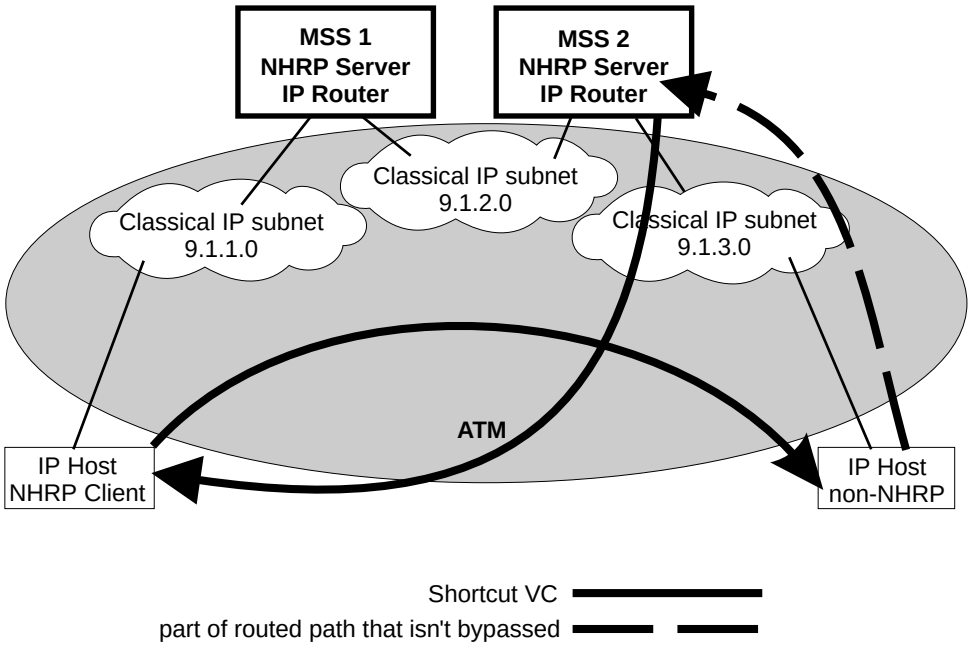


Figure 28-3. NHRP in a Classic IP Environment with non-NHRP Device

NHRP in a Pure LAN Emulation Environment

In the LAN emulation case, routers use the IBM extensions to provide NBMA information for devices on their ELANs. When MSS1 receives traffic from host A destined to host B, it originates a Next Hop Resolution Request and sends it on the routed path. MSS2 replies to the request with NBMA information about host B, one of the stations that it serves because they are on the same ELAN. MSS1 then can establish a data direct VCC to host B even though host B does not participate in or support the NHRP exchanges. Note that this VCC would be used only for traffic in the direction from A to B. Similarly when host B sends traffic to host A, MSS2 generates a Next Hop Resolution Request, MSS1 replies with addressing information about host A, and MSS2 establishes a data direct VCC to A for traffic from B to A.

The LECs in this example are standard-compliant devices with no NHRP support. They must satisfy the LEC requirements described in “NHRP Implementation” on page 28-8).

Nothing special has to be configured in these devices or in the NHRP servers. The NHRP traffic flows over the ELAN subnet with no additional VCs.

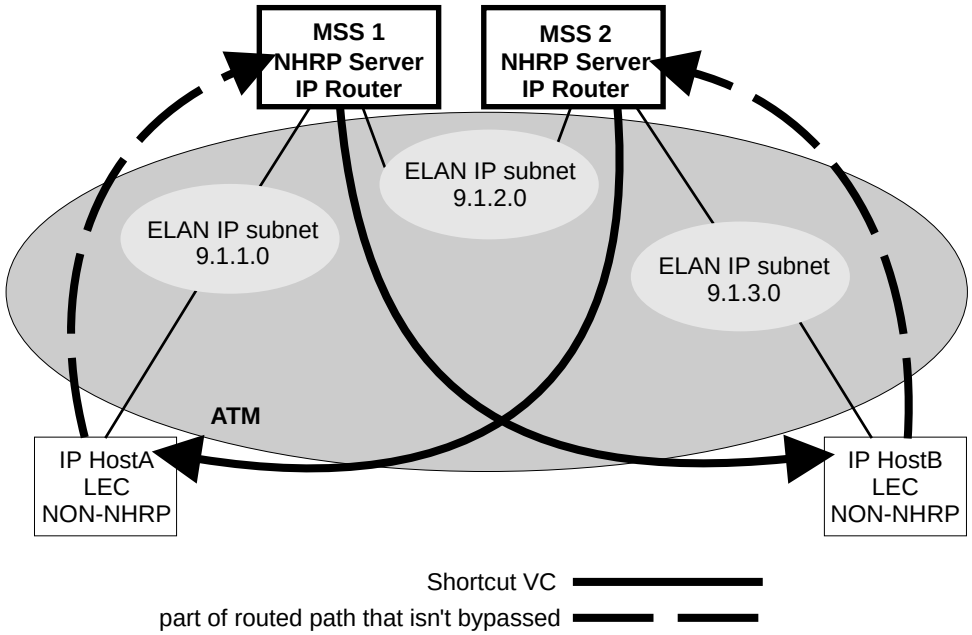


Figure 28-4. NHRP in an ELAN Environment

NHRP in a LAN Emulation Environment with LAN Switches

In this example, the source and destination stations are attached to legacy LANs and do not connect to the ATM net work. LAN switches operating as LAN Emulation Clients give ATM connectivity to the legacy LAN devices. The MSS enhancements to NHRP and the IBM extensions allow the same kind of "one-hop routing" in this environment as described in the previous example. With the enhancements, the servers exchange the actual MAC addresses and routing information for the legacy-LAN devices. The MSS Servers can then establish data direct VCCs with the switches and pass the traffic directly. There is only one router "hop" in the path, although the traffic passes through two LAN switches.

This example also illustrates that the ELAN environment can be token-ring or Ethernet or any mixture of LAN types.

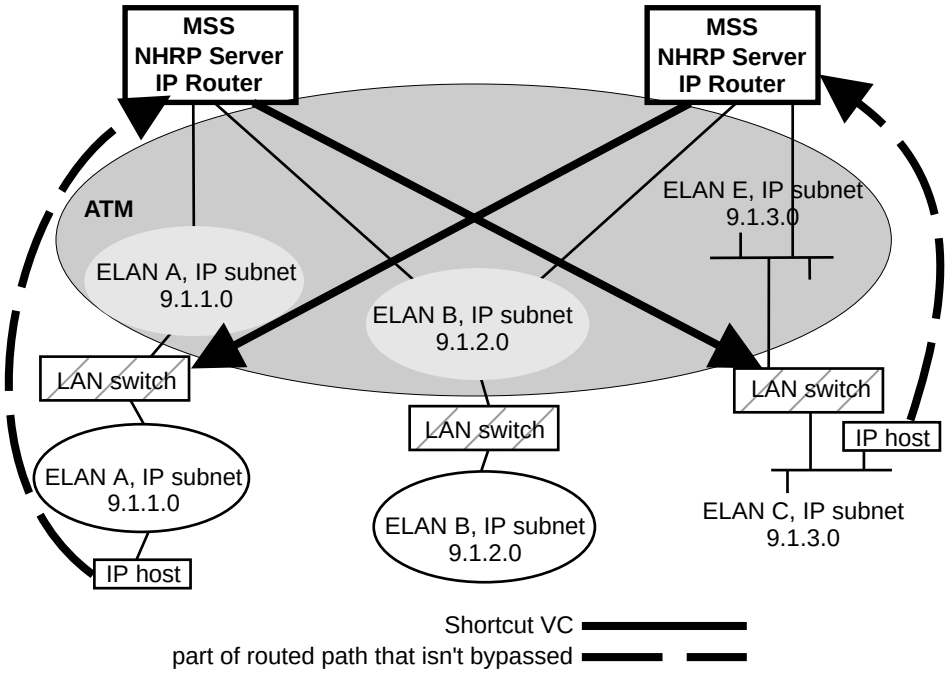


Figure 28-5. NHRP in an ELAN Environment with LAN Switches

NHRP in a Mixed Classical IP and ELAN Environment

The NHRP function in the MSS can operate with both Classic IP and ELAN interfaces in the same network. In this example, the NHRP client supports the IBM extensions and can shortcut directly to the LEC destination for traffic in that direction.

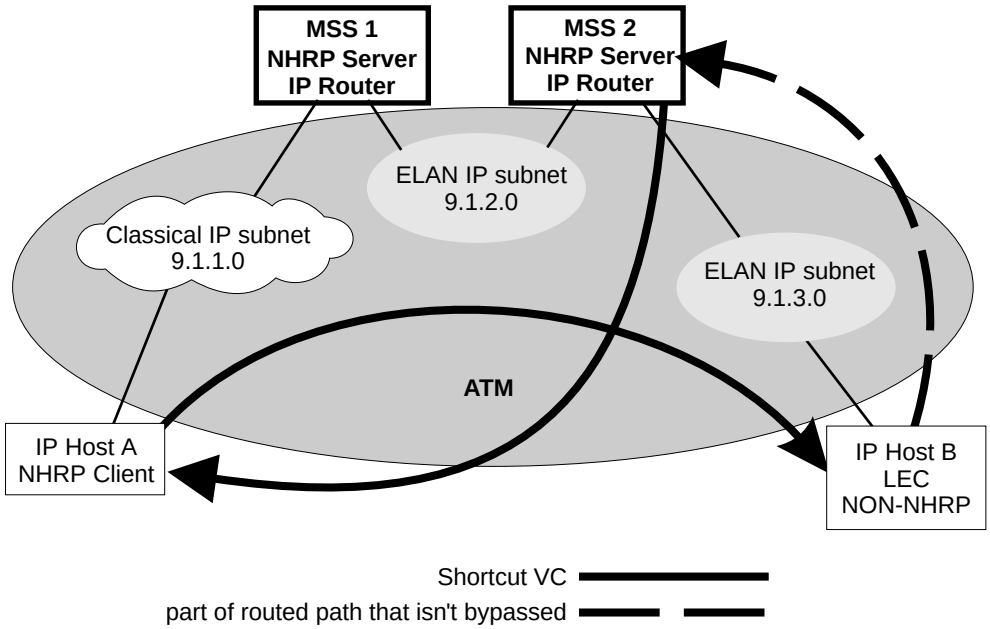


Figure 28-6. NHRP in a Mixed Classical IP and ELAN Environment

NHRP to an Egress Router

The source and/or destination stations of protocol traffic do not have to belong to subnets served by NHRP participants. They may access the ATM network via routers that communicate with the NHRP devices. In this case, the MSS Server provides shortcuts through the ATM network to eliminate as many hops as possible.

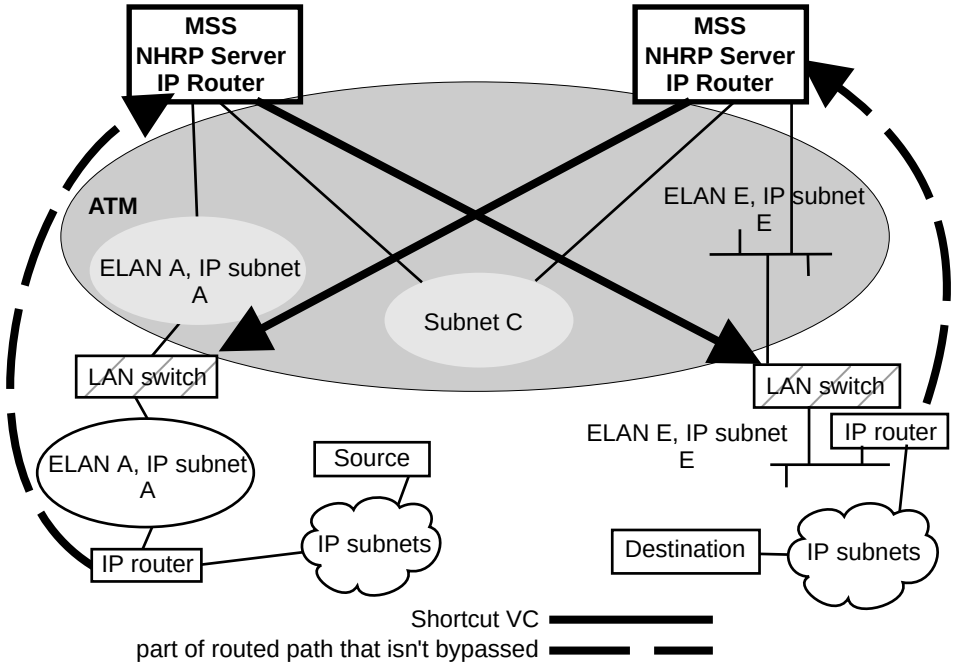


Figure 28-7. NHRP to an Egress Router

NHRP Implementation

NHRP interacts with the router function in the MSS. When the router function in the MSS is forwarding packets along the routed path and NHRP successfully obtains a shortcut VC, NHRP will update the router function to send the packet directly over the shortcut VC.

NHRP updates the routing function's forwarding table after the VC is up. This allows the switch from routed path to the shortcut path to occur without any packet loss.

When an NHRP shortcut is used, the MSS transmits frames to a next hop address on a subnetwork that the MSS itself is not a part of. So the NET, or interface, that provides the outbound path for the traffic is called a “virtual” network interface.

Virtual Network Interface (VNI)

Normally, outbound packet flow from a router is constrained by the following:

- Inability to send packets directly to network addresses that are not defined on a network interface.
- Inability to send packets to network types (for example, token-ring ELAN) unless that network type is defined on a network interface.

The Virtual Network Interface (VNI) net-handler removes all of these constraints, which allows the router to forward packets directly to next hops obtained via NHRP (shortcut routes). It enables one-hop routing, where NHRP shortcut routes can be made directly to devices that do not support NHRP.

The VNI supports token-ring, Ethernet V2 and Ethernet DIX ELAN network interfaces and classic IP network interfaces. When the outbound path is to use a classic IP (1577) interface, the implementation actually uses the existing 1577 net-handler interface for the VNI. However, when the outbound path is to use a LANE shortcut, a unique interface is accessed. This is called the LANE Shortcut Interface (LSI). The LSI is different from a traditional LEC interface because it can provide more than one LAN encapsulation type; that is, one VC may be established using token-ring encapsulation while another uses Ethernet V2. Also the LSI provides connections to more than one Emulated LAN; a traditional LEC interface connects to only one ELAN.

When you enable NHRP, an LSI is created for each ATM adapter. The LSI is assigned the next available interface number, and will be listed when you invoke console functions that display information about the MSS interfaces.

LANE Shortcut Interface (LSI)

The LANE shortcuts provided by the IBM extensions to NHRP are not compatible with some LAN Emulation Client (LEC) and end-station protocol stack implementations. This section describes how these incompatibilities can arise and, in some cases, how they can be overcome using configuration options.

Paranoid LECs are devices that use the LAN Emulation Flush Protocol to verify that clients setting up Data Direct VCCs to it are actually members of its ELAN. These devices will not work with NHRP shortcuts generated by LSIs since the LSI is not part of the target ELAN.

Note: The “Exclude List” configuration option can be used to prevent shortcuts to Paranoid LECs as described in “Exclude Lists” on page 28-10.

By default, the LSI will use the MAC address burned into the associated ATM adapter as the source MAC address of frames transmitted over the LANE shortcut VCCs. It is possible, though unlikely, that this could confuse some end-station protocol stack implementations, since the MAC address will not match that of the router that the end-station uses as a gateway to transmit packets to the associated IP address.

For this to happen, the end-station would have to learn router MAC addresses from unicast IP frames which is not normal (IP-to-MAC address mappings are normally learned from ARP packets). If this were to happen, the end-station might use the learned MAC address as the destination MAC address of frames that it transmits to the associated IP destination instead of using the MAC address of the router. Such frames would either be dropped or forwarded over the LANE shortcut VCC. Forwarding would only occur if the LEC learns MAC-to-ATM address binding from received frames (which is an optional implementation choice).

In either case, these frames will not reach the destination since the LSI discards frames received over a LANE shortcut VCC. Furthermore, the LSI releases the LANE shortcut VCC and no further shortcuts will be established to

the associated ATM address. Traffic for destinations associated with that ATM address will follow the routed path thereafter. Note that ELS messages and console display for LANE shortcuts aid in identifying these destinations.

The LSI can be configured not to use the burned-in MAC address as the source MAC address. With this option, the MAC address of the last-hop router, provided in the NHRP resolution reply packet, is used as the source MAC address. See “Configuring the LANE Shortcuts Interface (LSI)” on page 28-13 for further information and “Change” on page 28-20 for a description of the **change** command.

Using the last-hop router's MAC address as the source MAC address solves the problem of end-station protocol stack confusion but introduces another potential problem. It may confuse LECs that learn MAC-to-ATM address binding from received frames, and therefore should not be used with LECs that perform this type of learning. For example, the LEC in IBM's 8281 ATM-LAN bridge performs this type of learning. Nevertheless, this configuration option is provided to maximize flexibility in achieving compatibility with the largest possible set of destinations in a given installation.

Configuration Parameters

This section describes some of the NHRP related configuration parameters and their recommended usage. See “NHRP Configuration Commands” on page 28-15 for command syntax, command parameters, valid values and default values.

NHRP Minimum Configuration

NHRP can be enabled by entering the **enable NHRP** command from the `NHRP config>` prompt. This enables NHRP for the MSS using default configuration options. NHRP can be disabled by entering the **disable NHRP** command from the `NHRP config>` prompt. See“Accessing the NHRP Configuration Process” on page 28-15.

Exclude Lists

Configuration allows you to create a list of protocol addresses (and associated masks) that represent two types of devices:

- Next-hop routers that do not contain an NHRP server function
- Destination devices to which shortcut VCs should not be allowed

Next-hop Routers: The exclude list can be used to identify routers that are on the routed path but do not support NHRP server function.

The MSS Server responds to a Next Hop Resolution Request by providing the ATM address of the next-hop router when all of the following are true:

- The next-hop address is different from the destination address.
- The MSS interface to the next-hop router is either an ATM classical IP or an ELAN subnet.
- The next-hop address is in the exclude list.

In processing the request, the MSS does not forward the Resolution Request on to the next-hop address, but responds to the client with addressing information that allows the client to establish a shortcut VC to the next-hop router.

Note: If the next-hop router is one of the Disallowed R2R Shortcuts, the MSS sends a NAK to the Resolution Request instead of a positive reply.

In general, if the next-hop router is on the exclude list, the MSS does not send it any NHRP packets that would only be handled by an NHRP server.

Destination Devices: The exclude list can also be used to prevent shortcut VCs to a given protocol address (for example, a device on a CIP or ELAN subnet that can support only a small number of VCs).

When processing a Next Hop Resolution Request for a destination device, the MSS Server responds to the client with addressing information that allows the client to establish a shortcut VC to the MSS itself when all of the following are true:

- The next-hop address equals the destination address.
- The MSS interface to the destination is either an ATM classical IP or an ELAN subnet.
- The destination address is in the exclude list.

Extensions

The NHRP protocol includes **Extensions**. Extensions are appended to NHRP packets. Extensions are used to request additional functions from the NHRP participants. The use of the **extensions** parameter lets you determine if the MSS sends certain extensions:

- path information extensions
- IBM vendor-private extensions

Path Information Extensions: Three extensions are defined in NHRP to provide path information. These extensions can be used to help monitor the request itself, to determine the path taken by the request, to determine who generated the reply, and the path taken by the reply. The path information extensions are:

- Forward Transmit - Each Next-Hop Server (NHS) that forwards the request along the way should append information about itself.
- Responder Address - The Next-Hop Server (NHS) that generates the reply should append information about itself.
- Reverse Transmit - Each Next-Hop Server (NHS) that forwards the reply along the way should append information about itself.

The router can be configured to send any or all of these extensions in Next Hop Resolution Request packets that it generates. The information received in the reply packets is displayed in the router's NHRP ELS messages.

IBM Vendor-Private Extensions: To support NHRP in an Emulated LAN environment, the MSS Server adds vendor-unique extensions to NHRP packets. Three of these extensions act as “queries”; the NHRP client places them in the Next Hop Resolution Request. If the server supports this function, it responds with three corresponding extensions containing ELAN address information (MAC address, ATM address and Routing information); these extensions are included in the Next Hop Resolution Reply.

The router can be configured so that it does not support the IBM-specific extensions. If the IBM specific extensions are not used, shortcuts directly to

ELAN devices are not possible. Use the “Exclude List” option to disallow shortcuts selectively to certain ELAN devices.

Disallowed Router-to-Router Shortcuts

Operation of NHRP may result in establishing transit paths across NBMA network between routers. However, establishing an NHRP shortcut across a boundary where information used in route selection is lost may result in a routing loop. Such situations include the loss of BGP path vector information, and the interworking of multiple routing protocols with dissimilar metrics. Under such circumstances, NHRP shortcuts between routers should be disallowed. This situation can be avoided if there are no “back door” paths between the entry and egress router outside the NBMA network.

The MSS Server allows router-to-router (R2R) shortcuts by default. However, by configuring disallowed R2R shortcuts, you can create a list of destination or router addresses for which the MSS does not allow shortcuts.

To create a disallowed R2R shortcut, you must specify both a protocol address and a mask. The protocol address is either the destination or a router, and the mask allows for a range of addresses.

To illustrate how to specify disallowed R2R shortcuts using protocol addresses and masks, consider the following network diagram:

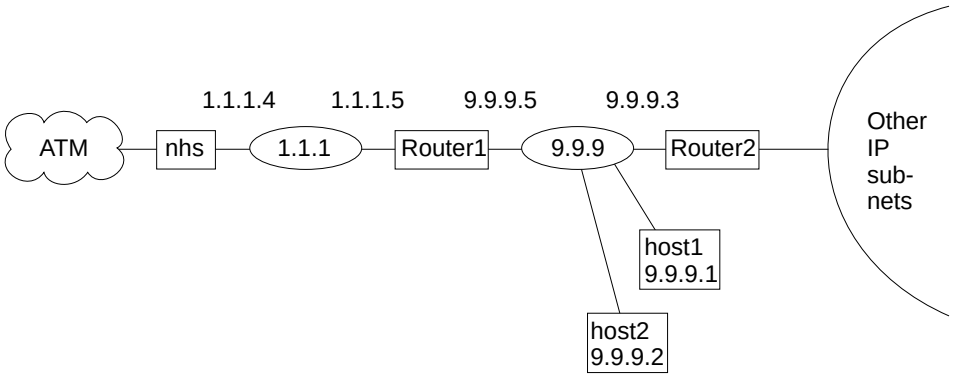


Figure 28-8. Using Disallowed Router-to-Router Shortcuts

Example 1: An entry with *address=9.9.9.1 mask=255.255.255.255* would cause the NHS to send a NAK to the sender of a Next Hop Resolution Request with destination protocol address 9.9.9.1 (HOST1). Since 9.9.9.1 is not directly attached to one of the MSS subnets, but is reached by another router, the MSS checks the Disallowed R2R Shortcuts List.

Example 2: An entry with *address=9.9.9.0 mask=255.255.255.0* would cause the MSS to send a NAK for any destination address 9.9.9.1 through 9.9.9.255. HOST1, HOST2, and ROUTER2 could not be reached using shortcuts to the MSS but devices on the other subnets serviced by ROUTER2 could be reached.

Example 3: An entry with *address=1.1.1.5 mask=255.255.255.255* would cause the MSS to respond negatively for any destination whose next-hop router is 1.1.1.5, ROUTER1. The MSS would respond negatively for any address on subnet 9.9.9 and for any address on the other IP subnets reached via router 9.9.9.3 because next hop is 1.1.1.5.

Example 4: An entry with *address=anything mask=0.0.0.0* would disable R2R shortcuts for all addresses.

Protocol Access Control Usage

This parameter determines how the protocol layer access controls will be applied to the NHRP packets. NHRP implementation checks the IP filters, packet filters, and access controls when requesting, forwarding, and providing shortcut routes.

The configuration default is “Source and Destination”: when the NHRP requester is not a router, the NHRP client's IP address is assumed to be the source of all IP packets that will be transmitted by that client using the NHRP shortcut route. The MSS denies NHRP shortcut requests from a non-router NHRP client if any IP packets are being filtered for that IP destination/source address pair, where the source is the NHRP client's address. Selecting the “Destination only” option causes the MSS to deny shortcut requests from any NHRP client if any IP packets are being filtered to the destination address. If NHRP clients should not be trusted, “Destination only” should be selected. “Destination only” might be the best option when NHRP clients are non-routers with multiple IP addresses or non-router clients that transmit packet that originate from other sources.

NHRP clients that reside in the routers use the NHRP shortcut routes to forward packets from other sources: therefore, if “Source and Destination” is configured and the MSS receives a shortcut request from a router, the MSS applies the IP filters the same way as when “Destination only” is selected.

ATM Network ID

Since an MSS server may have more than one ATM adapter, it may be connected to two different or unassociated networks. This must be considered when deciding when shortcut VCs should be allowed.

You can determine which interfaces should be treated as if they are connected to the same physical ATM network by assigning each ATM interface a Network-ID by using the **set** command at the `ATM Interface Config>` prompt as described in the “Using and Configuring ATM” chapter in *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

ATM interfaces with the same Network-ID are considered to belong to the same network. By default, all ATM interfaces are assigned to Network-ID 0.

Configuring the LANE Shortcuts Interface (LSI)

The NHRP LANE Shortcut Interface (LSI) is automatically created for each ATM adapter when NHRP is enabled for the MSS. The LSI uses default values for the following parameters.

- ESI
- Selector
- Use Best Effort Service for Data VCCs
- Peak Cell Rate of outbound Data VCCs
- Sustained Cell Rate of outbound VCCs
- Use ATM adapter's burned-in MAC address for source

The default values may be modified using the **change** command from the `NHRP Advanced config>` prompt. See “Change” on page 28-20.

Configuring non-MSS Devices in an ATM Network

If you have a non-MSS NHRP client/server and its configuration requires you to give the ATM address of the MSS NHRP server,you must select the proper ATM address. You must use an address associated with an "ATM interface" in the MSS, and an IP address must be assigned to this interface. The last two digits of the MSS ATM address, the selector, are assigned dynamically after the MSS is activated (and may change if the configuration of the MSS changes), unless you have configured a specific selector.

You can specify the ATM address, including selector, by entering **prot arp** at the talk 6 Config> prompt, followed by **add atm**, giving the desired IP address and then specifying a selector. This is the same procedure used to define an ATMARP client.

Accessing the NHRP Configuration Process

To access the NHRP configuration:

1. At the operator console prompt (*) type `talk 6` and press enter.
2. At the `config>` prompt type `protocol nhrp` and press enter.
3. The `NHRP config>` prompt is displayed.

NHRP Configuration Commands

This section explains all of the NHRP configuration commands as shown in Table 28-1. Enter the commands at the `NHRP config>` prompt.

Table 28-1. NHRP Configuration Command Summary

Command	Function
? (Help)	Displays all the NHRP commands or lists subcommand options for specific commands.
Enable NHRP	Turns on NHRP for all interfaces that are not explicitly defined.
Disable NHRP	Turns off NHRP for all interfaces that are not explicitly defined.
List	Displays the NHRP configuration.
Advanced config	Gets you to the `NHRP Advanced config>` prompt, from which you can enter other commands as described in "NHRP Advanced Configuration Commands" on page 28-17.
Exit	Exits the NHRP configuration process.

? (Help)

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

Syntax: ?
Example: ?

```
enable nhrp
disable nhrp
list
advanced config
exit
```

Enable NHRP

Use the `enable` command to enable NHRP on all interfaces not explicity defined using an NHRP advanced config command. It is a simple way to get NHRP up and running with default parameters.

Syntax: enable NHRP.
Example: enable

Disable NHRP

Use the `disable` command to disable NHRP on all interfaces not explicitly defined using an NHRP advanced config command.

Syntax: disable NHRP
Example: disable

```
Disable NHRP for the MSS [No]:
```

Advanced Config

Use the **advanced** command to get to the NHRP advanced configuration prompt, `NHRP Advanced config>` . From this prompt, you can enter the commands described in “NHRP Advanced Configuration Commands” on page 28-17.

Syntax: advanced NHRP
Example:

```
NHRP config> advanced
NHRP Advanced config>
```

Note: Most installations will not need to use this “advanced” command. The **enable NHRP** command is sufficient to enable NHRP with recommended default options.

List

Use the **list** command to list the NHRP configuration.

Syntax: list
Example: list

```
Box level NHRP enabled
Explicit interface definitions override box level settings

Interfaces explicitly defined for NHRP
--------------------------------------
Interface 1: ELAN
    NHRP enabled

NHRP LANE Shortcut Interface:
-------------------------------------------------
Interface: 3  ESI: burned-in          Sel: auto
Use Best Effort: yes  (Data)
Cell Rate(kbps):  Peak:      0/     0    Sustained:   1000/538767240
ATM adapter's burned-in MAC address is used as source address


NHRP IP exclude list is empty.

No disallowed router-to-router shortcuts defined.

Holding time is 20 minutes.
Data-rate threshold is 10 packets/second.
Resolution cache size is 1028 entries.
Server purge cache size is 512 entries.
Server registrations cache size is 512 entries.
Protocol access controls use source and destination address.
NHC attempts shortcuts based on data-rate.
NHS allows shortcuts to ATMARP clients.
Use NHRP Forward transit NHS record client extension: No
Use NHRP Reverse transit NHS record client extension: No
Use Responder Address client extension:               No
Use LANE shortcuts extension:                         Yes
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit
Example: exit

NHRP Advanced Configuration Commands

This section explains all of the NHRP advanced configuration commands as shown in Table 28-2 on page 28-18. Enter the commands from the `NHRP Advanced config>` prompt.

Table 28-2. NHRP Advanced Configuration Command Summary

Command	Function
? (Help)	Displays all the NHRP commands or lists subcommand options for specific commands.
Add	Adds an NHRP interface, an exclude list, or disallowed R2R shortcuts.
Change	Changes an NHRP interface, or changes a LANE shortcut interface definition.
Delete	Deletes an NHRP interface, an exclude list, or disallowed R2R shortcuts.
List	Displays the NHRP configuration.
Set	Sets NHRP parameters.
Exit	Exits the NHRP advanced configuration process.

? (Help)

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

Syntax: ?
Example: ?

```
add
delete
change
list
set
exit
```

Add

Use the **add** command to add an explicit interface definition, an exclude list entry, or disallowed router-to-router shortcuts.

Syntax: add interface definition
exclude list
disallowed router-to-router shortcuts

interface definition

Adds an explicit interface definition to either enable or disable an NHRP interface.

If NHRP is disabled on a particular network interface, NHRP packets are not forwarded to any routers that are reached via that interface. Also, incoming NHRP frames are discarded.

Note: Any explicit interface definitions override the “NHRP enabled/disabled” box-level setting.

Example: add int

```
Interface Number [0]?
Enable NHRP [Yes]:
```

exclude list

Adds an exclude list entry. Specify a protocol address which must be excluded from the NHRP network. See “Exclude Lists” on page 28-10 for more information.

Valid values: IP address and mask.

Default: Empty.

Example: add exc

```
IP Address [0.0.0.0]? 6.6.6.5
Address Mask [255.255.255.255]?
```

disallowed router-to-router shortcuts

Adds a router protocol address to which shortcuts are not allowed.

See “Disallowed Router-to-Router Shortcuts” on page 28-12 for more information.

Example: add dis

```
IP address [0.0.0.0]?  8.8.8.1
Address Mask [255.255.255.255]?
```

Valid values: IP address and mask.

Default: Empty.

Delete

Use the **delete** command to delete an interface definition for NHRP, an exclude list entry, or disallowed router-to-router shortcuts.

Syntax: delete interface definition for NHRP
exclude list
disallowed router-to-router shortcuts

interface definition for NHRP

Deletes an explicit NHRP interface definition.

Example: del int

```
Interface Number [0]?
```

exclude list

Deletes an exclude list entry. You must specify an index which must be deleted. Use the **list exclude** command to determine the right index.

Example: del exc

```
Enter index of access control to be deleted [1]?
```

disallowed router-to-router shortcuts

Deletes a disallowed router-to-router shortcuts entry. You must specify an index to be deleted. Use the **list disallowed** command to determine the right index.

Example: del dis

```
Disallowed shortcuts index [1]?
```

Change

Use the **change** command to modify NHRP interface definitions.

Syntax: change interface definition
nhrp lane shortcut interface

interface definition for NHRP

Change an explicit interface definition to either enable or disable an NHRP interface.

Example: ch int

```
Interface Number [0]?
Enable NHRP [Yes]:
```

NHRP LANE shortcut Interface

Change a LANE shortcut interface definition.

Example: ch nhrp

```
Interface Number of NHRP LANE Shortcut Interface [0]?
(  1) Use burned in ESI
Select ESI [1]?
Use internally assigned selector? [Yes]:
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]?
Use ATM adapter's burned-in MAC address for source?
```

Interface Number of NHRP LANE Shortcut Interface

Use the interface number assigned to the LSI. The interface number can be determined by using the **list interface** command.

(1) Use burned in ESI

Use burned-in ESI as part of the ATM address. You may be given other choices depending upon your configuration.

Select ESI

Specify the ESI.

Use internally assigned selector

Use internally assigned selector or assign a selector in the range 00 to FF.

Use Best Effort Service for Data VCCs

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.

Sustained Cell Rate of outbound Data VCCs (Kbps)

Specifies the Sustained Cell Rate (SCR) traffic parameter for the Data VCCs.

Use ATM adapter's burned-in MAC address for source?
Use the adapter's burned-in MAC address or the one supplied in the NHRP resolution reply, as the source for LANE shortcuts.

See "ATM and LAN Emulation" in *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1* for further information.

Note: It is recommended that you use the default values until you have determined the specific processing options required by your environment.

List

Use the **list** command to display the NHRP configuration information.

Syntax: list all
exclude list
disallowed router-to-router shortcuts
interface definitions
cache size

all
Displays the entire NHRP configuration.

Example: li all

Output is the same as for the **list** command. See "List" on page 28-16.

exclude list
Displays the exclude list entries.

Example: li exc

```
List of NHRP IP exclude records
-------------------------------
  # Address         Mask
  1 7.7.7.7         255.255.255.255
```

disallowed router-to-router shortcuts
Displays disallowed router-to-router shortcuts.

Example: li dis

```
Disallowed router-to-router shortcuts for IP
--------------------------------------------
  1 8.8.8.1         255.255.255.255
  2 6.6.6.1         255.255.255.255
```

interface definitions
Displays the NHRP interface definitions.

Example: li int

```
Interfaces explicitly defined for NHRP
--------------------------------------
   None

NHRP LANE Shortcut Interface:
-------------------------------------------------
Interface: 3  ESI: burned-in          Sel: auto
Use Best Effort: yes  (Data)
Cell Rate(kbps):  Peak:      0/     0    Sustained:   1000/538764944
MAC address supplied by NHS is used as source address
```

cache size
: Displays cache sizes.

Example: li ca

```
Resolution cache size is 1024 entries.
Server purge cache size not configured, default is 512 entries.
Server registrations cache size is 256 entries.
```

Set

Use the **set** command for the following:

Syntax: set
- protocol access control usage
- attempt shortcuts?
- holding time
- data-rate threshold
- extensions ...
- cache size ...
- shortcuts to atmarp clients

protocol access control usage
: Determines whether both IP *source and destination* or only the IP *destination* address is used to check against the configured IP access controls. See “Protocol Access Control Usage” on page 28-13 for more information.

Example: set prot

```
(Destination, Source & Destination) [Destination]?
```

attempt shortcuts?
: Determines how the NHRP client decides when to originate resolution requests.

  Valid values: Y, N, Data-rate.

  Y Yes. Always try to establish a shortcut VC by building a Next Hop Resolution Request and sending it to the next hop station.

  N No. Never try to establish a shortcut. Using this option essentially disables the client function in the router. This setting might be used in an intermediate router (one that is not an entry point into the NBMA network for routed traffic) to eliminate the “domino effect,” where traffic following the routed path triggers NHRP Resolution Requests at each NHRP router along the path.

Data-rate Try to establish a shortcut only after the datarate threshold is reached.

Note: This setting can prevent the creation of VCCs for "one-time" traffic, such as SNMP traps that are sent to an SNMP manager.

Default: Data-rate.

Example: set attempt

```
Try shortcut VCs? (Yes, No, Data-rate) [Data-rate]?
```

holding time

Sets the holding time in minutes.

The holding time parameter is used for these functions:

- When the MSS responds to a Next Hop Resolution Request with information about itself (that is, the MSS is to become the next hop shortcut), the holding time is sent to the requestor as the length of time that the information can be considered valid.
- When the MSS responds to a Next Hop Resolution Request with information about another NBMA station that was not learned using NHRP (for example, the destination station is an ATM device with an IP address on one of the MSS subnets), the holding time is sent to the requestor as the length of time that the information can be considered valid.

Valid values: 1 - 60 minutes.

Default: 20 minutes.

Example: set hold

```
Holding time (in minutes)  [20]?
```

data-rate threshold

Sets the data rate threshold in packets/second.

The datarate threshold is used when the **attempt shortcuts** parameter is set to **Data-rate**.

When traffic is destined for a particular station, but the rate is less than this threshold, then the MSS does not attempt to establish shortcuts. (In other words, it does not generate Next Hop Resolution Requests and send them to the next hop along the routed path.) Once the traffic rate exceeds the threshold, the MSS tries to establish a shortcut. If it can successfully create a shortcut path, the path is used even if the traffic drops below the threshold. The path continues to be used until the traffic stops for a period of time. This is done to avoid going back and forth from the routed path to the shortcut path if traffic is sporadic.

Valid values: Minimum 1 packet/second. Maximum is 5120 packets/second.

Default: 10 packets/second.

Example: set data

```
Data-rate threshold in packets/second [10]?
```

extensions
Sets the selected NHRP extension usage to *YES* or *NO*.

Forward transmit NHS (default NO)
Reverse transmit NHS (default NO)
Responder Address (default NO)
Lane Shortcuts (default YES)

Valid Values: YES or **NO**

Example: set ext lane

```
Use LANE shortcuts extension [Yes]?
```

cache size *resolution* OR *registration* OR *server purge*
Sets the selected cache's maximum entries.

Cache sizes can be selected for any of the following:

resolution cache
This parameter lets you determine the number of entries in the cache for client functions. Each cache entry contains the protocol address-to-NBMA address mapping that can be used to create shortcut VCs. Entries are in the cache when the MSS has:

- Successfully resolved a protocol address to an NBMA address by sending Next Hop Resolution Requests.
- Attempted to resolve a protocol request to an NBMA address but has either not received a reply, or has received a negative reply, and the associated timer has not expired. These entries are kept in the cache to prevent the MSS from generating additional Next Hop Resolution Requests for some period of time.
- Received a registration request from a client and the holding time indicated in that request has not yet expired.

When the cache size is exceeded, no new attempts are made to resolve protocol addresses to NBMA addresses (in other words, no new Next Hop Resolution Requests are sent) until existing entries are purged, either because the holding time has expired or a specific purge request has been received from the originator of the information. Also, when cache size is exceeded, Registration Requests from new clients are rejected.

Valid values: 256 - 65535 entries.

Default: 512 entries.

Example: set cache res

```
Number of cache entries [512]?
```

registration cache
Sets a limit on the number of registration entries in the resolution cache. When the server receives a registration request, it checks to see if the number of NHRP client registrations is below this limit before adding a registration entry in the resolution cache.

Valid values: 256 - 16384 entries.

Default: 512 entries.

Example: set cache reg

```
Number of cache entries [512]?
```

server purge cache

This parameter lets you determine the number of entries in the server purge cache. An entry in this cache represents a destination protocol address and a client to which the server has provided Authoritative NBMA information for that destination.

The destination address may represent the server itself, devices on subnetworks to which the server is attached, NHRP clients that have registered with the server, or routers for which a R2R shortcut has been advertised. The MSS uses the information in these cache entries to notify clients to purge address information that becomes invalid before the holding time expires.

When the server purge cache size is exceeded, the server rejects Authoritative Next Hop Resolution Requests.

Valid values: 256 - 65535 entries.

Default: 512 entries.

Example: set cache serv

```
Number of cache entries [512]?
```

shortcuts to ATMARP clients

Allows or disallows shortcuts to ATMARP clients.

This parameter can be used to allow or disallow the MSS server from giving out shortcuts to native ATMARP clients that do not support NHRP. This may be required if these clients are not capable of supporting large number of VCs. Use the “Exclude List” option if shortcuts need to be disallowed selectively to certain clients or subnets.

Example: set shortcut

```
Allow shortcuts to Classical IP clients?  [Yes]:
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: exit

Chapter 29. Monitoring NHRP

This chapter describes how to monitor Next Hop Resolution Protocol (NHRP). For a description of this protocol, refer to “Next Hop Resolution Protocol (NHRP) Overview” on page 28-1.

This chapter contains the following sections:

- “Accessing the NHRP Console Process”
- “NHRP Console Commands”
- “NHRP Packet Tracing” on page 29-7

Accessing the NHRP Console Process

You can access the NHRP console commands from Talk 5. After entering the talk 5 command, the operator console prompt (+) is displayed. At the console prompt enter `protocol nhrp`. This will get you to the NHRP console prompt `NHRP>` .

For example:

```
* talk 5
+ protocol nhrp
  NHRP>
```

NHRP Console Commands

This section summarizes and then explains all of the NHRP console commands as shown in Table 29-1. Enter the commands from the `NHRP>` prompt.

Table 29-1. NHRP Console Command Summary

Command	Function
? (Help)	Displays all the NHRP commands or lists subcommand options for specific commands.
Box Status	Displays NHRP enable/disable status.
Interface Status	Displays NHRP interface status.
Statistics	Displays NHRP interface statistics.
Cache	Displays NHRP resolution cache entries.
MIB	Displays MIB information.
LANE Shortcuts	Displays LANE shortcut entries.
CONFIG Parameters	Displays NHRP configuration information.
Exit	Exits the NHRP console process.

? (Help)

Syntax: ?
Example: **?**

```
box status
interface status
statistics
cache
mib
LANE shortcuts
config parameters
exit
```

Box Status

Use the **box status** command to display NHRP status as configured for the box (for example, all interfaces not explicitly defined).

Syntax: box-status

Example: box status

```
Box level NHRP is ON by config
```

Interface Status

Use the **interface status** command to display NHRP status on interfaces.

Syntax: interface-status

Example: interface status

```
Interface 0: UP  (NHRP enabled)
Interface 1: UP  (NHRP disabled)
Interface 2: DOWN
Interface 3: UP  (NHRP LANE Shortcut Interface)
```

Statistics

Use the **statistics** command to display NHRP statistics for all interfaces or for a specific interface.

Syntax: statistics all
interface

all — Lists NHRP statistics on all interfaces.

Example: statistics all

Output is the same as that for the **statistics interface** command as shown in the following example.

interface — Lists NHRP statistics on a specified interface.

Example: statistics interface

```
Interface number [0]? 0

Statistics for Interface 0
--------------------------
Field Description                          Value
-----------------                          -----
Inbound Requests                            5
Outbound Requests                           3
Inbound Replies                             3
Outbound Replies                            5
Inbound Registers                           0
Outbound Registers                          0
Inbound Error Packets                       0
Inbound Error Indication Packets            0
Outbound Error Indication Packets           0
Reply Forwards                              0
Unrecognized Options                        0
Registration Overflows                      0
ProtocolErrors                              0
Negative Outbound Replies                   0
Inbound Packets on NHRP disabled interface  0
'Send_to_me' Outbound Replies               0
Inbound Purges                              0
Outbound Purges                             2
```

Cache

Use the **cache** command to display all NHRP resolution cache entries or a specific cache entry identified by a destination address.

Syntax: cache list
entry

list — Lists NHRP cache entries.

Example: cache list

```
Total Client Cache Entries = 3

NHRP Client Cache Entries
=========================

Dest Address    NextHop Address State Htime MTU  Net
------------    --------------- ----- ----- ---  ---
5.5.5.1         5.5.5.1         Act   1121  4490  1
5.5.5.2         5.5.5.2         Inact 1185  4490  1
6.6.6.1         6.6.6.1         Act   602   9180  0
```

entry — Lists a specific NHRP cache entry.

Example: cache entry

```
Enter destination address  [0.0.0.0]? 6.6.6.1
Destination: 6.6.6.1
NextHop:     6.6.6.1
ATM Address: 39840F000000000000000000000410005A00DEADCA
State:       Act
Net:         0
HoldingTime: 433 seconds
MTU size:    9180
Flags:       0x00420000
```

MIB

Use the **MIB** command to display NHRP MIB related information.

Syntax: mib list ...
entry ...

list — Lists NHRP mib entries for:

- Server table
- Client table
- Next-Hop Server (NHS) statistics table
- Next-Hop Client (NHC) statistics table
- Resolution cache table

Example:

```
NHRP> mib list server table
MIB Server Table List
=====================

Index Server Address  State ATM Addr
----- --------------  ----- --------
0     6.6.6.2         UP    39840F00000000000000000000210005A00DEADC8
```

entry — Lists a specific NHRP mib entry in either:

- Server table
- Client table
- Next-Hop Server (NHS) statistics table
- Next-Hop Client (NHC) statistics table
- Resolution cache table

Example: mib entry serv

```
Index [0]? 0
Index           : 0
Protocol        : 1x0800
Protocol Address: 6.6.6.2
ATM Address type: 0x0 (NSAP)
ATM Address     : 39840F000....
SubnetworkId    : 0
Authentication  : 1
Current Clients : 0
Max Clients     : 512
State           : 1
Net             : 1
```

LANE Shortcuts

Use the **lane shortcuts** command to display all or specific entries using LANE shortcuts. You can also display any ATM addresses for which LANE shortcuts are disallowed due to operational problems.

Syntax: lane-shortcuts all
entry
disallowed

all — Displays all LANE shortcuts.

Example: lane all

```
LANE Shortcut Interface #: 1, ATM Network Interface #: 0
=========================================================
   Next Hop Prot @   Dest Mac @         VPI/VCI
   ---------------   -----------------  --------
   5.5.5.1           04-AA-AA-AA-AA-01    0/34

   Current MTU being used: 4490
```

entry — Displays a LANE shortcut entry.

Example: lane entry

```
LANE Shortcut Interface number [0]? 1
Enter IP address of next hop [0.0.0.0]? 5.5.5.1
Next Hop Addr:  5.5.5.1
Dest Mac Addr:  04-AA-AA-AA-AA-01
ATM Address:    39840F000000000000000000310005A00DEAD02
Media type:     Token Ring
VPI/VCI:          0/34
Holding Time:   20 minutes
MTU size:       4490
RI Field:064001020203
```

disallowed — Displays all disallowed LANE shortcut entries.

Any ATM address listed in this display means that the NHRP LANE Shortcut Interface received data from that ATM address. This is not allowed since all NHRP LANE Shortcut Interface VCCs will be used only to transmit data to a LEC at the other end. If the LEC attempts to send data over a VCC set up by an NHRP LANE Shortcut Interface, then the VCC will be brought down and no further LANE shortcuts will be set up to that LEC.

Once the condition which caused the NHRP LANE Shortcut Interface to receive data has been corrected, then the MSS must be restarted in order to allow that ATM address to be again used for NHRP LANE shortcuts.

Example: lan dis

```
LAN Shortcut Interface #: 2, ATM Network Interface #: 0
=======================================================
Atm Address
-------------------------------------------
39840F000000000000000000310005A00DEAD02
```

CONFIG Parameters

Use the **config parameters** command to display the current NHRP configuration parameter setting.

Syntax: config protocol-access-control-usage
attempt-shortcuts
holding-time
data-rate threshold
cache-sizes
extensions
exclude-list
disallowed-router-to-router-shortcuts
shortcuts-to-atmarp-clients

protocol-access-control-usage
Displays the protocol access control usage parameter.

Example: con prot

```
Protocol access controls use source and destination address.
```

attempt-shortcuts
Displays the attempt shortcuts parameter.

Example: con att

```
NHC attempts shortcuts always
```

holding-time
Displays the holding time parameter.

Example: con hold

```
Holding time is 20 minutes
```

data-rate-threshold
Displays the data-rate threhold parameter.

Example: con data

```
Data rate threshold is 10 packets/second
```

cache-sizes
Displays the cache size.

Example: con cache

```
Resolution cache size is 512 entries
Server purge cache size is 512 entries
Server registrations cache size is 512 entries
```

extensions
Displays the extensions

Example: con ext

```
NHRP Forward transit NHS record client extension OFF
NHRP Reverse transit NHS record client extension OFF
Responder Address client extension OFF
LANE shortcuts extension ON
```

exclude-list
Displays the exclude list entries.

Example: con exc

```
List of NHRP IP exclude records:
   # IP Addr          Mask
   1 7.7.7.7          255.255.255.255
```

disallowed-router-to-router-shortcuts
Displays the disallowed router-to-router list entries.

Example: con dis

```
Disallowed router-to-router shortcuts for IP:
  # Addr             Mask
  1 8.8.8.8          255.255.255.255
```

shortcuts-to-atmarp-clients
Displays the shortcuts to atmarp clients parameter.

Example: con s

```
NHS allows shortcuts to ATMARP clients
```

Exit

Use the **exit** command to exit from the NHRP console environment and return to the Talk 5 (+) prompt.

Syntax: exit

Example: exit

NHRP Packet Tracing

NHRP packet traces can be activated from the Event Logging System (ELS) which is an integral part of the router operating system. See “Using and Configuring the Event Logging System” and “Monitoring the Event Logging System” in *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

The NHRP packet tracing mechanism supports the “set trace decode on” option. This option enables the NHRP packet trace output to be interpreted for viewing. The control frames over the LSI can also be traced apart from the NHRP protocol packets. For details on using the trace facility see the description of the **trace** command in “Monitoring the Event Logging System” in *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1*.

The NHRP protocol packets are identified by event 19 and the LSI control packets are identified by event 113.

Sample trace output #1:

```
#1 Dir:INCOMING  Time:0.0.0.0  Trap:5965
Comp:NHRP  Type:UNKNOWN  Port:1  Circuit:0x000000
Size:164

------------------------------------------------------------------
** NHRP Frame **
AddressFamily:ATM_NSAP  ProtocolType:IPv4  HopCount:16
PacketSize:164
Checksum:0xD5D0  ExtensionOffset:0x0038  Version:1
PktType:ResolutionRequest
SrcAddrTL:20  SrcSubAddrTL:0  SrcProtoLen:4
DstProtoLen:4
Flags:requester is a router  Flags:want authoritative only
Flags:want
unique..
Src NBMA:39840F000000000000000000410005A00DEADCA
Src Protocol Addr:06060601  Dest Protocol Addr:05050501
0038: 00 08 00 1C 08 00 5A 00  00 01 00 0A 00 00 00 00
0048: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0058: 00 08 00 34 08 00 5A 00  00 01 00 0C 00 00 00 00
0068: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0078: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0088: 00 00 00 00 00 00 00 00  00 08 00 08 08 00 5A 00
0098: 00 01 00 06 80 03 00 00  80 00 00 00
```

Sample trace output #2:

```
#2 Dir:OUTGOING  Time:0.0.0.0  Trap:5965
Comp:NHRP  Type:UNKNOWN  Port:1  Circuit:0x000000
Size:236
----------------------------------------------------------------
** NHRP Frame **
AddressFamily:ATM_NSAP  ProtocolType:IPv4  HopCount:64
PacketSize:236
Checksum:0x7981  ExtensionOffset:0x005C  Version:1
PktType:ResolutionReply
SrcAddrTL:20  SrcSubAddrTL:0  SrcProtoLen:4
DstProtoLen:4
Flags:requester is a router  Flags:authoritative info
Flags:requested
info ...
Src NBMA:39840F00000000000000000000410005A00DEADCA
Src Protocol Addr:06060601  Dest Protocol Addr:05050501
0038: 00 FF 00 00 23 DC 04 B0  14 00 04 FF 39 84 0F 00
0048: 00 00 00 00 00 00 00 00  02 10 00 5A 00 DE AD C8
0058: 06 06 06 02 00 08 00 1C  08 00 5A 00 00 01 00 0B
0068: 04 AA AA AA AA 01 01 04  05 05 05 01 11 8A 10 00
0078: 5A 00 DE AD 00 08 00 34  08 00 5A 00 00 01 00 0D
0088: 04 B0 14 00 39 84 0F 00  00 00 00 00 00 00 00 00
0098: 03 10 00 5A 00 DE AD 02  00 00 00 00 00 00 00 00
00A8: 00 00 00 00 00 00 00 00  00 00 00 00 00 08 00 08
00B8: 08 00 5A 00 00 01 00 06  80 03 00 24 00 00 00 00
00C8: 00 00 04 B0 14 00 04 00  39 84 0F 00 00 00 00 00
00D8: 00 00 00 00 02 10 00 5A  00 DE AD C8 06 06 06 02
00E8: 80 00 00 00
```

List of Abbreviations

AAL ATM Adaptation Layer

AAL-5 ATM Adaptation Layer 5

AARP AppleTalk Address Resolution Protocol

ABR area border router

ack acknowledgement

AIX Advanced Interactive Executive

AMA arbitrary MAC addressing

AMP active monitor present

ANSI American National Standards Institute

AP2 AppleTalk Phase 2

APPN Advanced Peer-to-Peer Networking

ARE all-routes explorer

ARI/FCI address recognized indicator/frame copied indicator

ARP Address Resolution Protocol

AS autonomous system

ASBR autonomous system boundary router

ASCII American National Standard Code for Information Interchange

ASN.1 abstract syntax notation 1

ASRT adaptive source routing transparent

ASYNC asynchronous

ATCP AppleTalk Control Protocol

ATM Asynchronous Transfer Mode

ATMARP ARP in Classical IP

ATP AppleTalk Transaction Protocol

AUI attachment unit interface

ayt are you there

BAN Boundary Access Node

BBCM Bridging Broadcast MAnager

BCM BroadCast Manager

BECN backward explicit congestion notification

BGP Border Gateway Protocol

BGP Border Growth 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

BUS Broadcast and Unknown Server

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

CIPC Classical IP Client

CIR committed information rate

CLNP Connectionless-Mode Network Protocol

CPU central processing unit

CRC cyclic redundancy check

CRS configuration report server

CTS clear to send

CUD call user data

DAF destination address filtering

DB database

DBsum database summary

DCD data channel received line signal detector

DCE data circuit-terminating equipment

DDLC dual data-link controller

DDN Defense Data Network

DDP Datagram Delivery Protocol

DDT Dynamic Debugging Tool

DHCP Dynamic Host Configuration Protocol

dir directly connected

DL data link

DLC data link control

DLCI data link connection identifier

DLS data link switching

DLSw data link switching

DMA direct memory access

DNA Digital Network Architecture

DNCP DECnet Protocol Control Protocol

DNIC Data Network Identifier Code

DoD Department of Defense

DOS Disk Operating System

DR designated router

DRAM Dynamic Random Access Memory

DSAP destination service access point

DSE data switching equipment

DSE data switching exchange

DSR data set ready

DSU data service unit

DTE data terminal equipment

DTR data terminal ready

Dtype destination type

DVMRP Distance Vector Multicast Routing Protocol

E1 2.048 Mbps transmission rate

EDEL end delimiter

EDI error detected indicator

EGP Exterior Gateway Protocol

EIA Electronics Industries Association

ELAN Emulated Local Area Network

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

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

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

IETF Internet Engineering Task Force

Ifc# interface number

IGP interior gateway protocol

ILMI Interim Local Management Interface

InARP Inverse Address Resolution Protocol

IP Internet Protocol

IPCP IP Control Protocol

IPPN IP Protocol Network

IPX Internetwork Packet Exchange

IPXCP IPX Control Protocol

ISDN integrated services digital network

ISO International Organization for Standardization

Kbps kilobits per second

LAN local area network

LAPB link access protocol-balanced

LAT local area transport

LCP Link Control Protocol

LE LAN Emulation

LEC LAN Emulation Client

LED light-emitting diode

LECS LAN Emulation Configuration Server

LES LAN Emulation Server

LES-BUS LAN Emulation Server - Broadcast and Unknown Server

LF largest frame; line feed

LIS Logical IP subnet

LLC logical link control

LLC2 logical link control 2

LMI local management interface

LRM LAN reporting mechanism

LS link state

LSA link state advertisement

LSB least significant bit

LSreq link state request

LSrxl link state retransmission list

LU logical unit

MAC medium access control

Mb megabit

MB megabyte

Mbps megabits per second

MBps megabytes per second

MC multicast

MCF MAC filtering

MIB Management Information Base

MIB II Management Information Base II

MILNET military network

MOS Micro Operating System

MOSDDT Micro Operating System Dynamic Debugging Tool

MOSPF Open Shortest Path First with multicast extensions

MSB most significant bit

MSDU MAC service data unit

MSS Multiprotocol Switched Services

MTU maximum transmission unit

nak not acknowledged

NBP Name Binding Protocol

NBR neighbor

NCP Network Control Protocol

NCP Network Core Protocol

NetBIOS Network Basic Input/Output System

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 non-volatile configuration

OPCON Operator Console

OSI open systems interconnection

OSICP OSI Control Protocol

OSPF Open Shortest Path First

OUI organization unique identifier

PC personal computer

PDN public data network

PING Packet internet groper

PDU protocol data unit

PID process identification

P-P Point-to-Point

PPP Point-to-Point Protocol

PROM programmable read-only memory

PU physical unit

PVC permanent virtual circuit

QoS Quality of Service

RAM random access memory

RD route descriptor

REM ring error monitor

REV receive

RFC Request for Comments

RI ring indicator; routing information

RIF routing information field

RII routing information indicator

RIP Routing Information Protocol

RISC reduced instruction-set computer

RNR receive not ready

ROM read-only memory

ROpcon Remote Operator Console

RPS ring parameter server

RTMP Routing Table Maintenance Protocol

RTP RouTing update Protocol

RTS request to send

Rtype route type

rxmits retransmissions

rxmt retransmit

SAF source address filtering

SAP service access point

SAP Service Advertising Protocol

sdel start delimiter

SDLC SDLC relay, synchronous data link control

SDU Service Data Unit

seqno sequence number

SGMP Simple Gateway Monitoring Protocol

SL	serial line
SLIP	Serial Line IP
SMP	standby monitor present
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
SNAP	Subnetwork Access Protocol
	SubNetwork Attachment Point
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
SVN	Switched Virtual Networking
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
TLV	Type/Length/Value
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
UNI	User-Network Interface
UTP	unshielded twisted pair
VCC	Virtual Channel connection
VINES	VIrtual NEtworking System
VIR	variable information rate
VL	virtual link
VR	virtual route
WAN	wide area network
WRS	WAN restoral
X.25	packet-switched networks
X.251	X.25 physical layer
X.252	X.25 frame layer
X.253	X.25 packet layer
XID	exchange identification
XNS	Xerox Network Systems
XSUM	checksum
ZIP	AppleTalk Zone Information Protocol
ZIP2	AppleTalk Zone Information Protocol 2
ZIT	Zone Information Table

Glossary

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- Internet Request for Comments: 1208, *Glossary of Networking Terms*.
- Internet Request for Comments: 1392, *Internet Users' Glossary*.
- The *Object-Oriented Interface Design: IBM Common User Access Guidelines*, Carmel, Indiana: Que, 1992.

The following cross-references are used in this glossary:

Contrast with: This refers to a term that has an opposed or substantively different meaning.

Synonym for: This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with: This is a backward reference from a defined term to all other terms that have the same meaning.

See: This refers the reader to multiple-word terms that have the same last word.

See also: This refers the reader to terms that have a related, but not synonymous, meaning.

A

AAL. ATM Adaptation Layer, the layer that adapts user data to/from the ATM network by adding/removing headers and segmenting/reassembling the data into/from cells.

AAL-5. ATM Adaptation Layer 5, one of several standard AALs. AAL-5 was designed for data communications, and is used by LAN Emulation and Classical IP.

abstract syntax. A data specification that includes all distinctions that are needed in data transmissions, but that omits (abstracts) other details such as those that depend on specific computer architectures. See also *abstract syntax notation 1 (ASN.1)* and *basic encoding rules (BER)*.

abstract syntax notation 1 (ASN.1). The Open Systems Interconnection (OSI) method for abstract syntax specified in the following standards:

- ITU-T Recommendation X.208 (1988) | ISO/IEC 8824: 1990
- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1: 1994

See also *basic encoding rules (BER)*.

ACCESS. In the Simple Network Management Protocol (SNMP), the clause in a Management Information Base (MIB) module that defines the minimum level of support that a managed node provides for an object.

acknowledgment. (1) The transmission, by a receiver, of acknowledge characters as an affirmative response to a sender. (T) (2) An indication that an item sent was received.

active monitor. In a token-ring network, a function performed at any one time by one ring station that initiates the transmission of tokens and provides token error recovery facilities. Any active adapter on the ring has the ability to provide the active monitor function if the current active monitor fails.

address. In data communication, the unique code assigned to each device, workstation, or user connected to a network.

address mapping table (AMT). A table, maintained within the AppleTalk router, that provides a current mapping of node addresses to hardware addresses.

address mask. For internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address. Synonymous with *subnet mask* and *subnetwork mask*.

address resolution. (1) A method for mapping network-layer addresses to media-specific addresses. (2) See also *Address Resolution Protocol (ARP)* and *AppleTalk Address Resolution Protocol (AARP)*.

Address Resolution Protocol (ARP). (1) In the Internet suite of protocols, the protocol that dynamically maps an IP address to an address used by a supporting metropolitan or local area network such as Ethernet or token-ring. (2) See also *Reverse Address Resolution Protocol (RARP)*.

addressing. In data communication, the way in which a station selects the station to which it is to send data.

adjacent nodes. Two nodes connected together by at least one path that connects no other node. (T)

Administrative Domain. A collection of hosts and routers, and the interconnecting networks, managed by a single administrative authority.

Advanced Peer-to-Peer Networking (APPN). An extension to SNA featuring (a) greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure; (b) dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection; (c) dynamic definition of network resources; and (d) automated resource registration and directory lookup. APPN extends the LU 6.2 peer orientation for end-user services to network control and supports multiple LU types, including LU 2, LU 3, and LU 6.2.

Advanced Peer-to-Peer Networking (APPN) end node. A node that provides a broad range of end-user services and supports sessions between its local control point (CP) and the CP in an adjacent network node. It uses these sessions to dynamically register its resources with the adjacent CP (its network node server), to send and receive directory search requests, and to obtain management services. An APPN end node can also attach to a subarea network as a peripheral node or to other end nodes.

Advanced Peer-to-Peer Networking (APPN) network. A collection of interconnected network nodes and their client end nodes.

Advanced Peer-to-Peer Networking (APPN) network node. A node that offers a broad range of end-user services and that can provide the following:

- Distributed directory services, including registration of its domain resources to a central directory server
- Topology database exchanges with other APPN network nodes, enabling network nodes throughout the network to select optimal routes for LU-LU sessions based on requested classes of service
- Session services for its local LUs and client end nodes
- Intermediate routing services within an APPN network

Advanced Peer-to-Peer Networking (APPN) node. An APPN network node or an APPN end node.

alert. A message sent to a management services focal point in a network to identify a problem or an impending problem.

all-stations address. In communications, synonym for *broadcast address*.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. (A)

analog. (1) Pertaining to data consisting of continuously variable physical quantities. (A) (2) Contrast with *digital*.

AppleTalk. A network protocol developed by Apple Computer, Inc. This protocol is used to interconnect network devices, which can be a mixture of Apple and non-Apple products.

AppleTalk Address Resolution Protocol (AARP). In AppleTalk networks, a protocol that (a) translates AppleTalk node addresses into hardware addresses and (b) reconciles addressing discrepancies in networks that support more than one set of protocols.

AppleTalk Transaction Protocol (ATP). In AppleTalk networks, a protocol that provides client/server request and response functions for hosts accessing the Zone Information Protocol (ZIP) for zone information.

APPN network. See *Advanced Peer-to-Peer Networking (APPN) network*.

APPN network node. See *Advanced Peer-to-Peer Networking (APPN) network node*.

arbitrary MAC addressing (AMA). In DECnet architecture, an addressing scheme used by DECnet

Phase IV-Prime that supports universally administered addresses and locally administered addresses.

area. In Internet and DECnet routing protocols, a subset of a network or gateway grouped together by definition of the network administrator. Each area is self-contained; knowledge of an area's topology remains hidden from other areas.

asynchronous (ASYNC). Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T)

ATM. Asynchonous Transfer Mode, a connection-oriented, high-speed networking technology based on cell switching.

ATMARP. ARP in Classical IP.

attachment unit interface (AUI). In a local area network, the interface between the medium attachment unit and the data terminal equipment within a data station. (I) (A)

authentication failure. In the Simple Network Management Protocol (SNMP), a trap that may be generated by an authentication entity when a requesting client is not a member of the SNMP community.

autonomous system. In TCP/IP, a group of networks and routers under one administrative authority. These networks and routers cooperate closely to propagate network reachability (and routing) information among themselves using an interior gateway protocol of their choice.

autonomous system number. In TCP/IP, a number assigned to an autonomous system by the same central authority that also assigns IP addresses. The autonomous system number makes it possible for automated routing algorithms to distinguish autonomous systems.

B

BCM. BroadCast Manager, an IBM extension to LAN Emulation designed to limit the effects of broadcast frames.

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).

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)

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)

Border Gateway Protocol (BGP). An Internet Protocol (IP) routing protocol used between domains and autonomous systems. Contrast with *Exterior Gateway Protocol (EGP)*.

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*.

BUS. Broadcast and Unknown Server, a LAN Emulation Service component responsible for the delivery of multicast and unknown unicast frames.

C

cache. (1) A special-purpose buffer storage, smaller and faster than main storage, used to hold a copy of instructions and data obtained from main storage and likely to be needed next by the processor. (T) (2) A buffer storage that contains frequently accessed instructions and data; it is used to reduce access time. (3) An optional part of the directory database in network nodes where frequently used directory information may be stored to speed directory searches. (4) To place, hide, or store in a cache.

call request packet. (1) A call supervision packet that a data terminal equipment (DTE) transmits to ask that a connection for a call be established throughout the network. (2) In X.25 communications, a call supervision packet transmitted by a DTE to ask for a call establishment through the network.

canonical address. In LANs, the IEEE 802.1 format for the transmission of medium access control (MAC) addresses for token-ring and Ethernet adapters. In canonical format, the least significant (rightmost) bit of each address byte is transmitted first. Contrast with *noncanonical address*.

carrier. An electric or electromagnetic wave or pulse train that may be varied by a signal bearing information to be transmitted over a communication system. (T)

carrier detect. Synonym for *received line signal detector (RLSD)*.

carrier sense. In a local area network, an ongoing activity of a data station to detect whether another station is transmitting. (T)

carrier sense multiple access with collision detection (CSMA/CD). A protocol that requires carrier sense and in which a transmitting data station that detects another signal while transmitting, stops sending, sends a jam signal, and then waits for a variable time before trying again. (T) (A)

channel. (1) A path along which signals can be sent, for example, data channel, output channel. (A) (2) A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment.

channel service unit (CSU). A unit that provides the interface to a digital network. The CSU provides line conditioning (or equalization) functions, which keep the signal's performance consistent across the channel bandwidth; signal reshaping, which constitutes the binary pulse stream; and loopback testing, which includes the transmission of test signals between the CSU and the network carrier's office channel unit. See also *data service unit (DSU)*.

checksum. (1) The sum of a group of data associated with the group and used for checking purposes. (T) (2) In error detection, a function of all bits in a block. If the written and calculated sums do not agree, an error is indicated. (3) On a diskette, data written in a sector for error detection purposes; a calculated checksum that does not match the checksum of data written in the sector indicates a bad sector. The data are either numeric or other character strings regarded as numeric for the purpose of calculating the checksum.

CIP. Classical IP.

CIPC. Classical IP Client.

Classical IP. An IETF standard for ATM-attached hosts to communicate using IP over ATM.

Classical IP Client. A Classical IP component that represents users of the Logical IP Subnet.

circuit switching. (1) A process that, on demand, connects two or more data terminal equipment (DTEs) and permits the exclusive use of a data circuit between them until the connection is released. (I) (A) (2) Synonymous with *line switching*.

class A network. In Internet communications, a network in which the high-order (most significant) bit of the IP address is set to 0 and the host ID occupies the three low-order octets.

class B network. In Internet communications, a network in which the two high-order (most significant and next-to-most significant) bits of the IP address are set to 1 and 0, respectively, and the host ID occupies the two low-order octets.

class of service (COS). A set of characteristics (such as route security, transmission priority, and bandwidth) used to construct a route between session partners. The class of service is derived from a mode name specified by the initiator of a session.

client. (1) A functional unit that receives shared services from a server. (T) (2) A user.

client/server. In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

clocking. (1) In binary synchronous communication, the use of clock pulses to control synchronization of data and control characters. (2) A method of controlling the number of data bits sent on a telecommunication line in a given time.

collision. An unwanted condition that results from concurrent transmissions on a channel. (T)

collision detection. In carrier sense multiple access with collision detection (CSMA/CD), a signal indicating that two or more stations are transmitting simultaneously.

Committed information rate. The maximum amount of data in bits that the network agrees to deliver.

community. In the Simple Network Management Protocol (SNMP), an administrative relationship between entities.

community name. In the Simple Network Management Protocol (SNMP), a string of octets identifying a community.

compression. (1) The process of eliminating gaps, empty fields, redundancies, and unnecessary data to shorten the length of records or blocks. (2) Any encoding to reduce the number of bits used to represent a given message or record.

configuration. (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (T) (2) The devices and programs that make up a system, subsystem, or network.

configuration file. A file that specifies the characteristics of a system device or network.

configuration parameter. A variable in a configuration definition, the values of which can characterize the relationship of a product to other products in the same network or can define characteristics of the product itself.

configuration report server (CRS). In the IBM Token-Ring Network Bridge Program, the server that accepts commands from the LAN Network Manager (LNM) to get station information, set station parameters, and remove stations on its ring. This server also collects and forwards configuration reports generated by stations on its ring. The configuration reports include the new active monitor reports and the nearest active upstream neighbor (NAUN) reports.

congestion. See *network congestion*.

control point (CP). (1) A component of an APPN or LEN node that manages the resources of that node. In an APPN node, the CP is capable of engaging in CP-CP sessions with other APPN nodes. In an APPN network node, the CP also provides services to adjacent end nodes in the APPN network. (2) A component of a node that manages resources of that node and optionally provides services to other nodes in the network. Examples are a system services control point (SSCP) in a type 5 subarea node, a network node control point (NNCP) in an APPN network node, and an end node control point (ENCP) in an APPN or LEN end node. An SSCP and an NNCP can provide services to other nodes.

control point management services (CPMS). A component of a control point, consisting of management services function sets, that provides facilities to assist in performing problem management, performance and accounting management, change management, and configuration management. Capabilities provided by the CPMS include sending requests to physical unit management services (PUMS) to test system resources, collecting statistical information (for example, error and performance data) from PUMS about the system resources, and analyzing and presenting test results and statistical information collected about the system resources. Analysis and presentation responsibilities for problem determination and performance monitoring can be distributed among multiple CPMSs.

control point management services unit (CP-MSU). The message unit that contains management services data and flows between management services function sets. This message unit is in general data stream (GDS) format. See also *management services unit (MSU)* and *network management vector transport (NMVT)*.

D

D-bit. Delivery-confirmation bit. In X.25 communications, the bit in a data packet or call-request packet that is set to 1 if end-to-end acknowledgment (delivery confirmation) is required from the recipient.

daemon. A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their task; others operate periodically.

data carrier detect (DCD). Synonym for *received line signal detector (RLSD)*.

data circuit. (1) A pair of associated transmit and receive channels that provide a means of two-way data communication. (I) (2) In SNA, synonym for *link connection*. (3) See also *physical circuit* and *virtual circuit*.

Notes:

1. Between data switching exchanges, the data circuit may include data circuit-terminating equipment (DCE), depending on the type of interface used at the data switching exchange.
2. Between a data station and a data switching exchange or data concentrator, the data circuit includes the data circuit-terminating equipment at the data station end, and may include equipment similar to a DCE at the data switching exchange or data concentrator location.

data circuit-terminating equipment (DCE). In a data station, the equipment that provides the signal conversion and coding between the data terminal equipment (DTE) and the line. (I)

Notes:

1. The DCE may be separate equipment or an integral part of the DTE or of the intermediate equipment.
2. A DCE may perform other functions that are usually performed at the network end of the line.

data link connection identifier (DLCI). The numeric identifier of a frame-relay subport or PVC segment in a frame-relay network. Each subport in a single frame-relay port has a unique DLCI. The following table, excerpted from the American National Standards Institute (ANSI) Standard T1.618 and the International Telegraph and Telephone Consultative Committee (ITU-T/CCITT) Standard Q.922, indicates the functions associated with certain DLCI values:

DLCI Values	Function
0	in-channel signaling
1–15	reserved
16–991	assigned using frame-relay connection procedures
992–1007	layer 2 management of frame-relay bearer service
1008–1022	reserved
1023	in-channel layer management

data link control (DLC). A set of rules used by nodes on a data link (such as an SDLC link or a token ring) to accomplish an orderly exchange of information.

data link control (DLC) layer. In SNA, the layer that consists of the link stations that schedule data transfer over a link between two nodes and perform error control for the link. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

Note: The DLC layer is usually independent of the physical transport mechanism and ensures the integrity of data that reaches the higher layers.

data link layer. In the Open Systems Interconnection reference model, the layer that provides services to transfer data between entities in the network layer over a communication link. The data link layer detects and possibly corrects errors that may occur in the physical layer. (T)

data link level. (1) In the hierarchical structure of a data station, the conceptual level of control or processing logic between high level logic and the data link that maintains control of the data link. The data link level performs such functions as inserting transmit bits and deleting receive bits; interpreting address and control fields; generating, transmitting, and interpreting commands and responses; and computing and interpreting frame check sequences. See also *packet level* and *physical level*. (2) In X.25 communications, synonym for *frame level*.

data link switching (DLSw). A method of transporting network protocols that use IEEE 802.2 logical link control (LLC) type 2. SNA and NetBIOS are examples of protocols that use LLC type 2. See also *encapsulation* and *spoofing*.

data packet. In X.25 communications, a packet used for the transmission of user data on a virtual circuit at the DTE/DCE interface.

data service unit (DSU). A device that provides a digital data service interface directly to the data terminal equipment. The DSU provides loop equalization, remote and local testing capabilities, and a standard EIA/CCITT interface.

data set ready (DSR). Synonym for *DCE ready*.

data switching exchange (DSE). The equipment installed at a single location to provide switching functions, such as circuit switching, message switching, and packet switching. (I)

data terminal equipment (DTE). That part of a data station that serves as a data source, data sink, or both. (I) (A)

data terminal ready (DTR). A signal to the modem used with the EIA 232 protocol.

data transfer rate. The average number of bits, characters, or blocks per unit time passing between corresponding equipment in a data transmission system. (I)

Notes:

1. The rate is expressed in bits, characters, or blocks per second, minute, or hour.
2. Corresponding equipment should be indicated; for example, modems, intermediate equipment, or source and sink.

datagram. (1) In packet switching, a self-contained packet, independent of other packets, that carries information sufficient for routing from the originating data terminal equipment (DTE) to the destination DTE without relying on earlier exchanges between the DTEs and the network. (I) (2) In TCP/IP, the basic unit of information passed across the Internet environment. A datagram contains a source and destination address along with the data. An Internet Protocol (IP) datagram consists of an IP header followed by the transport layer data. (3) See also *packet* and *segment*.

Datagram Delivery Protocol (DDP). In AppleTalk networks, a protocol that provides network connectivity by means of connectionless socket-to-socket delivery service on the internet layer.

DCE ready. In the EIA 232 standard, a signal that indicates to the data terminal equipment (DTE) that the local data circuit-terminating equipment (DCE) is connected to the communication channel and is ready to send data. Synonymous with *data set ready (DSR)*.

DECnet. A network architecture that defines the operation of a family of software modules, databases, and hardware components typically used to tie Digital Equipment Corporation systems together for resource sharing, distributed computation, or remote system configuration. DECnet network implementations follow the Digital Network Architecture (DNA) model.

default. Pertaining to an attribute, condition, value, or option that is assumed when none is explicitly specified. (I)

designated router. A router that informs end nodes of the existence and identity of other routers. The selection of the designated router is based upon the router with the highest priority. When several routers share the highest priority, the router with the highest station address is selected.

destination node. The node to which a request or data is sent.

destination port. The 8-port asynchronous adapter that serves as a connection point with a serial service.

destination service access point (DSAP). In SNA and TCP/IP, a logical address that allows a system to route data from a remote device to the appropriate communications support. Contrast with *source service access point (SSAP)*.

device. A mechanical, electrical, or electronic contrivance with a specific purpose.

digital. (1) Pertaining to data that consist of digits. (T) (2) Pertaining to data in the form of digits. (A) (3) Contrast with *analog*.

Digital Network Architecture (DNA). The model for all DECnet hardware and software implementations.

direct memory access (DMA). The system facility that allows a device on the Micro Channel bus to get direct access to the system or bus memory without the intervention of the system processor.

directory. A table of identifiers and references to the corresponding items of data. (I) (A)

directory service (DS). An application service element that translates the symbolic names used by application processes into the complete network addresses used in an OSI environment. (T)

directory services (DS). A control point component of an APPN node that maintains knowledge of the location of network resources.

disable. To make nonfunctional.

disabled. (1) Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. (2) Pertaining to the state in which a transmission control unit or audio response unit cannot accept incoming calls on a line.

domain. (1) That part of a computer network in which the data processing resources are under common control. (T) (2) In Open Systems Interconnection (OSI), a part of a distributed system or a set of managed objects to which a common policy applies. (3) See *Administrative Domain* and *domain name*.

domain name. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of a host system is `ralvm7.vnet.ibm.com`, each of the following is a domain name:

- `ralvm7.vnet.ibm.com`
- `vnet.ibm.com`
- `ibm.com`

domain name server. In the Internet suite of protocols, a server program that supplies name-to-address translation by mapping domain names to IP addresses. Synonymous with *name server*.

Domain Name System (DNS). In the Internet suite of protocols, the distributed database system used to map domain names to IP addresses.

dotted decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

dump. (1) Data that has been dumped. (T) (2) To copy the contents of all or part of virtual storage for the purpose of collecting error information.

dynamic reconfiguration (DR). The process of changing the network configuration (peripheral PUs and LUs) without regenerating complete configuration tables or deactivating the affected major node.

Dynamic Routing. Routing using learned routes rather than routes statically configured at initialization.

E

echo. In data communication, a reflected signal on a communications channel. For example, on a communications terminal, each signal is displayed twice, once when entered at the local terminal and again when returned over the communications link. This allows the signals to be checked for accuracy.

EIA 232. In data communication, a specification of the Electronic Industries Association (EIA) that defines the interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE), using serial binary data interchange.

ELAN. Emulated Local Area Network, a LAN segment implemented with ATM technology.

Electronic Industries Association (EIA). An organization of electronics manufacturers that advances the technological growth of the industry, represents the views of its members, and develops industry standards.

encapsulation. (1) In communications, a technique used by layered protocols by which a layer adds control information to the protocol data unit (PDU) from the layer it supports. In this respect, the layer encapsulates the data from the supported layer. In the Internet suite of protocols, for example, a packet would contain control information from the physical layer, followed by control information from the network layer, followed by the application protocol data. (2) See also *data link switching*.

encode. To convert data by the use of a code in such a manner that reconversion to the original form is possible. (T)

end node (EN). (1) See *Advanced Peer-to-Peer Networking (APPN) end node* and *low-entry networking (LEN) end node*. (2) In communications, a node that is frequently attached to a single data link and cannot perform intermediate routing functions.

entry point (EP). In SNA, a type 2.0, type 2.1, type 4, or type 5 node that provides distributed network management support. It sends network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal-point initiated commands to manage and control its resources.

ESI. End System Identifier, a 6-byte component of an ATM address.

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. Contrast with *Border Gateway Protocol (BGP)*.

F

File Transfer Protocol (FTP). In the Internet suite of protocols, an application layer protocol that uses TCP and Telnet services to transfer bulk-data files between machines or hosts.

flow control. (1) In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. (2) See also *pacing*.

fragment. See *fragmentation*.

fragmentation. (1) The process of dividing a datagram into smaller parts, or fragments, to match the capabilities of the physical medium over which it is to be transmitted. (2) See also *segmenting*.

frame. (1) In Open Systems Interconnection architecture, a data structure pertaining to a particular area of knowledge and consisting of slots that can accept the values of specific attributes and from which inferences can be drawn by appropriate procedural attachments. (T) (2) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (3) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

frame level. Synonymous with *data link level*. See *link level*.

frame relay. (1) An interface standard describing the boundary between a user's equipment and a fast-packet network. In frame-relay systems, flawed frames are discarded; recovery comes end-to-end rather than hop-by-hop. (2) A technique derived from the integrated services digital network (ISDN) D channel standard. It assumes that connections are reliable and dispenses with the overhead of error detection and control within the network.

G

gateway. (1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T) (2) In the IBM Token-Ring Network, a device and its associated software that connect a local area network to another local area network or a host that uses different logical link protocols. (3) In TCP/IP, synonym for *router*.

general data stream (GDS). The data stream used for conversations in LU 6.2 sessions.

general data stream (GDS) variable. A type of RU substructure that is preceded by an identifier and a length field and includes either application data, user control data, or SNA-defined control data.

H

header. (1) System-defined control information that precedes user data. (2) The portion of a message that contains control information for the message such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

heap memory. The amount of RAM used to dynamically allocate data structures.

Hello. A protocol used by a group of cooperating, trusting routers to allow them to discover minimal delay routes.

hello message. (1) A message sent periodically to establish and test reachability between routers or between routers and hosts. (2) In the Internet suite of protocols, a message defined by the Hello protocol as an Interior Gateway Protocol (IGP).

heuristic. Pertaining to exploratory methods of problem solving in which solutions are discovered by evaluation of the progress made toward the final result.

high-level data link control (HDLC). In data communication, the use of a specified series of bits to control data links in accordance with the International Standards for HDLC: ISO 3309 Frame Structure and ISO 4335 Elements of Procedures.

hop. (1) In APPN, a portion of a route that has no intermediate nodes. It consists of only a single transmission group connecting adjacent nodes. (2) To the routing layer, the logical distance between two nodes in a network.

hop count. (1) A metric or measure of distance between two points. (2) In Internet communications, the number of routers that a datagram passes through on its way to its destination. (3) In SNA, a measure of the number of links to be traversed in a path to a destination.

host. In the Internet suite of protocols, an end system. The end system can be any workstation; it does not have to be a mainframe.

hysteresis. The amount the temperature must change past the set alert threshold before the alert condition is cleared.

I

I frame. Information frame.

IETF. Internet Engineering Task Force, an organization that produces Internet specifications.

ILMI. Interim Local Management Interface, SNMP-based procedures for managing the User-Network Interface (UNI).

information (I) frame. A frame in I format used for numbered information transfer.

input/output channel. In a data processing system, a functional unit that handles transfer of data between internal and peripheral equipment. (I) (A)

integrated services digital network (ISDN). A digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

Note: ISDNs are used in public and private network architectures.

interface. (1) A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions. (T) (2) Hardware, software, or both, that links systems, programs, or devices.

interior gateway. In Internet communications, a gateway that communicates only with its own autonomous system. Contrast with *exterior gateway*.

Interior Gateway Protocol (IGP). In the Internet suite of protocols, a protocol used to propagate network reachability and routing information within an autonomous system. Examples of IGPs are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

intermediate node. A node that is at the end of more than one branch. (T)

intermediate session routing (ISR). A type of routing function within an APPN network node that provides session-level flow control and outage reporting for all sessions that pass through the node but whose end points are elsewhere.

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

International Telecommunication Union (ITU). The specialized telecommunication agency of the United Nations, established to provide standardized communication procedures and practices, including frequency allocation and radio regulations worldwide.

internet. A collection of networks interconnected by a set of routers that allow them to function as a single, large network. See also *Internet*.

Internet. The internet administered by the Internet Architecture Board (IAB), consisting of large national backbone networks and many regional and campus networks all over the world. The Internet uses the Internet suite of protocols.

Internet address. See *IP address*.

Internet Architecture Board (IAB). The technical body that oversees the development of the Internet suite of protocols that are known as TCP/IP.

Internet Control Message Protocol (ICMP). The protocol used to handle errors and control messages in the Internet Protocol (IP) layer. Reports of problems and incorrect datagram destinations are returned to the original datagram source. ICMP is part of the Internet Protocol.

Internet Control Protocol (ICP). The VIrtual NEtworking System (VINES) protocol that provides exception notifications, metric notifications, and PING support. See also *RouTing update Protocol (RTP)*.

Internet Engineering Task Force (IETF). The task force of the Internet Architecture Board (IAB) that is responsible for solving the short-term engineering needs of the Internet.

Internet Protocol (IP). A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery

and flow control and does not guarantee the reliability of the physical network.

Internetwork Packet Exchange (IPX). (1) The network protocol used to connect Novell's servers, or any workstation or router that implements IPX, with other workstations. Although similar to the Internet Protocol (IP), IPX uses different packet formats and terminology. (2) See also *Xerox Network Systems (XNS)*.

interoperability. The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units. (T)

intra-area routing. In Internet communications, the routing of data within an area.

Inverse Address Resolution Protocol (InARP). In the Internet suite of protocols, the protocol used for locating a protocol address through the known hardware address. In a frame-relay context, the data link connection identifier (DLCI) is synonymous with the known hardware address.

IP address. The 32-bit address defined by the Internet Protocol, standard 5, Request for Comments (RFC) 791. It is usually represented in dotted decimal notation.

IP datagram. In the Internet suite of protocols, the fundamental unit of information transmitted through an internet. It contains source and destination addresses, user data, and control information such as the length of the datagram, the header checksum, and flags indicating whether the datagram can be or has been fragmented.

IP router. A device in an IP internet that is responsible for making decisions about the paths over which network traffic will flow. Routing protocols are used to gain information about the network and to determine the best route over which the datagram should be forwarded toward the final destination. The datagrams are routed based on IP destination addresses.

IPXWAN. A Novell protocol that is used to exchange router-to-router information before exchanging standard Internetwork Packet Exchange (IPX) routing information and traffic over wide area networks (WANs).

L

LAN bridge server (LBS). In the IBM Token-Ring Network Bridge Program, the server that keeps statistical information about frames forwarded between two or more rings (through a bridge). The LBS sends these statistics to the appropriate LAN managers through the LAN reporting mechanism (LRM).

LAN Emulation (LE). An ATM Forum standard that supports legacy LAN applications over ATM networks.

LAN Emulation Client (LEC). A LAN Emulation component that represents users of the Emulated LAN.

LAN Emulation Configuration Server (LECS). A LAN Emulation Service component that centralizes and disseminates configuration data.

LAN Emulation Server (LES). A LAN Emulation Service component that resolves LAN Destinations to ATM Addresses.

LAN Network Manager (LNM). An IBM licensed program that enables a user to manage and monitor LAN resources from a central workstation.

LAN segment. (1) Any portion of a LAN (for example, a bus or ring) that can operate independently, but that is connected to other parts of the network by means of bridges. (2) A ring or bus network without bridges.

layer. (1) In network architecture, a group of services that is complete from a conceptual point of view, that is one out of a set of hierarchically arranged groups, and that extends across all systems that conform to the network architecture. (T) (2) In the Open Systems Interconnection reference model, one of seven conceptually complete, hierarchically arranged groups of services, functions, and protocols, that extend across all open systems. (T) (3) In SNA, a grouping of related functions that are logically separate from the functions in other groups. Implementation of the functions in one layer can be changed without affecting functions in other layers.

LE. LAN Emulation.

LEC. LAN Emulation Client.

LECS. LAN Emulation Configuration Server.

LES. LAN Emulation Server.

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)*.

LIS. Logical IP Subnet, an IP subnet implemented with ATM technology Virtual Networking (SVN) framework.

local. (1) Pertaining to a device accessed directly without use of a telecommunication line. (2) Contrast with *remote*. (3) Synonym for *channel-attached*.

local area network (LAN). (1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T) (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network. (3) See also *Ethernet* and *token ring*. (4) Contrast with *metropolitan area network (MAN)* and *wide area network (WAN)*.

local bridging. A function of a bridge program that allows a single bridge to connect multiple LAN segments without using a telecommunication link. Contrast with *remote bridging*.

local management interface (LMI). See *local management interface (LMI) protocol*.

local management interface (LMI) protocol. In NCP, a set of frame-relay network management procedures and messages used by adjacent frame-relay nodes to exchange line status information over DLCI X'00'. NCP supports both the American National Standards Institute (ANSI) and International Telegraph and Telephone Consultative Committee (ITU-T/CCITT) versions of LMI protocol. These standards refer to LMI protocol as *link integrity verification tests (LIVT)*.

locally administered address. In a local area network, an adapter address that the user can assign to override the universally administered address. Contrast with *universally administered address*.

logical channel. In packet mode operation, a sending channel and a receiving channel that together are used to send and receive data over a data link at the same time. Several logical channels can be established on the same data link by interleaving the transmission of packets.

logical link. A pair of link stations, one in each of two adjacent nodes, and their underlying link connection, providing a single link-layer connection between the two nodes. Multiple logical links can be distinguished while they share the use of the same physical media connecting two nodes. Examples are 802.2 logical links used on local area network (LAN) facilities and LAP E logical links on the same point-to-point physical link between two nodes. The term logical link also includes the multiple X.25 logical channels that share the use of the access link from a DTE to an X.25 network.

logical link control (LLC). The data link control (DLC) LAN sublayer that provides two types of DLC operation for the orderly exchange of information. The first type is connectionless service, which allows information to be sent and received without establishing a link. The LLC sublayer does not perform error recovery or flow control for connectionless service. The second type is connection-oriented service, which requires establishing a link prior to the exchange of information. Connection-oriented service provides sequenced information transfer, flow control, and error recovery.

logical link control (LLC) protocol. In a local area network, the protocol that governs the exchange of transmission frames between data stations independently of how the transmission medium is shared. (T) The LLC protocol was developed by the

IEEE 802 committee and is common to all LAN standards.

logical link control (LLC) protocol data unit. A unit of information exchanged between link stations in different nodes. The LLC protocol data unit contains a destination service access point (DSAP), a source service access point (SSAP), a control field, and user data.

logical unit (LU). A type of network accessible unit that enables users to gain access to network resources and communicate with each other.

loopback test. A test in which signals from a tester are looped at a modem or other network element back to the tester for measurements that determine or verify the quality of the communications path.

low-entry networking (LEN). A capability of nodes to attach directly to one another using basic peer-to-peer protocols to support multiple and parallel sessions between logical units.

low-entry networking (LEN) end node. A LEN node receiving network services from an adjacent APPN network node.

low-entry networking (LEN) node. A node that provides a range of end-user services, attaches directly to other nodes using peer protocols, and derives network services implicitly from an adjacent APPN network node, that is, without the direct use of CP-CP sessions.

M

Management Information Base (MIB). (1) A collection of objects that can be accessed by means of a network management protocol. (2) A definition for management information that specifies the information available from a host or gateway and the operations allowed. (3) In OSI, the conceptual repository of management information within an open system.

management station. In Internet communications, the system responsible for managing all, or a portion of, a network. The management station communicates with network management agents that reside in the managed node by means of a network management protocol, such as the Simple Network Management Protocol (SNMP).

mapping. The process of converting data that is transmitted in one format by the sender into the data format that can be accepted by the receiver.

mask. (1) A pattern of characters used to control retention or elimination of portions of another pattern of characters. (I) (A) (2) To use a pattern of characters to control retention or elimination of portions of another pattern of characters. (I) (A)

maximum transmission unit (MTU). In LANs, the largest possible unit of data that can be sent on a given physical medium in a single frame. For example, the MTU for Ethernet is 1500 bytes.

medium access control (MAC). In LANs, the sublayer of the data link control layer that supports medium-dependent functions and uses the services of the physical layer to provide services to the logical link control (LLC) sublayer. The MAC sublayer includes the method of determining when a device has access to the transmission medium.

medium access control (MAC) protocol. In a local area network, the protocol that governs access to the transmission medium, taking into account the topological aspects of the network, in order to enable the exchange of data between data stations. (T)

medium access control (MAC) sublayer. In a local area network, the part of the data link layer that applies a medium access method. The MAC sublayer supports topology-dependent functions and uses the services of the physical layer to provide services to the logical link control sublayer. (T)

metric. In Internet communications, a value, associated with a route, which is used to discriminate between multiple exit or entry points to the same autonomous system. The route with the lowest metric is preferred.

metropolitan area network (MAN). A network formed by the interconnection of two or more networks which may operate at higher speed than those networks, may cross administrative boundaries, and may use multiple access methods. (T) Contrast with *local area network (LAN)* and *wide area network (WAN)*.

MIB object. Synonym for *MIB variable*.

MIB variable. In the Simple Network Management Protocol (SNMP), a specific instance of data defined in a MIB module. Synonymous with *MIB object*.

MIB view. In the Simple Network Management Protocol (SNMP), the collection of managed objects, known to the agent, that is visible to a particular community.

MILNET. The military network that was originally part of ARPANET. It was partitioned from ARPANET in 1984. MILNET provides a reliable network service for military installations.

modem (modulator/demodulator). (1) A functional unit that modulates and demodulates signals. One of the functions of a modem is to enable digital data to be

transmitted over analog transmission facilities. (T) (A) (2) A device that converts digital data from a computer to an analog signal that can be transmitted on a telecommunication line, and converts the analog signal received to data for the computer.

modulo. (1) Pertaining to a modulus; for example, 9 is equivalent to 4 modulo 5. (2) See also *modulus*.

modulus. A number, such as a positive integer, in a relationship that divides the difference between two related numbers without leaving a remainder; for example, 9 and 4 have a modulus of 5 (9 - 4 = 5; 4 - 9 = -5; and 5 divides both 5 and -5 without leaving a remainder).

monitor. (1) A device that observes and records selected activities within a data processing system for analysis. Possible uses are to indicate significant departure from the norm, or to determine levels of utilization of particular functional units. (T) (2) Software or hardware that observes, supervises, controls, or verifies operations of a system. (A) (3) The function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations.

MSS. Multiprotocol Switched Services, a component of IBM's Switched Virtual Networking (SVN) framework.

multicast. (1) Transmission of the same data to a selected group of destinations. (T) (2) A special form of broadcast in which copies of a packet are delivered to only a subset of all possible destinations.

multiple-domain support (MDS). A technique for transporting management services data between management services function sets over LU-LU and CP-CP sessions. See also *multiple-domain support message unit (MDS-MU)*.

multiple-domain support message unit (MDS-MU). The message unit that contains management services data and flows between management services function sets over the LU-LU and CP-CP sessions used by multiple-domain support. This message unit, as well as the actual management services data that it contains, is in general data stream (GDS) format. See also *control point management services unit (CP-MSU)*, *management services unit (MSU)*, and *network management vector transport (NMVT)*.

N

Name Binding Protocol (NBP). In AppleTalk networks, a protocol that provides name translation function from the AppleTalk entity (resource) name (character string) into an AppleTalk IP address (16-bit number) on the transport layer.

name resolution. In Internet communications, the process of mapping a machine name to the corresponding Internet Protocol (IP) address. See also *Domain Name System (DNS)*.

name server. In the Internet suite of protocols, synonym for *domain name server*.

nearest active upstream neighbor (NAUN). In the IBM Token-Ring Network, the station sending data directly to a given station on the ring.

neighbor. A router on a common subnetwork that has been designated by a network administrator to receive routing information.

NetBIOS. Network Basic Input/Output System. A standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not need to handle the details of LAN data link control (DLC) protocols.

network. (1) A configuration of data processing devices and software connected for information interchange. (2) A group of nodes and the links interconnecting them.

network accessible unit (NAU). A logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with *network addressable unit*.

network address. According to ISO 7498-3, a name, unambiguous within the OSI environment, that identifies a set of network service access points.

network addressable unit (NAU). Synonym for *network accessible unit*.

network architecture. The logical structure and operating principles of a computer network. (T)

Note: The operating principles of a network include those of services, functions, and protocols.

network congestion. An undesirable overload condition caused by traffic in excess of what a network can handle.

network identifier. (1) In TCP/IP, that part of the IP address that defines a network. The length of the network ID depends on the type of network class (A, B, or C). (2) A 1- to 8-byte customer-selected name or an 8-byte IBM-registered name that uniquely identifies a specific subnetwork.

Network Information Center (NIC). In Internet communications, local, regional, and national groups throughout the world who provide assistance, documentation, training, and other services to users.

network layer. In Open Systems Interconnection (OSI) architecture, the layer that is responsible for routing, switching, and link-layer access across the OSI environment.

network management. The process of planning, organizing, and controlling a communication-oriented data processing or information system.

network management station. In the Simple Network Management Protocol (SNMP), a station that executes management application programs that monitor and control network elements.

network management vector transport (NMVT). A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

network manager. A program or group of programs that is used to monitor, manage, and diagnose the problems of a network.

network node (NN). See *Advanced Peer-to-Peer Networking (APPN) network node*.

Next Hop Resolution Protocol (NHRP). A routing protocol, specified in Internet Draft Version 10 which has been submitted for RFC status. The Next Hop Resolution Protocol defines a method for a source station to determine the Non-Broadcast Multi-Access (NBMA) address of the “NBMA next hop” towards a destination. The NBMA next hop may be the destination itself or the router in the NBMA network that is “nearest” to the destination. The source station can then establish an NBMA virtual circuit directly with the destination or the router and reduce the number of routing hops through the NBMA network.

network user address (NUA). In X.25 communications, the X.121 address containing up to 15 binary code digits.

NHRP. Next Hop Resolution Protocol

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (I) (2) Any device, attached to a network, that transmits and receives data.

noncanonical address. In LANs, a format for the transmission of medium access control (MAC) addresses for token-ring adapters. In noncanonical format, the most significant (leftmost) bit of each address byte is transmitted first. Contrast with *canonical address*.

nonseed router. In AppleTalk networks, a router that acquires network number range and zone list information from a seed router attached to the same network.

O

Open Shortest Path First (OSPF). In the Internet suite of protocols, a function that provides intradomain information transfer. An alternative to the Routing Information Protocol (RIP), OSPF allows the lowest-cost routing and handles routing in large regional or corporate networks.

Open Systems Interconnection (OSI). (1) The interconnection of open systems in accordance with standards of the International Organization for Standardization (ISO) for the exchange of information. (T) (A) (2) The use of standardized procedures to enable the interconnection of data processing systems.

Note: OSI architecture establishes a framework for coordinating the development of current and future standards for the interconnection of computer systems. Network functions are divided into seven layers. Each layer represents a group of related data processing and communication functions that can be carried out in a standard way to support different applications.

Open Systems Interconnection (OSI) architecture. Network architecture that adheres to that particular set of ISO standards that relates to Open Systems Interconnection. (T)

Open Systems Interconnection (OSI) reference model. A model that describes the general principles of the Open Systems Interconnection, as well as the purpose and the hierarchical arrangement of its seven layers. (T)

origin. An external logical unit (LU) or application program from which a message or other data originates. See also *destination*.

orphan circuit. A non-configured circuit whose availability is learned dynamically.

P

pacing. (1) A technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. (2) See also *flow control*, *receive pacing*, *send pacing*, *session-level pacing*, and *virtual route (VR) pacing*.

packet. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)

packet internet groper (PING). (1) In Internet communications, a program used in TCP/IP networks to test the ability to reach destinations by sending the destinations an Internet Control Message Protocol (ICMP) echo request and waiting for a reply. (2) In communications, a test of reachability.

packet mode operation. Synonym for *packet switching*.

packet switching. (1) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during transmission of a packet. On completion of the transmission, the channel is made available for transfer of other packets. (I) (2) Synonymous with *packet mode operation*. See also *circuit switching*.

parallel bridges. A pair of bridges connected to the same LAN segment, creating redundant paths to the segment.

parallel transmission groups. Multiple transmission groups between adjacent nodes, with each group having a distinct transmission group number.

path. (1) In a network, any route between any two nodes. A path may include more than one branch. (T) (2) The series of transport network components (path control and data link control) that are traversed by the information exchanged between two network accessible units. See also *explicit route (ER)*, *route extension*, and *virtual route (VR)*.

path control (PC). The function that routes message units between network accessible units in the network and provides the paths between them. It converts the basic information units (BIUs) from transmission control (possibly segmenting them) into path information units (PIUs) and exchanges basic transmission units containing one or more PIUs with data link control. Path control differs by node type: some nodes (APPN nodes, for example) use locally generated session identifiers for routing, and others (subarea nodes) use network addresses for routing.

path cost. In link-state routing protocols, the sum of the link costs along the path between two nodes or networks.

path information unit (PIU). A message unit consisting of a transmission header (TH) alone, or a TH followed by a basic information unit (BIU) or a BIU segment.

pattern-matching character. A special character such as an asterisk (*) or a question mark (?) that can be used to represent one or more characters. Any character or set of characters can replace a pattern-matching character. Synonymous with *global character* and *wildcard character*.

permanent virtual circuit (PVC). In X.25 and frame-relay communications, a virtual circuit that has a logical channel permanently assigned to it at each data terminal equipment (DTE). Call-establishment protocols are not required. Contrast with *switched virtual circuit (SVC)*.

physical circuit. A circuit established without multiplexing. See also *data circuit*. Contrast with *virtual circuit*.

physical layer. In the Open Systems Interconnection reference model, the layer that provides the mechanical, electrical, functional, and procedural means to establish, maintain, and release physical connections over the transmission medium. (T)

physical unit (PU). (1) The component that manages and monitors the resources (such as attached links and adjacent link stations) associated with a node, as requested by an SSCP via an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. This term applies to type 2.0, type 4, and type 5 nodes only. (2) See also *peripheral PU* and *subarea PU*.

ping command. The command that sends an Internet Control Message Protocol (ICMP) echo-request packet to a gateway, router, or host with the expectation of receiving a reply.

Point-to-Point Protocol (PPP). A protocol that provides a method for encapsulating and transmitting packets over serial point-to-point links.

polling. (1) On a multipoint connection or a point-to-point connection, the process whereby data stations are invited, one at a time, to transmit. (I) (2) Interrogation of devices for such purposes as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (A)

port. (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices

such as display stations and printers are attached. (3) The representation of a physical connection to the link hardware. A port is sometimes referred to as an adapter; however, there can be more than one port on an adapter. There may be one or more ports controlled by a single DLC process. (4) In the Internet suite of protocols, a 16-bit number used to communicate between TCP or the User Datagram Protocol (UDP) and a higher-level protocol or application. Some protocols, such as File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP), use the same well-known port number in all TCP/IP implementations. (5) An abstraction used by transport protocols to distinguish among multiple destinations within a host machine. (6) Synonymous with *socket*.

port number. In Internet communications, the identification of an application entity to the transport service.

problem determination. The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environmental failure such as a power loss, or user error.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM in a current unaltered release of the program.

protocol. (1) A set of semantic and syntactic rules that determine the behavior of functional units in achieving communication. (I) (2) In Open Systems Interconnection architecture, a set of semantic and syntactic rules that determine the behavior of entities in the same layer in performing communication functions. (T) (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. Synonymous with *line control discipline* and *line discipline*. See *bracket protocol* and *link protocol*.

protocol data unit (PDU). A unit of data specified in a protocol of a given layer and consisting of protocol control information of this layer, and possibly user data of this layer. (T)

Q

Quality of Service (QoS). The user-oriented performance of an end-to-end service which is accessed using performance parameters. In ATM networks, the following performance parameters determine the QoS of an end-to-end ATM connection: cell loss ratio, cell transfer delay, and cell delay variation.

R

Rapid Transport Protocol (RTP) connection. In high-performance routing (HPR), the connection established between the endpoints of the route to transport session traffic.

reachability. The ability of a node or a resource to communicate with another node or resource.

read-only memory (ROM). Memory in which stored data cannot be modified by the user except under special conditions.

reassembly. In communications, the process of putting segmented packets back together after they have been received.

receive not ready (RNR). In communications, a data link command or response that indicates a temporary condition of being unable to accept incoming frames.

receive not ready (RNR) packet. See *RNR packet*.

received line signal detector (RLSD). In the EIA 232 standard, a signal that indicates to the data terminal equipment (DTE) that it is receiving a signal from the remote data circuit-terminating equipment (DCE). Synonymous with *carrier detect* and *data carrier detect (DCD)*.

Recognized Private Operating Agency (RPOA). Any individual, company, or corporation, other than a government department or service, that operates a telecommunication service and is subject to the obligations undertaken in the Convention of the International Telecommunication Union and in the Regulations; for example, a communication common carrier.

reduced instruction-set computer (RISC). A computer that uses a small, simplified set of frequently used instructions for rapid execution.

remote. (1) Pertaining to a system, program, or device that is accessed through a telecommunication line. (2) Synonym for *link-attached*. (3) Contrast with *local*.

remote bridging. The function of a bridge that allows two bridges to connect multiple LANs using a telecommunication link. Contrast with *local bridging*.

Remote Execution Protocol (REXEC). A protocol that allows the execution of a command or program on any host in the network. The local host receives the results of the command execution.

Request for Comments (RFC). In Internet communications, the document series that describes a part of the Internet suite of protocols and related

experiments. All Internet standards are documented as RFCs.

reset. On a virtual circuit, reinitialization of data flow control. At reset, all data in transit are eliminated.

reset request packet. In X.25 communications, a packet transmitted by the data terminal equipment (DTE) to the data circuit-terminating equipment (DCE) to request that a virtual call or a permanent virtual circuit be reset. The reason for the request can also be specified in the packet.

ring. See *ring network*.

ring network. (1) A network in which every node has exactly two branches connected to it and in which there are exactly two paths between any two nodes. (T) (2) A network configuration in which devices are connected by unidirectional transmission links to form a closed path.

ring segment. A section of a ring that can be isolated (by unplugging connectors) from the rest of the ring. See *LAN segment*.

rlogin (remote login). A service, offered by Berkeley UNIX-based systems, that allows authorized users of one machine to connect to other UNIX systems across an internet and interact as if their terminals were connected directly. The rlogin software passes information about the user's environment (for example, terminal type) to the remote machine.

RNR packet. A packet used by a data terminal equipment (DTE) or by a data circuit-terminating equipment (DCE) to indicate a temporary inability to accept additional packets for a virtual call or permanent virtual circuit.

root bridge. The bridge that is the root of a spanning tree formed between other active bridges in the bridging network. The root bridge originates and transmits bridge protocol data units (BPDUs) to other active bridges to maintain the spanning tree topology. It is the bridge with the highest priority in the network.

route. (1) An ordered sequence of nodes and transmission groups (TGs) that represent a path from an origin node to a destination node traversed by the traffic exchanged between them. (2) The path that network traffic uses to get from source to destination.

route bridge. A function of an IBM bridge program that allows two bridge computers to use a telecommunication link to connect two LANs. Each bridge computer is connected directly to one of the LANs, and the telecommunication link connects the two bridge computers.

route extension (REX). In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *explicit route (ER)*, *path*, and *virtual route (VR)*.

Route Selection control vector (RSCV). A control vector that describes a route within an APPN network. The RSCV consists of an ordered sequence of control vectors that identify the TGs and nodes that make up the path from an origin node to a destination node.

router. (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached. (4) In TCP/IP, synonymous with *gateway*. (5) Contrast with *bridge*.

routing. (1) The assignment of the path by which a message is to reach its destination. (2) In SNA, the forwarding of a message unit along a particular path through a network, as determined by parameters carried in the message unit, such as the destination network address in a transmission header.

routing domain. In Internet communications, a group of intermediate systems that use a routing protocol so that the representation of the overall network is the same within each intermediate system. Routing domains are connected to each other by exterior links.

Routing Information Protocol (RIP). In the Internet suite of protocols, an interior gateway protocol used to exchange intradomain routing information and to determine optimum routes between internet hosts. RIP determines optimum routes on the basis of route metrics, not link transmission speed.

routing loop. A situation that occurs when routers circulate information among themselves until convergence occurs or until the networks involved are considered unreachable.

routing protocol. A technique used by a router to find other routers and to remain up to date about the best way to get to reachable networks.

routing table. A collection of routes used to direct datagram forwarding or to establish a connection. The information is passed among routers to identify network topology and destination feasibility.

Routing Table Maintenance Protocol (RTMP). In AppleTalk networks, a protocol that provides routing information generation and maintenance on the transport layer by means of the AppleTalk routing table. The AppleTalk routing table directs packet transmission through the internet from source socket to destination socket.

RouTing update Protocol (RTP). The VIrtual NEtworking System (VINES) protocol that maintains the routing database and allows the exchange of routing information between VINES nodes. See also *Internet Control Protocol (ICP).*

rsh. A variant of the rlogin command that invokes a command interpreter on a remote UNIX machine and passes the command-line arguments to the command interpreter, skipping the login step completely.

S

SDU. Service Data Unit, data as it appears at the interface between a layer and the layer immediately above.

seed router. In AppleTalk networks, a router that maintains configuration data (network range numbers and zone lists, for example) for the network. Each network must have at least one seed router. The seed router must be initially set up using the configurator tool. Contrast with *nonseed router*.

segment. (1) A section of cable between components or devices. A segment may consist of a single patch cable, several patch cables that are connected, or a combination of building cable and patch cables that are connected. (2) In Internet communications, the unit of transfer between TCP functions in different machines. Each segment contains control and data fields; the current byte-stream position and actual data bytes are identified along with a checksum to validate received data.

segmenting. In OSI, a function performed by a layer to map one protocol data unit (PDU) from the layer it supports into multiple PDUs.

sequence number. In communications, a number assigned to a particular frame or packet to control the transmission flow and receipt of data.

server. A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server. (T)

service access point (SAP). (1) In Open Systems Interconnection (OSI) architecture, the point at which the services of a layer are provided by an entity of that layer to an entity of the next higher layer. (T) (2) A logical point made available by an adapter where information can be received and transmitted. A single service access point can have many links terminating in it.

Service Advertising Protocol (SAP). In Internetwork Packet Exchange (IPX), a protocol that provides the following:

- A mechanism that allows IPX servers on an internet to advertise their services by name and type. Servers using this protocol have their name, service type, and IP address recorded in all file servers running NetWare.
- A mechanism that allows a workstation to broadcast a query to discover the identities of all servers of all types, all servers of a specific type, or the nearest server of a specific type.
- A mechanism that allows a workstation to query any file server running NetWare to discover the names and addresses of all servers of a specific type.

session. (1) In network architecture, for the purpose of data communication between functional units, all the activities which take place during the establishment, maintenance, and release of the connection. (T) (2) A logical connection between two network accessible units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) accompanying any transmissions exchanged during the session.

Simple Network Management Protocol (SNMP). In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application's Management Information Base (MIB).

SLIP. Serial Line IP, an IETF standard for running IP over serial communication links.

SNA management services (SNA/MS). The services provided to assist in management of SNA networks.

SNAP. (1) SubNetwork Access Protocol. (2) SubNetwork Attachment Point.

socket. An endpoint for communication between processes or application programs.

source route bridging. In LANs, a bridging method that uses the routing information field in the IEEE 802.5 medium access control (MAC) header of a frame to determine which rings or token-ring segments the frame must transit. The routing information field is inserted into the MAC header by the source node. The information in the routing information field is derived from explorer packets generated by the source host.

source routing. In LANs, a method by which the sending station determines the route the frame will follow and includes the routing information with the frame. Bridges then read the routing information to determine whether they should forward the frame.

source service access point (SSAP). In SNA and TCP/IP, a logical address that allows a system to send data to a remote device from the appropriate communications support. Contrast with *destination service access point (DSAP)*.

spanning tree. In LAN contexts, the method by which bridges automatically develop a routing table and update that table in response to changing topology to ensure that there is only one route between any two LANs in the bridged network. This method prevents packet looping, where a packet returns in a circuitous route back to the sending router.

sphere of control (SOC). The set of control point domains served by a single management services focal point.

sphere of control (SOC) node. A node directly in the sphere of control of a focal point. A SOC node has exchanged management services capabilities with its focal point. An APPN end node can be a SOC node if it supports the function to exchange management services capabilities.

split horizon. A technique for minimizing the time to achieve network convergence. A router records the interface over which it received a particular route and does not propagate its information about the route back over the same interface.

spoofing. For data links, a technique in which a protocol initiated from an end station is acknowledged and processed by an intermediate node on behalf of the final destination. In IBM 6611 data link switching, for example, SNA frames are encapsulated into TCP/IP packets for transport across a non-SNA wide area network, unpacked by another IBM 6611, and passed to the final destination. A benefit of spoofing is the prevention of end-to-end session timeouts.

standard MIB. In the Simple Network Management Protocol (SNMP), a MIB module that is located under the management branch of the Structure of Management Information (SMI) and that is considered a standard by the Internet Engineering Task Force (IETF).

static route. The route between hosts, networks, or both that is manually entered into a routing table.

station. An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line.

StreetTalk. In the VIrtual NEtworking System (VINES), a unique network-wide naming and addressing system that allows users to locate and access any resource on the network without knowing the network topology. See also *Internet Control Protocol (ICP)* and *RouTing update Protocol (RTP)*.

Structure of Management Information (SMI). (1) In the Simple Network Management Protocol (SNMP), the rules used to define the objects that can be accessed by means of a network management protocol. (2) In OSI, the set of standards relating to management information. The set includes the *Management Information Model* and the *Guidelines for the Definition of Managed Objects*.

subarea. A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all network accessible units (NAUs), links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subnet. (1) In TCP/IP, a part of a network that is identified by a portion of the IP address. (2) Synonym for *subnetwork*.

subnet address. In Internet communications, an extension to the basic IP addressing scheme where a portion of the host address is interpreted as the local network address.

subnet mask. Synonym for *address mask*.

subnetwork. (1) Any group of nodes that have a set of common characteristics, such as the same network ID. (2) Synonymous with *subnet*.

Subnetwork Access Protocol (SNAP). In LANs, a 5-byte protocol discriminator that identifies the non-IEEE standard protocol family to which a packet belongs. The SNAP value is used to differentiate between protocols that use `$AA` as their service access point (SAP) value.

SubNetwork Attachment Point (SNAP). An LLC header extension that identifies the protocol type of a frame.

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)

SVN. Switched Virtual Networking, the name of IBM's framework for building and managing switch-based networks.

switched virtual circuit (SVC). An X.25 circuit that is dynamically established when needed. The X.25 equivalent of a switched line. Contrast with *permanent virtual circuit (PVC)*.

synchronous. (1) Pertaining to two or more processes that depend upon the occurrence of specific events such as common timing signals. (T) (2) Occurring with a regular or predictable time relationship.

Synchronous Data Link Control (SDLC). (1) A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. (I) (2) Contrast with *binary synchronous communication (BSC)*.

SYNTAX. In the Simple Network Management Protocol (SNMP), a clause in the MIB module that defines the abstract data structure that corresponds to a managed object.

system configuration. A process that specifies the devices and programs that form a particular data processing system.

system services control point (SSCP). A component within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory services and other session services for users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks. The layered structure of SNA allows the ultimate origins and destinations of information, that is, the users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

T

Telnet. In the Internet suite of protocols, a protocol that provides remote terminal connection service. It allows users of one host to log on to a remote host and interact as directly attached terminal users of that host.

threshold. (1) In IBM bridge programs, a value set for the maximum number of frames that are not forwarded across a bridge due to errors, before a “threshold exceeded” occurrence is counted and indicated to network management programs. (2) An initial value from which a counter is decremented to 0, or a value to which a counter is incremented or decremented from an initial value.

throughput class. In packet switching, the speed at which data terminal equipment (DTE) packets travel through the packet switching network.

time to live (TTL). A technique used by best-effort delivery protocols to inhibit endlessly looping packets. The packet is discarded if the TTL counter reaches 0.

timeout. (1) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event. (I) (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

TLV. Type/Length/Value, a generalized information element in a LAN Emulation packet.

token. (1) In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. Each data station has an opportunity to acquire and use the token to control the medium. A token is a particular message or bit pattern that signifies permission to transmit. (T) (2) In LANs, a sequence of bits passed from one device to another along the transmission medium. When the token has data appended to it, it becomes a frame.

token ring. (1) According to IEEE 802.5, network technology that controls media access by passing a token (special packet or frame) between media-attached stations. (2) A FDDI or IEEE 802.5 network with a ring topology that passes tokens from one attaching ring station (node) to another. (3) See also *local area network (LAN)*.

token-ring network. (1) A ring network that allows unidirectional data transmission between data stations, by a token passing procedure, such that the transmitted data return to the transmitting station. (T) (2) A network that uses a ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to

send can capture the token and insert data for transmission.

topology. In communications, the physical or logical arrangement of nodes in a network, especially the relationships among nodes and the links between them.

topology database update (TDU). A message about a new or changed link or node that is broadcast among APPN network nodes to maintain the network topology database, which is fully replicated in each network node. A TDU contains information that identifies the following:

- The sending node
- The node and link characteristics of various resources in the network
- The sequence number of the most recent update for each of the resources described.

trace. (1) A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A) (2) For data links, a record of the frames and bytes transmitted or received.

transceiver (transmitter-receiver). In LANs, a physical device that connects a host interface to a local area network, such as Ethernet. Ethernet transceivers contain electronics that apply signals to the cable and that sense collisions.

Transmission Control Protocol (TCP). A communications protocol used in the Internet and in any network that follows the U.S. Department of Defense standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet Protocol (IP) as the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transmission group (TG). (1) A connection between adjacent nodes that is identified by a transmission group number. (2) In a subarea network, a single link or a group of links between adjacent nodes. When a transmission group consists of a group of links, the links are viewed as a single logical link, and the transmission group is called a *multilink transmission group (MLTG)*. A *mixed-media multilink transmission group (MMMLTG)* is one that contains links of different medium types (for example, token-ring, switched SDLC, nonswitched SDLC, and frame-relay links). (3) In an APPN network, a single link between adjacent nodes. (4) See also *parallel transmission groups*.

transmission header (TH). Control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit*.

transparent bridging. In LANs, a method for tying individual local area networks together through the medium access control (MAC) level. A transparent bridge stores the tables that contain MAC addresses so that frames seen by the bridge can be forwarded to another LAN if the tables indicate to do so.

transport layer. In the Open Systems Interconnection reference model, the layer that provides a reliable end-to-end data transfer service. There may be relay open systems in the path. (T) See also *Open Systems Interconnection reference model*.

trap. In the Simple Network Management Protocol (SNMP), a message sent by a managed node (agent function) to a management station to report an exception condition.

tunneling. To treat a transport network as though it were a single communication link or LAN. See also *encapsulation*.

T1. In the United States, a 1.544-Mbps public access line. It is available in twenty-four 64-Kbps channels. The European version (E1) transmits 2.048 Mbps. The Japanese version (J1) transmits 1.544 Mbps.

U

UNI. User-Network Interface, the interface between user equipment and an ATM switch network.

universally administered address. In a local area network, the address permanently encoded in an adapter at the time of manufacture. All universally administered addresses are unique. Contrast with *locally administered address*.

User Datagram Protocol (UDP). In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

V

V.24. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE).

V.25. In data communication, a specification of the CCITT that defines the automatic answering equipment and parallel automatic calling equipment on the General Switched Telephone Network, including procedures for disabling of echo controlled devices for both manually and automatically established calls.

V.35. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) at various data rates.

V.36. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) at rates of 48, 56, 64, or 72 kilobits per second.

VCC. Virtual Channel Connection, a connection between parties.

VINES. VIrtual NEtworking System.

virtual circuit. (1) In packet switching, the facilities provided by a network that give the appearance to the user of an actual connection. (T) See also *data circuit*. Contrast with *physical circuit*. (2) A logical connection established between two DTEs.

virtual link. In Open Shortest Path First (OSPF), a point-to-point interface that connects border routers that are separated by a non-backbone transit area. Because area routers are part of the OSPF backbone, the virtual link connects the backbone. The virtual links ensure that the OSPF backbone does not become discontinuous.

Virtual Local Area Network (VLAN). A logical grouping of one or more LANs based on protocol and subnet and used to isolate network traffic within these groups.

VIrtual NEtworking System (VINES). The network operating system and network software from Banyan Systems, Inc. In a VINES network, virtual linking allows all devices and services to appear to be directly connected to each other, when they may actually be thousands of miles apart. See also *StreetTalk*.

virtual route (VR). (1) In SNA, either (a) a logical connection between two subarea nodes that is physically realized as a particular explicit route or (b) a logical connection that is contained wholly within a subarea node for intranode sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual route pacing, and provides data integrity through sequence numbering of path information units (PIUs). (2) Contrast with *explicit route (ER)*. See also *path* and *route extension (REX)*.

W

wide area network (WAN). (1) A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities. (T) (2) A data communication network designed to serve an area of hundreds or thousands of miles; for example, public and private packet-switching networks, and national telephone networks. (3) Contrast with *local area network (LAN)* and *metropolitan area network (MAN)*.

wildcard character. Synonym for *pattern-matching character*.

X

X.21. An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for a general-purpose interface between data terminal equipment and data circuit-terminating equipment for synchronous operations on a public data network.

X.25. (1) An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. (2) See also *packet switching*.

Xerox Network Systems (XNS). The suite of internet protocols developed by the Xerox Corporation. Although similar to TCP/IP protocols, XNS uses different packet formats and terminology. See also *Internetwork Packet Exchange (IPX)*.

Z

zone. In AppleTalk networks, a subset of nodes within an internet.

Zone Information Protocol (ZIP). In AppleTalk networks, a protocol that provides zone management service by maintaining a mapping of the zone names and network numbers across the internet on the session layer.

zone information table (ZIT). A listing of network numbers and their associated zone name mappings in

the internet. This listing is maintained by each internet router in an AppleTalk internet.

Index

Special Characters

Numerics

A

B

C

D

E

F

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Tell Us What You Think!

Multiprotocol Switched Services (MSS) Server
Command Line Interface Volume 2
User's Guide and Protocol Reference

Publication No. SC30-3819-01

We hope you find this publication useful, readable, and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form. If you are in the U.S.A., you can mail this form postage free or fax it to us at 1-800-253-3520. Elsewhere, your local IBM branch office or representative will forward your comments or you may mail them directly to us.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	☐	☐

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	☐	☐
Complete	☐	☐
Easy to find	☐	☐
Easy to understand	☐	☐
Well organized	☐	☐
Applicable to your task	☐	☐

Specific comments or problems:

Please tell us how we can improve this book:

Thank you for your comments. If you would like a reply, provide the necessary information below.

Name

Address

Company or Organization

Phone No.

Tell Us What You Think!
SC30-3819-01

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Design & Information Development
Dept. CGF/Bldg. 656
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

Fold and Tape **Please do not staple** Fold and Tape

SC30-3819-01

Cut or Fold
Along Line

Printed in USA

The server Library

GA27-4140	*IBM 8210 Nways Multiprotocol Switched Services (MSS) Server Setup and Problem Determination Guide*
GX27-4017	*IBM 8210 Nways Multiprotocol Switched Services (MSS) Server Operations Reference Card*
SC30-3821	*IBM Multiprotocol Switched Services (MSS) Server Configuration and Operations Guide*
GX27-4018	*IBM Nways Multiprotocol Switched Services (MSS) Server Module Reference Card*
GA27-4141	*IBM Nways Multiprotocol Switched Services (MSS) Server Module Setup and Problem Determination Guide*
GC30-3820	*IBM Multiprotocol Switched Services (MSS) Server Introduction and Planning Guide*
SC30-3818	*IBM Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1: User's Guide and Protocol Reference*
SC30-3819	*IBM Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 2: User's Guide and Protocol Reference*
SC30-3682	*Event Logging System Messages Guide*
GY27-0354	*IBM 8210 Nways Multiprotocol Switched Services (MSS) Server Service Manual*

SC30-3819-01