

Distributed Routing Software

Bridging Configuration Guide

Part Number: AA-QL29E-TE

December 1997

This manual includes information on bridging methods, operational features of bridging, configuration methods and basic configurations, and monitoring of your bridging software.

Revision/Update Information: This is a revised manual.
Software Version: Distributed Routing Software V3.0

**Digital Equipment Corporation
Maynard, Massachusetts**

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from DIGITAL or an authorized sublicensor.

© Digital Equipment Corporation 1997
All Rights Reserved.
Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation: DEC, DECnet, DECswitch, DIGITAL, OpenVMS, PATHWORKS, RouteAbout, ThinWire, VAX, VAXcluster, VMS, VT, and the DIGITAL logo.

The following are third-party trademarks:

Apollo is a registered trademark of Apollo Computer, Inc., a subsidiary of Hewlett-Packard Company.

AppleTalk, EtherTalk, and LocalTalk are registered trademarks of Apple Computer, Inc.

Banyan and Vines are registered trademarks of Banyan Systems, Inc.

BSD is a trademark of the University of California, Berkeley, CA.

IBM is a registered trademark of International Business Machines Corporation.

Intel is a registered trademark of Intel Corporation.

Lotus Notes is a registered trademark of Lotus Development Corporation.

MS-DOS and Windows 95 are registered trademarks, and Windows NT is a trademark of Microsoft Corporation.

NetBIOS is a trademark of Micro Computer Systems, Inc.

NetWare and Novell are registered trademarks of Novell, Inc.

Proteon, ProNET, and TokenVIEW are registered trademarks of Proteon, Inc.

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

All other trademarks and registered trademarks are the property of their respective holders.

This manual was produced by DIGITAL Information Design.

Contents

Preface

Objectives	xi
Audience	xi
Using This Guide	xii
Using Related Documentation	xiv
DIGITAL Documents	xiv
Document Set Structure	xv
Conventions	xvii
Symbols	xviii
Commands	xix
Entering Commands	xx
Accepting the Current Setting	xx
Correspondence	xxi
Documentation Comments	xxi
Online Services	xxi
How to Order Additional Documentation	xxii

1 Bridging Basics

1.1	Bridging Overview	1-1
1.2	Bridges versus Routers	1-2
1.2.1	Router Connections	1-3
1.2.2	Bridge Connections	1-3
1.3	Types of Bridges	1-4
1.3.1	Simple Bridges	1-4
1.3.2	Complex Bridges	1-4
1.3.3	Local Bridges	1-5
1.3.4	Remote Bridges	1-5
1.4	Basic Bridge Operation	1-5
1.4.1	Operation Example 1: Local Bridge Connecting Two LANS	1-5
1.4.2	Operation Example 2: Remote Bridging Over a Serial Link	1-6
1.4.3	Data Encapsulation Scheme	1-7
1.5	MAC Bridge Frame Formats	1-8

1.5.1	CSMA/CD (Ethernet) MAC Frames	1-9
1.5.2	Token Ring MAC Frames	1-10

2 Bridging Methods

2.1	Transparent Bridge (STB)	2-1
2.1.1	Routers and Transparent Bridges	2-2
2.1.2	Network Requirements	2-2
2.1.3	Transparent Bridge Operation	2-3
2.1.4	Shaping the Spanning Tree	2-4
2.1.5	Spanning Tree Bridges and Ethernet Packet Format Translation	2-7
2.1.6	Transparent Bridge Terminology and Concepts	2-8
2.2	Source Routing Bridge (SRB)	2-12
2.2.1	Source Routing Bridge Operation	2-14
2.2.2	Source Routing Frames	2-14
2.2.3	The Spanning Tree Explore Option	2-18
2.2.3.1	Simulating a Spanning Tree Network	2-19
2.2.3.2	Balancing Traffic Loads	2-19
2.2.4	Protocol Filtering	2-20
2.2.5	Source Routing Bridge Terminology and Concepts	2-21
2.3	Source Routing Transparent Bridge (SRT)	2-22
2.3.1	General Description	2-23
2.3.2	Source Routing Transparent Bridge Operation and Architecture	2-24
2.3.3	Source Routing Transparent Bridge Terminology	2-25
2.4	Adaptive Source Routing Transparent Bridge (SR-TB) Conversion	2-26
2.4.1	General Description	2-27
2.4.2	Source Routing – Transparent Bridge Operation	2-28
2.4.2.1	Specific Source Routing and Transparent Bridging Operations	2-29
2.4.2.2	SR-TB Bridging: Four Examples	2-31
2.4.2.3	Example 1: Frame Sent from End Station A to End Station B	2-32
2.4.2.4	Example 2: Frame Sent from End Station A to End Station C	2-33
2.4.2.5	Example 3: Frame Sent from End Station C to End Station D	2-34
2.4.2.6	Example 4: Frame Sent from End Station C to End Station A	2-34
2.4.3	Source Routing – Transparent Bridge Terminology and Concepts	2-34
2.5	ASRT Bridge Overview	2-36
2.5.1	Transparent and Source-Routing Compatibility – Issues and Solutions	2-36
2.5.1.1	Elimination of Packet Size Problems	2-37
2.5.1.2	Bit Ordering in STB (802.3) and SRB (802.5) Bridges	2-37
2.5.2	ASRT Configuration Considerations	2-38

3 Bridging Features

3.1	Bridging Tunnel	3-1
3.1.1	Encapsulation and IP Routing Protocols	3-2

3.2	TCP/IP Host Services (Bridge-Only Management)	3-3
3.3	Bridge-MIB Support	3-4
3.4	Pseudo Serial Ethernet	3-4
3.5	Multiple Spanning Tree Protocol Options	3-4
3.5.1	Background: Problems with Multiple Spanning Tree Protocols	3-5
3.5.2	Spanning Tree Protocol and 8209 Bridging (STP/8209).	3-5
3.5.3	Logical Link Class 2 Support.	3-6
3.6	Threading	3-7
3.6.1	IP Threading with ARP.	3-7
3.6.2	DECnet (DNA) Threading.	3-8
3.6.3	IPX Threading	3-9
3.6.4	AppleTalk 1 and 2 Threading	3-9

4 Configuring and Monitoring ASRT

4.1	Accessing the ASRT Environments	4-1
4.1.1	ASRT Configuration Commands.	4-1
4.1.2	ASRT Console Commands	4-2
4.2	LNM Configuration Commands	4-58
4.3	Tunnel Configuration Commands.	4-63
4.3.1	Tunneling and Multicast Packets.	4-63

5 Basic Bridging Configurations

5.1	Accessing the ASRT Configuration Environment	5-1
5.2	Basic Bridging Configuration Procedures.	5-1
5.2.1	Bridging Interfaces.	5-2
5.2.2	Enabling the Transparent Bridge	5-3
5.2.3	Enabling the Source Routing Bridge	5-3
5.2.4	Enabling the SR-TB Bridge	5-3
5.3	ASRT Configuration Matrix	5-4
5.3.1	Obtain the MAC Address Required for Network Manager Software	5-5
5.3.2	Enable the LNM Agents on the Bridging Router	5-6
5.3.3	Check the Configuration by Displaying Enabled LNM Agents	5-6

6 Configuring and Monitoring TCP/IP Host Services

6.1	Accessing the TCP/IP Host Configuration and Console Environments	6-1
6.2	Basic Configuration Procedures	6-1
6.2.1	Setting the IP Address	6-1
6.2.2	Adding a Default Gateway.	6-1
6.2.3	Enabling TCP/IP Host Services.	6-2
6.3	TCP/IP Host Configuration and Console Commands.	6-2

7 Using MAC Filtering

7.1	About MAC Filtering	7-1
7.2	Using MAC Filtering Parameters	7-2
7.2.1	Filter-Item Parameters	7-2
7.2.2	Filter-List Parameters	7-2
7.2.3	Filter Parameters	7-3
7.3	Using MAC Filtering Tags	7-3
7.4	Configuring and Monitoring MAC Filtering	7-4
7.4.1	Accessing the MAC Filtering Prompts	7-4
7.4.2	Accessing the MAC Filtering Configuration Commands	7-5
7.5	Using the MAC Filtering Console Commands	7-17

8 Configuring and Monitoring IP Tunneling

8.1	Accessing the IP Tunnel Commands	8-1
8.1.1	Tunneling and Multicast Packets	8-2

9 Using NetBIOS

9.1	About NetBIOS	9-1
9.1.1	NetBIOS Names	9-1
9.1.2	NetBIOS Name Conflict Resolution	9-2
9.1.3	NetBIOS Session Setup Procedure	9-2
9.2	Reducing NetBIOS Traffic	9-2
9.2.1	Frame Type Filtering	9-3
9.2.1.1	Name Conflict Resolution	9-3
9.2.1.2	General Broadcast	9-4
9.2.1.3	Trace Control Frames	9-4
9.2.1.4	Configuring Frame Type Filtering	9-4
9.2.2	Duplicate Frame Filtering	9-4
9.2.2.1	How Duplicate Frame Filtering Works	9-5
9.2.2.2	Configuring Duplicate Frame Filtering	9-5
9.2.3	Response Frame Filtering	9-6
9.2.4	NetBIOS Name Caching and Route Caching	9-6
9.2.4.1	Enabling Caching	9-6
9.2.4.2	Types of Name Cache Entries	9-7
9.2.4.3	Setting Cache Parameters	9-8
9.2.4.4	Displaying Cache Entries	9-8
9.2.5	NetBIOS Name Filtering	9-9
9.2.6	NetBIOS Byte Filtering	9-9

10 NetBIOS Filtering and Caching Commands

10.1	About NetBIOS Configuration and Console Commands	10-1
10.2	Displaying the NetBIOS Configuration Prompt	10-1
10.3	Displaying the NetBIOS Console Prompt	10-2
10.4	Configuring NetBIOS Filtering and Caching	10-2
10.5	NetBIOS Commands	10-2

11 Configuring and Monitoring NetBIOS Name and Byte Filters

11.1	Displaying the NetBIOS Filtering Prompts	11-1
11.2	Setting Up NetBIOS Name and Byte Filters	11-2
11.2.1	Example 1: Creating a Name Filter	11-2
11.2.2	Example 2: Creating a Byte Filter	11-5
11.3	NetBIOS Name and Byte Filter Commands	11-9
11.4	Update Byte-Filter-List Commands	11-16
11.5	Update Name-Filter-List Commands	11-19

Index	Index-1
--------------------	---------

Figures

1	Document Set Structure	xvi
2	Command Components	xix
3	Set Framesize Command	xx
1-1	Simple and Complex Bridging Configuration	1-2
1-2	Two Port Bridge Connecting Two LANs.	1-6
1-3	Bridging Over a Point-to-Point Link	1-7
1-4	Data Encapsulation Over a Point-to-Point Link	1-7
1-5	Examples of MAC Frame Formats.	1-9
2-1	Networked LANs Before Spanning Tree	2-5
2-2	Spanning Tree Created with Default Values	2-6
2-3	User-Adjusted Spanning Tree	2-7
2-4	Example of Source Routing Bridge Connectivity	2-13
2-5	802.5 Source Address Format	2-15
2-6	802.5 Routing Information Field.	2-16
2-7	Example of Parallel Bridges.	2-19
2-8	Using Spanning Tree Explore for Load Balancing	2-19
2-9	Bridge Instances within a Bridge	2-21
2-10	SRT Bridge Operation	2-25
2-11	SR-TB Bridge Connecting Two Domains.	2-28
2-12	SR-TB Bridging Examples	2-32
3-1	Example of the Bridge Tunnel Feature	3-2
9-1	Duplicate Frame Filtering Process for Bridged Traffic	9-5

Tables

2-1	Route/Bridge Decision Table	2-20
2-2	SR-TB Bridge Decision Table	2-30
4-1	ASRT Configuration and Console Command Summary	4-2
4-2	LNLM Configuration Commands	4-58
4-3	Tunnel Configuration Commands	4-63
5-1	Bridge Configuration Settings Matrix	5-4
6-1	TCP/IP Host Configuration and Console Commands Summary	6-2
7-1	MAC Filtering Configuration Command Summary	7-5
7-2	MAC Filtering Update Subcommand Summary	7-13
7-3	MAC Filtering Console Command Summary	7-17
8-1	Tunnel Configuration and Console Command Summary	8-1
9-1	NetBIOS Filters	9-3
9-2	NetBIOS List Cache Configuration Commands	9-8
9-3	NetBIOS List Cache Monitoring Commands	9-8
10-1	NetBIOS Commands	10-2
11-1	NetBIOS Name and Byte Filter Commands	11-9

Preface

Objectives

This guide contains the information you need to configure your bridging router. This manual includes information on bridging methods, operational features of bridging, configuration methods and basic configurations, and monitoring of your bridging software.

Audience

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

Using This Guide

The following table helps you locate information in this guide.

If You Want Information About...	See Chapter or Appendix...
<ul style="list-style-type: none">• Summary of document contents• Related documentation• Document set structure• Documentation Conventions	Preface
<ul style="list-style-type: none">• Bridging Overview• Bridges vs. Routers• Types of Bridges• Basic Bridge Operation• MAC Bridge Frame Formats	1 Bridging Basics
<ul style="list-style-type: none">• Transparent Bridge (STB)• Source Routing Bridge (SRB)• Source Routing Transparent Bridge (SRT)• Adaptive Source Routing Transparent Bridge (ASRT)• ASRT Bridge Overview	2 Bridging Methods
<ul style="list-style-type: none">• Bridging Tunnel• TCO/IP Host Services• Bridge-MIB Support• Pseudo Serial Ethernet• Multiple Spanning Tree Protocol Options• Threading	3 Bridging Features
<ul style="list-style-type: none">• Accessing the ASRT Environments• LNM Configuration Commands• Tunnel Configuration Commands	4 Configuring and Monitoring ASRT

If You Want Information About...	See Chapter or Appendix...
<ul style="list-style-type: none"> • Accessing the ASRT Configuration Environment • Basic Bridging Configuration Procedures • ASRT Configuration Matrix • Basic LNM Configuration 	5 Basic Bridging Configurations
<ul style="list-style-type: none"> • Accessing the TCP/IP Configuration and Console Processes • Basic Configuration Procedures • TCP/IP Configuration and Console Commands 	6 Configuring and Monitoring TCP/IP Host Services
<ul style="list-style-type: none"> • About MAC Filtering • Using MAC Filtering Parameters • Using MAC Filtering Tags • Configuring and Monitoring MAC Filtering • Using the MAC Filtering Console Commands 	7 Using MAC Filtering
<ul style="list-style-type: none"> • Accessing the IP Tunnel Commands 	8 Configuring and Monitoring IP Tunneling
<ul style="list-style-type: none"> • About NetBIOS • Reducing NetBIOS Traffic 	9 Using NetBIOS
<ul style="list-style-type: none"> • About NetBIOS Configuration and Console Commands • Displaying the NetBIOS Configuration Prompt • Displaying the NetBIOS Console Prompt • Configuring NetBIOS Filtering and Caching • NetBIOS Commands 	10 NetBIOS Filtering and Caching Commands
<ul style="list-style-type: none"> • Displaying the NetBIOS Filtering Prompts • Setting Up NetBIOS Name and Byte Filters 	11 Configuring and Monitoring NetBIOS Name and Byte Filters

Using Related Documentation

DIGITAL Documents

This Document...	Describes...
<i>RouteAbout Access EI Installation</i> EK-DEXBR-IN	Installation and use of the RouteAbout Access EI router.
<i>RouteAbout Access EW Installation</i> EK-DEX2R-IN	Installation and use of the RouteAbout Access EW router.
<i>RouteAbout Access TW Installation</i> EK-DEWTR-IN	Installation and use of the RouteAbout Access TW router.
<i>RouteAbout Central EI Installation</i> EK-DEZBR-IN	Installation and use of the RouteAbout Central EI router.
<i>RouteAbout Central EP Installation</i> EK-DEZPR-IN	Installation and use of the RouteAbout Central EP router.
<i>RouteAbout Central EW Installation</i> EK-DEZ8R-IN	Installation and use of the RouteAbout Central EW router.
<i>clearVISN Router Configurator</i> <i>User's Guide</i> AA-R08YB-TE	The graphic user interface application which enables you to create and load a basic configuration for the bridging router.
<i>DTF (DIGITAL Trace Facility) User</i> <i>Guide</i> AA-R85DA-TE	How to install and use the DIGITAL Trace Facility, which enables you to trace packets within the protocol layers of the bridging router.
<i>Event Logging System Messages</i> <i>Guide</i> AA-QL2AE-TE	How events are logged, how to interpret Event Logging System (ELS) messages. Provides a description of each ELS message with a corresponding corrective action.
<i>Network Interface Operations Guide</i> AA-QL2BE-TE	Configuring and monitoring the network interfaces in the Distributed Routing Software bridging router.

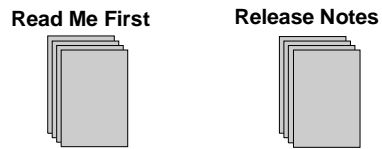
This Document...	Describes...
<i>Quick Reference Guide</i> AA-R7QAA-TE	How to configure and monitor the main protocols, features and interfaces, and lists the associated commands.
<i>Routing Protocols Reference Guide</i> AA-QL2CE-TE	Reference information about the micro-operating system structure, and the protocols and interfaces supported by bridging routers.
<i>Routing Protocols User's Guide</i> AA-QL2DE-TE	Configuring and monitoring the protocols in the Distributed Routing Software bridging router. How to use the DIGITAL Trace Facility.
<i>Systems Network Architecture Guide</i> AA-QU5SC-TE	SNA interfaces and protocols for the Distributed Routing Software System.
<i>System Software Guide</i> AA-QL2EE-TE	Installing, configuring, and operating the Distributed Routing Software system software.

Document Set Structure

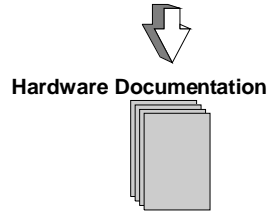
Figure 1 shows the structure of the documentation set.

Figure 1 Document Set Structure

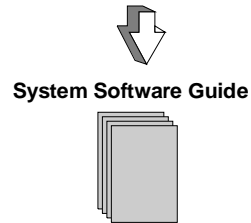
Start



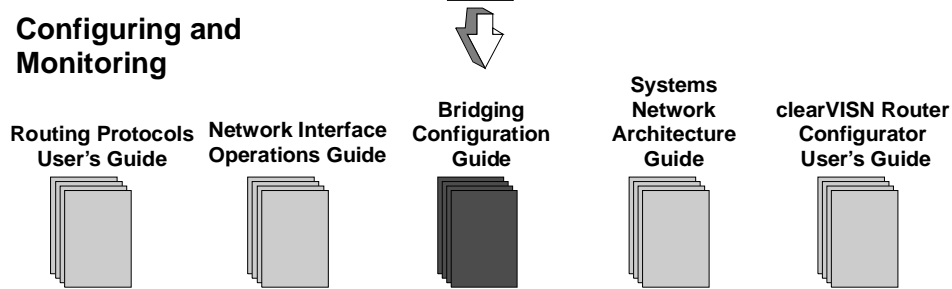
Installation



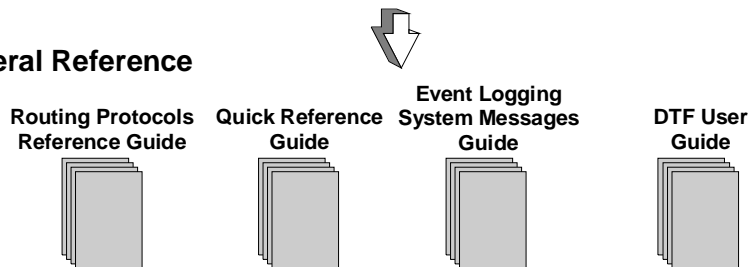
Router Operating System



Configuring and Monitoring



General Reference



Conventions

The following conventions are used in this manual:

Monospace type	Monospace type in examples indicates system output or user input.
Boldface type	Boldface type in examples indicates user input. Boldface type is also used for file names and command names within text.
<i>lowercase italics</i>	Lowercase italics in command syntax or examples indicate variables for which either the user or the system supplies a value.
[]	Brackets enclose operands or symbols that are either optional or conditional. Specify the operand and value if you want the condition to apply. Do not type the brackets in the line of code.
key	A key name in bold type indicates that you press the specified key.
Ctrl/X	Indicates that you hold the Ctrl key while pressing the key specified by the X. The server displays the key combination as ^X.
<u>underscore</u>	Characters underlined in a command listing represent the fewest number of characters you must enter to identify that command to the interpreter.
2-3	In the Index, page reference numbers in bold type indicate a reference to a command description.

Symbols



The configuring and monitoring chapters contain a description of all commands you can use to configure and monitor the protocol, feature, or interface.

C means you use the command to configure the router. You access configuration commands after you enter **talk 6** at the * prompt. Configuration commands change the router's nonvolatile database; a router restart is necessary to activate the change.

M means you use the command to monitor and dynamically configure the router. You access monitoring commands after you enter **talk 5** at the * prompt. Changes made in this mode take effect immediately, but are not made in the router's nonvolatile database (and therefore not preserved after a router restart).

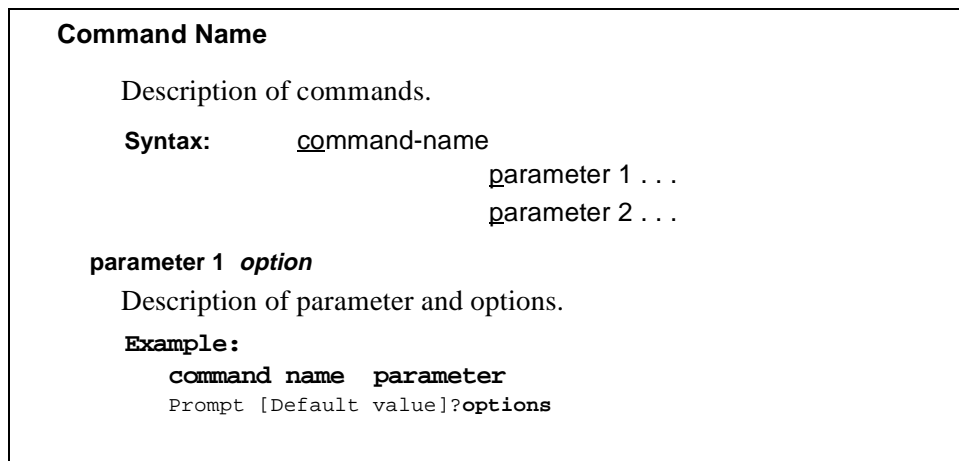
C M means you use the command both to configure and to monitor the router.

Note: **Talk 5** monitoring commands are also referred to as console commands in this guide. **Talk 6** configuration commands are sometimes referred to as config commands.

Commands

Figure 2 shows the components of a command description.

Figure 2 Command Components



Command Name	The name of the command followed by an overview description.
Syntax:	The command followed by each parameter you can configure using that command. If an ellipsis follows a parameter, you need to enter additional information (<i>options</i>). When you enter a command, you can save time by typing only the underlined letters.
parameter	Description of each parameter.
<i>option</i>	(in italics) Information you must enter with the command and parameter.
Example:	An example of how you enter that command and parameter.

Entering Commands

Instead of being prompted for options, you can save time by entering the complete command on one line. For example, you can enter the **set framesize** command shown in Figure 3 as follows:

```
set framesize 2048
```

If you abbreviate the command using the underlined letters, you can enter

```
s f 2048
```

Figure 3 Set Framesize Command

Set

Configures frame size and local address.

Syntax: set framesize . . .
 parameter 2 . . .

framesize 1024 or 2048 or 4096

The size of the network-layer portion of frames transmitted and received on the interface.

Example:

```
set framesize  
Framesize in bytes (1024/2048/4096) [1024]? 2048
```

Accepting the Current Setting

When the software prompts you for information, the current setting appears in brackets []. To accept the information in the brackets, press **Return**. In this example, the current setting is 1024.

```
Framesize in bytes (1024/2048/4096) [1024]?
```

Correspondence

Documentation Comments

If you have comments or suggestions about this document, send them to the DIGITAL Network Products Business Organization.

Attn: Documentation Project Manager

E-MAIL: doc_quality@lkg.mts.dec.com

Online Services

To locate product-specific information, refer to the DIGITAL Network Products Home Page on the World Wide Web, located at the following addresses:

North America <http://www.networks.digital.com>

Europe <http://www.networks.europe.digital.com>

Asia Pacific <http://www.networks.digital.com.au>

How to Order Additional Documentation

To order additional documentation, use the following information:

To Order:	Contact:
By Telephone	USA (except Alaska, New Hampshire, and Hawaii): 1-800-DIGITAL (1-800-344-4825) Alaska, New Hampshire, and Hawaii: 1-603-884-6660 Canada: 1-800-267-6215
Electronically (USA only)	Dial 1-800-DEC-DEMO (For assistance, call 1-800-DIGITAL)
By Mail (USA and Puerto Rico)	DIGITAL EQUIPMENT CORPORATION P.O. Box CS2008 Nashua, New Hampshire 03061 (Place prepaid orders from Puerto Rico with the local DIGITAL subsidiary: 809-754-7575)
By Mail (Canada)	DIGITAL EQUIPMENT of CANADA LTD. 940 Belfast Road Ottawa, Ontario, Canada K1G 4C2 Attn.: A&SG Business Manager
Internationally	DIGITAL EQUIPMENT CORPORATION Attn.: A&SG Business Manager c/o local DIGITAL subsidiary or approved distributor
Internally	U.S. Software Supply Business (SSB) DIGITAL EQUIPMENT CORPORATION 8 Cotton Road Nashua, New Hampshire 03063

Bridging Basics

This chapter discusses basic information about bridges and bridging operation.

1.1 Bridging Overview

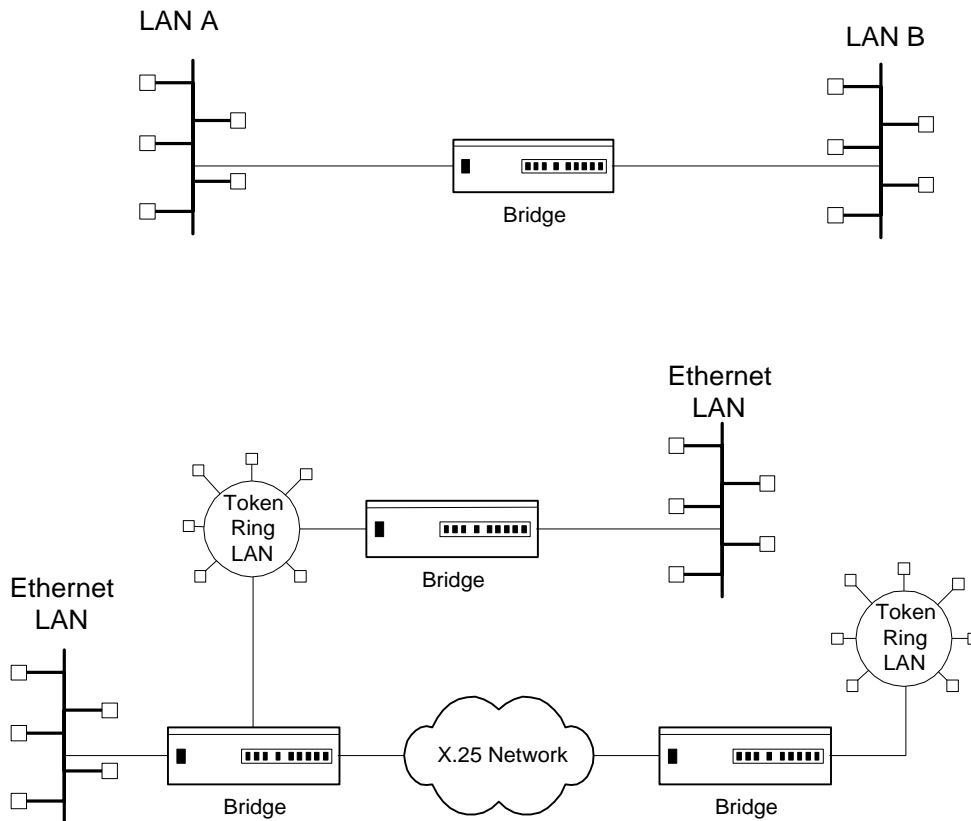
A bridge is a device that links two or more local area networks. The bridge accepts data frames from each connected network and then decides whether to forward each frame based on the MAC header contained in the frame. Bridges originally linked two or more homogeneous networks. The term *homogeneous*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

Current bridges also allow communication between non homogeneous networks. *Non homogeneous* refers to networks that may mix different bridging methods or different media types and may also offer more configuration options.

Figure 1–1 illustrates examples of simple and complex bridging configurations.

1.2 Bridges versus Routers

Figure 1–1 Simple and Complex Bridging Configuration



LKG-10570-97V

1.2 Bridges versus Routers

Internetworking devices such as bridges and routers connect network segments. However, each device uses a different method to establish and maintain their 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).

1.2 Bridges versus Routers

1.2.1 Router Connections

Network layer protocols efficiently move information in large and diverse network configurations. Connecting at the Network Layer (Layer 3) with a router allows connectivity and path selection between end stations at large geographical distances. Routing protocols are then used to select the best path for connecting distant and diverse LANs. Because a large variety of network and subnetwork configurations exist in networks of this type, connecting LANs through the Network Layer is usually the preferred method.

1.2.2 Bridge Connections

Connecting a bridge at Layer 2 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 the “Bridging Methods” chapter in this guide for more information on source routing and transparent bridges.

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

Bridges can also filter frames based on Layer 2 fields. This means that the bridge may be configured to accept and forward 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 used to divide large networks into manageable segments. The advantages of bridging are summed up as follows:

- Bridging lets you transparently extend the connectivity between systems attached to LAN segments by forming a single *extended LAN*.
- Bridging lets you form an extended LAN from segments of different media types.
- Bridging increases the effective throughput capacity of the extended LAN by automatically restricting traffic to only those segments that need to see it.

1.3 Types of Bridges

- Configurable filters let you regulate the amount of traffic that is forwarded to a specific segment.
- Bridging partitions the extended LAN into separate collision domains for each segment, effectively increasing the throughput capacity. (A collision domain is a shared physical medium to which systems must contend for access.)
- Bridging allows interconnection between systems using nonroutable protocols such as 3 LAT when those systems are attached to different LAN segments.

1.3 Types of Bridges

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

1.3.1 Simple Bridges

Figure 1–1 illustrates simple bridges consisting of two or more linked network interfaces connecting Local Area Networks (LANs). Bridges interconnect separate Local Area Networks by relaying data frames between the separate MAC (Media Access Control) entities of the bridged LANs.

The main functions of a simple bridge are summarized as follows:

- The bridge receives all frames transmitted on each of LAN A and LAN B.
- It forwards frames from the input LAN to the output LAN if the destination MAC Address of the frame is known to be attached to the output LAN (bridges can automatically learn on which LAN a MAC Address exists).

1.3.2 Complex Bridges

Complex bridges carry out more sophisticated functions than simple bridges. These functions 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. This type of exchange permits 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.

1.4 Basic Bridge Operation

The Adaptive Source Routing Transparent (ASRT) Bridge is a software collection capable of several of the bridging options just described and more. These functions are explained in greater detail later in this chapter.

1.3.3 Local Bridges

Local bridges provide connections between 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.

1.3.4 Remote Bridges

Remote bridges connect multiple LAN segments in different geographical areas. An example of this is bridges used to 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 the data transmission speed. WAN connections are generally slower than LAN connections. This speed difference can be significant when running time-sensitive applications. Another difference is the physical connections between remote and local bridges and LANs. In local bridges, the connection is made through local cabling media (for example, Ethernet). Remote bridge connections are made over serial lines.

1.4 Basic Bridge Operation

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

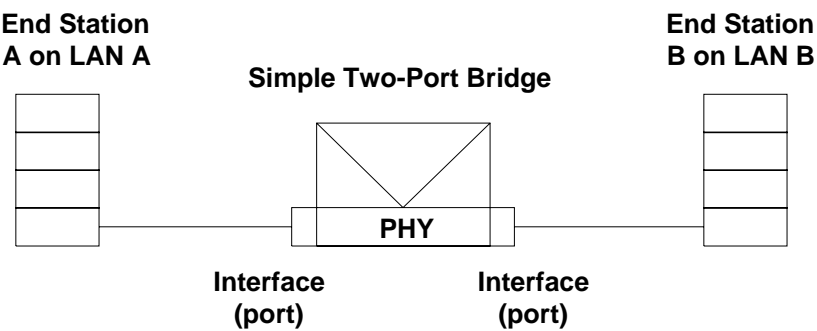
1.4.1 Operation Example 1: Local Bridge Connecting Two LANS

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

1.4 Basic Bridge Operation

The bridge captures MAC frames whose destination addresses are not on the local LAN (the LAN connected to the interface receiving the transmitted frame). It then forwards them to the appropriate destination LAN.

Figure 1–2 Two Port Bridge Connecting Two LANs



LKG-10571-97V

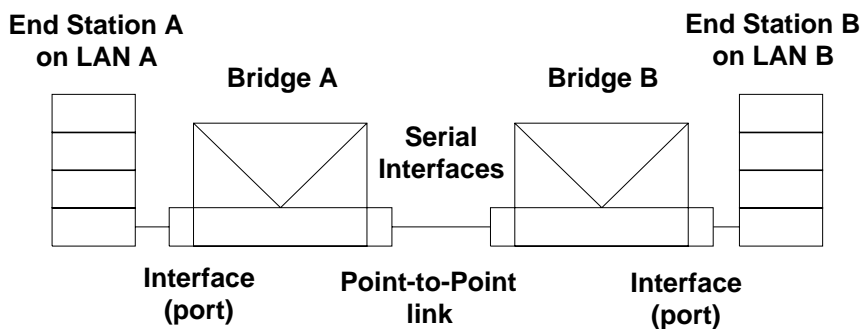
1.4.2 Operation Example 2: Remote Bridging Over a Serial Link

Figure 1–3 shows a pair of bridges connected over a serial link. These remote bridges connect LANs with identical LLC and MAC layers (two Ethernet LANs).

The bridge captures a MAC frame whose destination address is not on the local LAN and sends it to the appropriate destination LAN by way of the point-to-point link.

1.4 Basic Bridge Operation

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

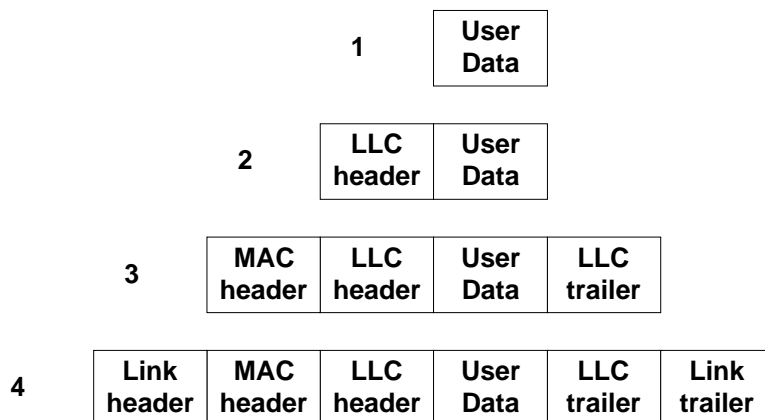


LKG-10572-97V

1.4.3 Data Encapsulation Scheme

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



LKG-10573-97V

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.

1.5 MAC Bridge Frame Formats

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, the bridge appends a Link Layer (for example, PPP and HDLC) header and trailer and transmits the point-to-point frame across the link.

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

1.5 MAC Bridge Frame Formats

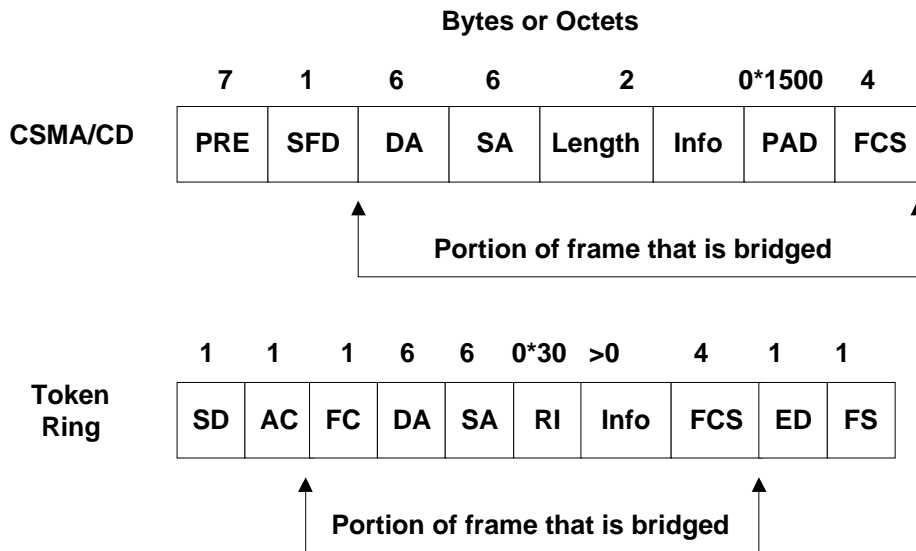
Bridges interconnect LANs by relaying data frames, specifically MAC frames, between the separate MAC entities of the bridged LANs. MAC frames provide the necessary “Where?” information for frame forwarding in the form of source and destination addresses.

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.

1.5 MAC Bridge Frame Formats

Figure 1–5 Examples of MAC Frame Formats



<p>AC = Access Control DA = Destination Address ED = Ending Delimiter FC = Frame Control FCS = Frame Check Status FS = Frame Status</p>	<p>PRE = Preamble RI = Routing Information SA = Source Address SD = Starting Delimiter SFD = Starting Frame Delimiter</p>
--	--

LKG-10574-97V

1.5.1 CSMA/CD (Ethernet) MAC Frames

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

- **Preamble (PRE)** – 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.

1.5 MAC Bridge Frame Formats

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 either 16- or 48-bit (2 or 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.

1.5.2 Token Ring MAC Frames

The following information describes each of the fields found 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 whether this is an LLC data frame. If it is 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.

1.5 MAC Bridge Frame Formats

Both address lengths carry a Routing Information Indicator (RII) bit that indicates whether a Routing Information Field (RIF) is present in the frame. When the RII bit is set to 1, it indicates that a Routing Information Field (RIF) is present after the source address. When the RII bit is set to 0, it indicates that a Routing Information Field (RIF) is not present after the source address. This field is explained in more detail in Chapter 2, section 2.2.

- **Routing Information Field (RIF)** – When the RII in the source address field is set to 1, this field is present after the source address. 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 Chapter 2, section 2.2.
- **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.
- **End Delimiter (ED)** – Contains the error detection (E) bit, and the intermediate frame (I) bit. The I bit indicates that this is the frame other than the final one of a multiple frame transmission.
- **Frame Status (FS)** – Contains the address recognized (A) and frame copied (C) bits.

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.

2.1 Transparent Bridge (STB)

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 nonlocal traffic to attached LANs in a way that is transparent to or unseen by the originating and receiving systems. End station applications do not know about the presence of the bridge. The bridge learns about the presence of end stations by listening to traffic. From this listening process, it builds a database of addresses of end stations 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 logical loop-free topology in a bridged network that contains 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 LAN's parallel bridges. This creates a redundancy in data traffic and produces the phenomenon known as looping.

2.1 Transparent Bridge (STB)

When looping occurs, you must reconfigure the local or remote LAN, or both, 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 algorithm transparently configures all bridges on the LAN into a single loop-free spanning tree.

In order to eliminate data loops, the spanning tree only has one active path between two end stations. To create a loop-free topology, the algorithm determines which bridge ports can forward data and which ones are blocked. Spanning tree provides the following features:

- **Loop detection** – Detects and eliminates looping in extended LAN configurations.
- **Automatic backup of data paths** – The bridges connecting to the redundant paths enter backup mode automatically. When a primary path fails, a backup path 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 determine the shape of the spanning tree.

2.1.1 Routers and Transparent Bridges

A system may be capable of performing both routing and bridging functions concurrently. In this mode, the router is called a *brouter*. During this mode of operation, the following occurs:

- Packets are routed if the packet's protocol was enabled for routing.
- Packets are filtered at the MAC layer if you configure specific bridging filters.
- Packets that are not routed or filtered are candidates for bridging, depending on the destination MAC (Media Access Control) address.

2.1.2 Network Requirements

Transparent Bridge implements a spanning tree bridge that conforms to the IEEE 802.1D standard. All Ethernet bridges on the network must be 802.1D spanning tree bridges. This spanning tree protocol is not compatible with bridges implementing LANbridge 100 spanning tree protocol from Digital Equipment Corporation.

2.1 Transparent Bridge (STB)

2.1.3 Transparent Bridge Operation

The spanning tree protocol specifies that 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 path costs to the root through 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 highest priority ID (lowest 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 equal path costs, the bridge with the lowest bridge ID value is selected as the designated bridge.
3. The nondesignated 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 failures) but it does not forward any data packets.

2.1 Transparent Bridge (STB)

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.

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.

2.1.4 Shaping the Spanning Tree

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

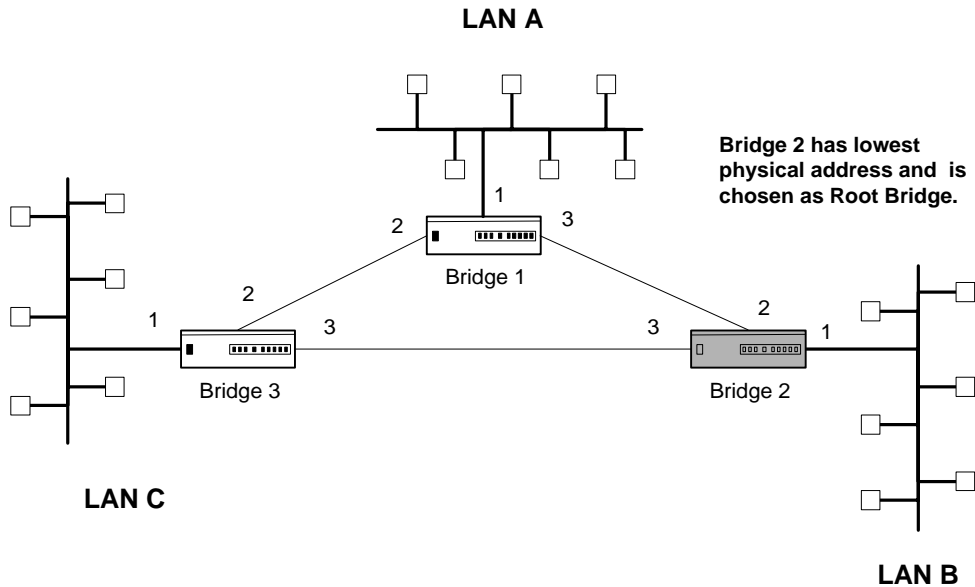
Figure 2–1 shows three LANs networked using three bridges. Each bridge is using default bridge priority settings for its spanning tree configuration. In this case, the bridge with the lowest physical address is chosen as the root bridge since the bridge priority of each bridge is the same. In the figure, 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.

2.1 Transparent Bridge (STB)

Figure 2–1 Networked LANs Before Spanning Tree

Bridge 1 Bridge priority:32768 Address: 08:00:2B:00:00:10	Bridge 2 Bridge priority:32768 Address: 08:00:2B:00:00:01	Bridge 3 Bridge priority:32768 Address: 08:00:2B:00:00:05
Port 1 Priority: 128 Path cost: 100	Port 1 Priority: 128 Path cost: 100	Port 1 Priority: 128 Path cost: 100
Port 2 Priority: 128 Path cost: 17857	Port 2 Priority: 128 Path cost: 17857	Port 2 Priority: 128 Path cost: 17857
Port 3 Priority: 128 Path cost: 17857	Port 3 Priority: 128 Path cost: 17857	Port 3 Priority: 128 Path cost: 17857



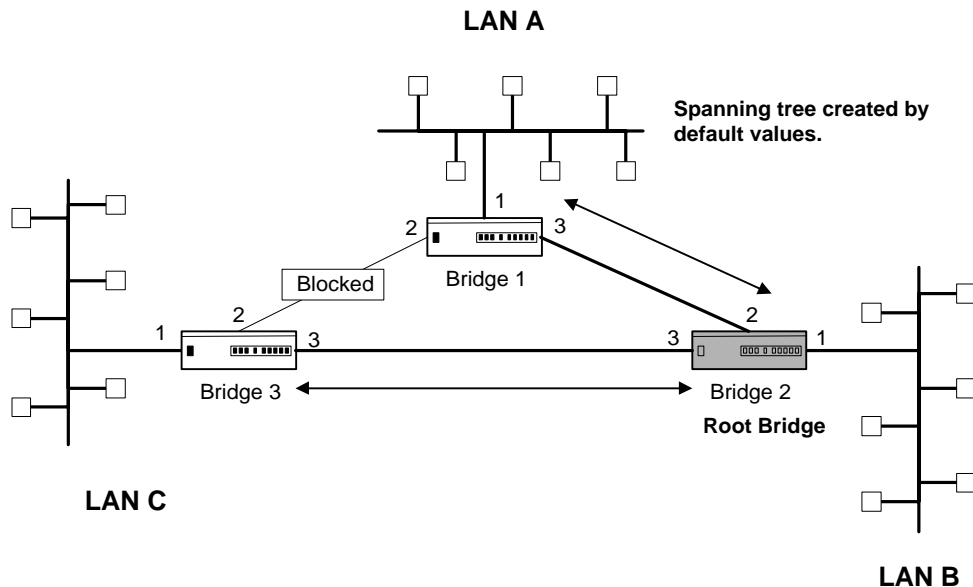
LKG-10575-97V

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 is shown in the Figure 2–2 as the heavy lines connecting Bridge 1 to Bridge 2, and then Bridge 2 to Bridge 3. The root bridge is Bridge 2.

2.1 Transparent Bridge (STB)

This spanning tree results in poor network performance because the workstations on LAN C can only get to the file server on LAN A 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

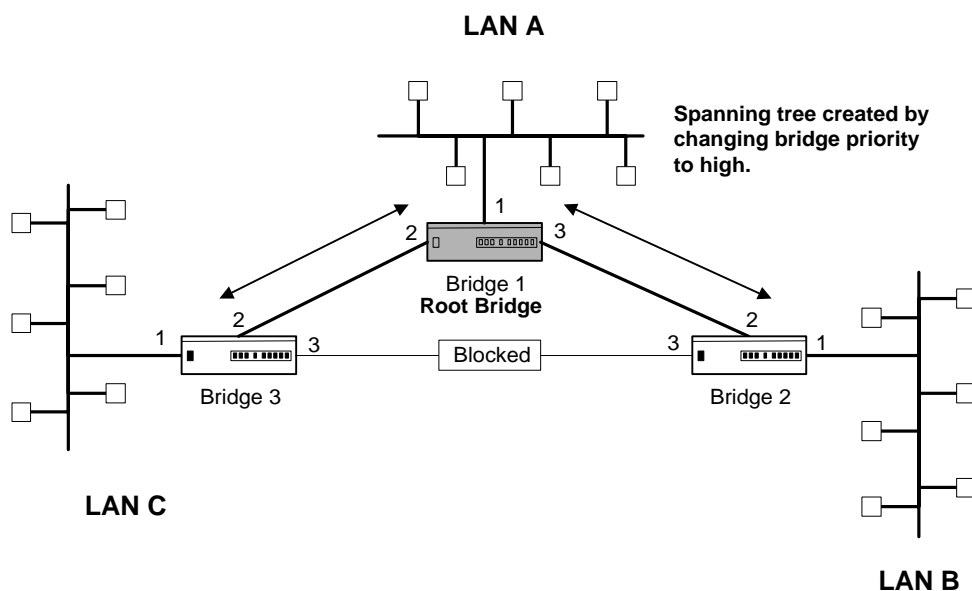


LKG-10576-97V

Normally this network seldom uses the port between Bridge 2 and Bridge 3. 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.

2.1 Transparent Bridge (STB)

Figure 2-3 User-Adjusted Spanning Tree



LKG-10577-97V

2.1.5 Spanning Tree Bridges and Ethernet Packet Format Translation

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. While IEEE 802.3 packets can be sent on FDDI (and vice versa) by only replacing of the MAC header, Ethernet packets require translation of the *data link* header to communicate with and across FDDI. Thus, the bridge must translate to and from Ethernet format to provide transparency across mixed LAN types.

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 Organizationally Unique Identifier (OUI) of 00-00-00, with the last two bytes being the Ethernet type value. The reverse mapping happens when going from FDDI to Ethernet.

2.1 Transparent Bridge (STB)

One special case is AppleTalk ARP Phase 1 which has an Ethernet type of 80F3 (hexadecimal) and Protocol Identifier 00-00-F8-80-F3 on FDDI. (AppleTalk ARP Phase 2 uses Protocol Identifier 00-00-00-80-F3 on IEEE 802.3 which leads to ambiguities.)

The translation occurs when a frame is sent on a LAN. The original frame format is preserved across serial lines.

2.1.6 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 refreshed before the aging time, they are deleted.

Bridge

A bridge is a protocol-independent device that connects local area networks. 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 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 since it controls the hello time for all bridges in the spanning tree. Use the **set protocol bridge** command to set the bridge hello time.

2.1 Transparent Bridge (STB)

Bridge Forwarding 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 since 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 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 change occurs. All the bridges in the spanning tree use this age to time out the received configuration information in their database. This allows a uniform time-out 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 priority for each bridge to become the root bridge of the network. In setting the bridge priority, the spanning tree algorithm chooses the bridge with the highest priority to be the root bridge of the spanning tree. The bridge with the lowest numerical value has the highest priority.

2.1 Transparent Bridge (STB)

Designated Bridge

The designated bridge is the bridge that claims to be the closest to the root bridge (or the lowest bridge ID value if two or more bridges have equal cost root paths) 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 ports connected to the LAN.

The filtering database is initialized with entries from the permanent database. These entries are permanent and survive power cycles or system resets. You can add or delete these entries through the bridge configuration commands. Entries in the permanent database are stored as records in nonvolatile memory, and the number of entries is limited by the size of nonvolatile memory.

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

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

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

- **Address** – 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.

2.1 Transparent Bridge (STB)

- 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 the database that are free to be filled by address entries.
- **Address Age (dynamic entries only)** – Resolution of time period when address entries are ticked down before being discarded. The user can set this value.

Make changes to the permanent database through the bridge 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 cost 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 port costs on the path to the root bridge 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-settable. Each port's 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.

2.2 Source Routing Bridge (SRB)

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 and the ports interconnecting them such that there is one and only one path between any two end stations.

Transparent Bridging

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

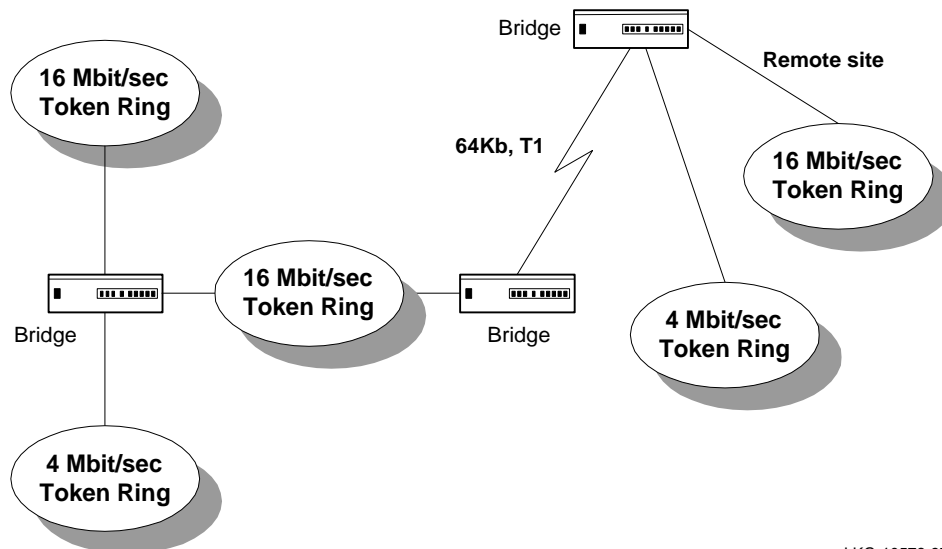
2.2 Source Routing Bridge (SRB)

Source routing is a method of forwarding frames through a bridged network in which the source station identifies the route that the frame follows. 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 provides local bridging over 4Mb/s and 16Mb/s Token Rings, as shown in Figure 2-4. It can also connect remote LANs through a WAN link.

2.2 Source Routing Bridge (SRB)

Figure 2-4 Example of Source Routing Bridge Connectivity



LKG-10578-97V

Among its features, the source-routing bridge provides:

- **IBM Compatibility** – The bridge is compatible with the IBM source-routing bridge. 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.
- **Bridge Tunneling** – By encapsulating source-routing packets in IP, the bridging router dynamically routes these packets through internetworks to the desired destination end station without network size restrictions.

Source routing end stations see this path (the tunnel) 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 nonsource-routing media (for example, Ethernet networks).

2.2 Source Routing Bridge (SRB)

2.2.1 Source Routing Bridge Operation

The source station defines the entire route in the transmitted frame in a source-routing configuration. 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 are ignored by other end stations.
5. The originating station receives the learned route path. It can then transmit information across this established path.

2.2.2 Source Routing Frames

Bridges interconnect LANs by relaying MAC frames between the MAC layers of the bridged LANs. MAC frames provide the necessary "Where?" information in the form of source and destination addresses.

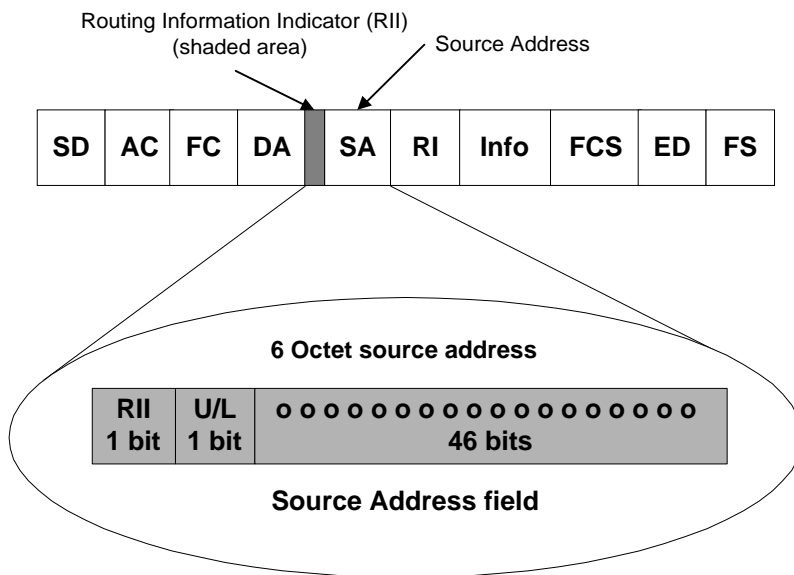
In source-routing, the data frame forwarding decision is based on routing information within the frame. The source station that originates the frame designates the route that the frame travels by embedding a description of the route in the Routing Information Field (RIF) of the transmitted frame. A closer look at the various types of source-routing bridge frames helps to further explain how bridges obtain and transmit routing information.

2.2 Source Routing Bridge (SRB)

Because source-routing MAC frames contain routing information necessary for data communication over multi-ring environments, their formats differ slightly from the typical Token Ring MAC frames. The presence of a “1” in the Routing Information Indicator (RII) within the source address field indicates that an RIF containing routing information follows the source address.

Figure 2–5 illustrates the format of the source address field of a source-routing frame.

Figure 2–5 802.5 Source Address Format



RII = Routing information indicator **RII = 0 means RI field is not present in frame**
U/L = Universal or local bit **RII = 1 means RI field is present in frame**

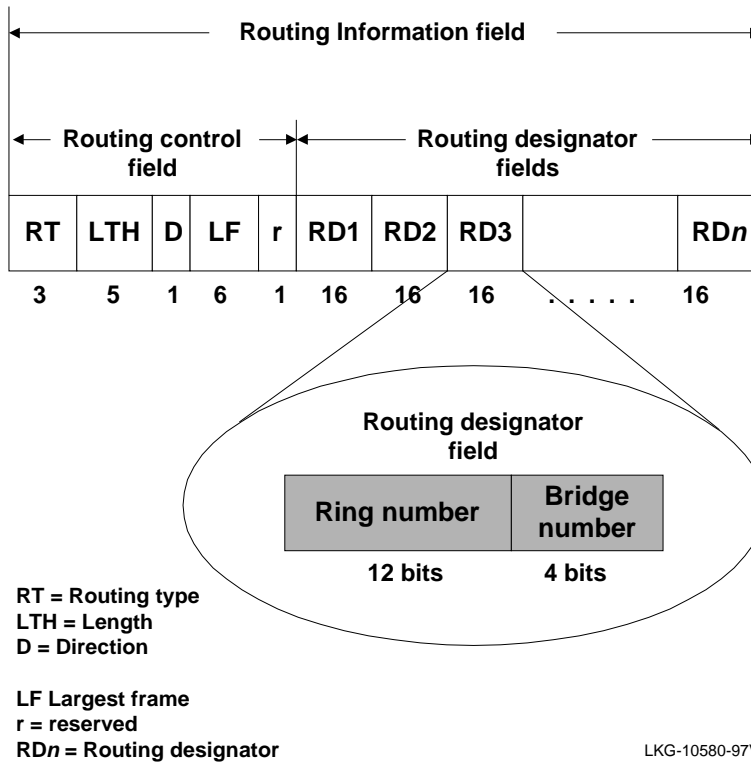
LKG-10579-97V

When the RII in the source address field is set to 1, an 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 illustrates the format of the Routing Information Field.

2.2 Source Routing Bridge (SRB)

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)

2.2 Source Routing Bridge (SRB)

All-paths explorer frames have RT bits set to 100. These frames are generated and routed along every nonrepeating 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. Each bridge on the path adds routing designators to the frame.

A Spanning-Tree Explorer frame has RT bits set to 110. Only spanning tree bridges relay the frame from one network to another. This means that the frame appears only once on every ring in the network and only once at the destination end station. A station initiating the route discovery process can use 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 have the first RT bit set to 0. In this case, the Route Designator (RD) fields contains specific routing information that guides the frame through the network to the destination address.

A spanning tree routed frame has the RT bits set to 111. 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. The bridges do not add any routing designators to the frame. If the frame was forwarded from network A and is being transmitted on network B, when the frame is stripped it is not forwarded back to network A. This type of frame is the default type that a station uses to send frames to stations using a group address.

- **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 RD n). If the direction bit is set to 1, the frame travels the networks in the reverse order.

2.2 Source Routing Bridge (SRB)

- **Largest Frame Bits (LF)** – Indicates the largest frame size of the INFO field that may 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 (SRF), the bridge ignores the LF bits and cannot alter them. A station originating an explorer frame sets the LF bits to the maximum frame size it can handle. Forwarding bridges set the LF bits to the largest value that does not exceed the minimum of the following:
 - The indicated value of the received LF bits
 - The largest 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

The destination station may further reduce the LF value to indicate its maximum frame capacity.

LF bit encodings are 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 so the bridge can 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).

2.2.3 The Spanning Tree Explore Option

The spanning tree explore option 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. This option allows you to simulate a spanning tree network and balance traffic loads.

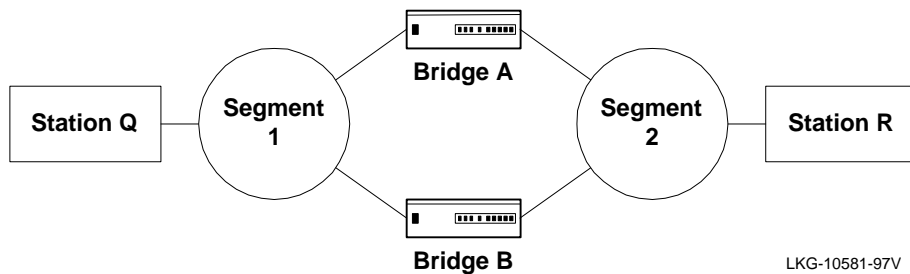
2.2 Source Routing Bridge (SRB)

2.2.3.1 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, lowering network overhead.

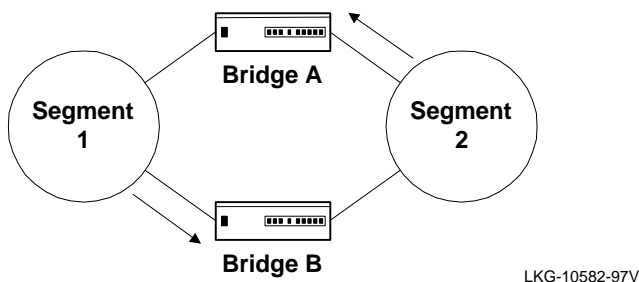
Figure 2–7 Example of Parallel Bridges



2.2.3.2 Balancing Traffic Loads

You can also use the spanning tree explore option for load balancing. For example, in Figure 2–8, 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



2.2 Source Routing Bridge (SRB)

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. 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.

2.2.4 Protocol Filtering

A router 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 and protocol type of incoming frames.

Table 2–1 shows how the “Bridge or Route?” question is answered, based on the destination address and protocol type contents.

Table 2–1 Route/Bridge Decision Table

If the Destination MAC Address in the Received Frame Contains:	The Bridge Uses This Logic:
The Bridge's MAC Address	The bridge passes the frame to the configured layer 3 protocol that either routes the frame or passes it to the application residing in the router.
Multicast or Broadcast Address	If the protocol type is configured for routing, then the frame is passed to the layer 3 protocol that either routes the frame or passes it to the application residing in the router. If the protocol type is not configured for routing, then the frame is bridged.
Unicast (not the Bridge's MAC Address)	The frame is bridged if its protocol type is not configured for routing, otherwise it is dropped.

2.2 Source Routing Bridge (SRB)

2.2.5 Source Routing Bridge Terminology and Concepts

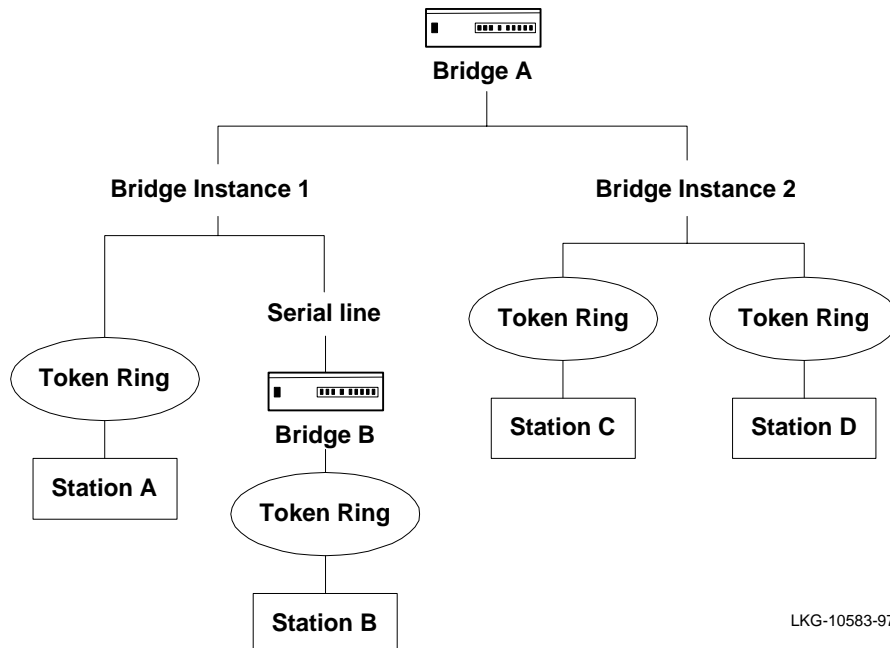
This section reviews the terms and concepts commonly used in source-routing bridging.

Bridge Instance

The bridge instance identifies a logical bridge defined within the system. For example, in a bridge with two configured bridge instances, the bridge instance identifiers are 1 and 2.

Bridge instances within a single system 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 only pass frames 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



LKG-10583-97V

Bridge Number

The bridge number identifies the source-routing bridge specific to a segment or ring. This is a 4-bit integer value which must be unique on a given segment or ring.

2.3 Source Routing Transparent Bridge (SRT)

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 device (network interface card) within the bridge and links it to other configuration information. When you configure the router software, the router/bridge sequentially numbers the interfaces. To use the source-routing bridge, you must use the numbers assigned in that process to identify the interface connecting each network segment.

Route

A path through a series of LAN segments and bridges (SRB bridges, for example).

Route Discovery

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.

Source Routing

Source routing is a bridging mechanism that routes frames through a multi-LAN network by specifying in the frame the sequence of bridges and ports that the frame travels to reach its destination.

2.3 Source Routing Transparent Bridge (SRT)

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.

2.3 Source Routing Transparent Bridge (SRT)

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.

Source Routing Transparent (SRT) bridging is defined in IEEE 802.1D (ISO 10038) Appendix C. 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 section describes SRT Bridging in more detail and includes the following:

- General Description
- Source Routing Transparent Bridge Operation and Architecture
- Source Routing Transparent Bridge Terminology and Concepts

2.3.1 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 sub-layer of the data link layer and are completely invisible to the end stations.

The SRT bridge distinguishes between frames that must be bridged transparently and those that must be bridged by source-routing by checking the value in the RII field of the frame (see section 2.2 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

2.3 Source Routing Transparent Bridge (SRT)

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 a source-routing bridging domain.

The Spanning Tree protocol and algorithm form a single tree involving all the networks connected by SRT bridges. The SRT-bridged network offers a larger domain of transparent bridging with a sub-domain of source-routing. Thus, transparent frames are capable of reaching to the farthest side of the SRT and TB bridged LAN while source-routed frames are limited to only the SRT and SRB bridged LAN. In the SRT bridging model, source-routing and transparent bridging parts use the same spanning tree. In the SRT bridged domain, end stations decide whether to use Source Routing or Transparent Bridging when sending a packet.

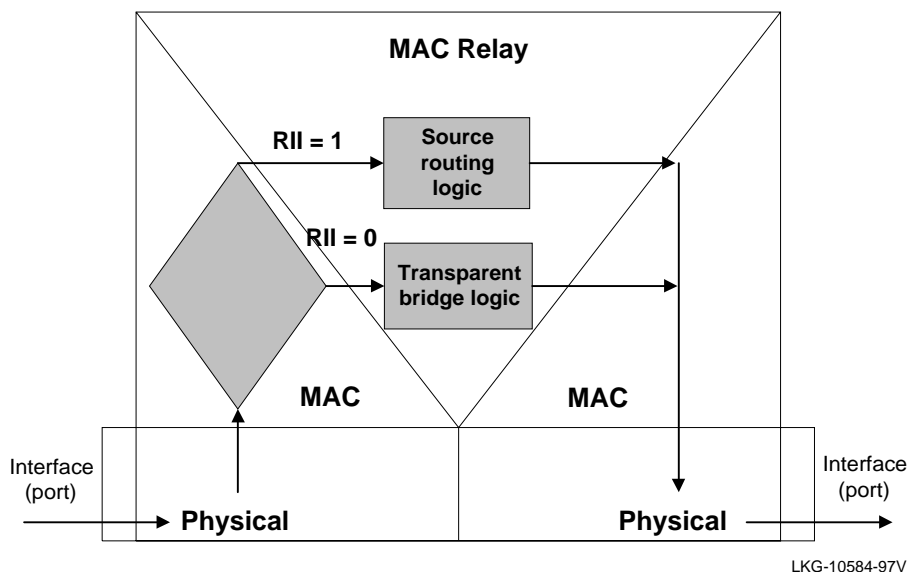
2.3.2 Source Routing Transparent Bridge Operation and Architecture

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.

2.3 Source Routing Transparent Bridge (SRT)

Figure 2–10 SRT Bridge Operation



SRT differentiates between source-routed and nonsource-routed traffic on a frame-by-frame basis.

2.3.3 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 and 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

2.4 Adaptive Source Routing Transparent Bridge (SR–TB) Conversion

originates the frame (the source station) designates the route that the frame travels by embedding a description of the route in the Routing Information Field (RIF) of the transmitted frame.

Routing Information Indicator (RII)

Since 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 nonsource-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 multiple Token Ring LAN network by specifying in the frame the route it travels.

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 the operation of the spanning tree algorithm.

2.4 Adaptive Source Routing Transparent Bridge (SR–TB) Conversion

While source-routing is still available in the SRT model, it is only available 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.

2.4 Adaptive Source Routing Transparent Bridge (SR–TB) Conversion

The following section describes the ASRT bridge in detail and includes the following sections:

- General Description
- Source Routing – Transparent Bridge Operation
- Source Routing – Transparent Bridge Terminology and Concepts

2.4.1 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 functionality 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.

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

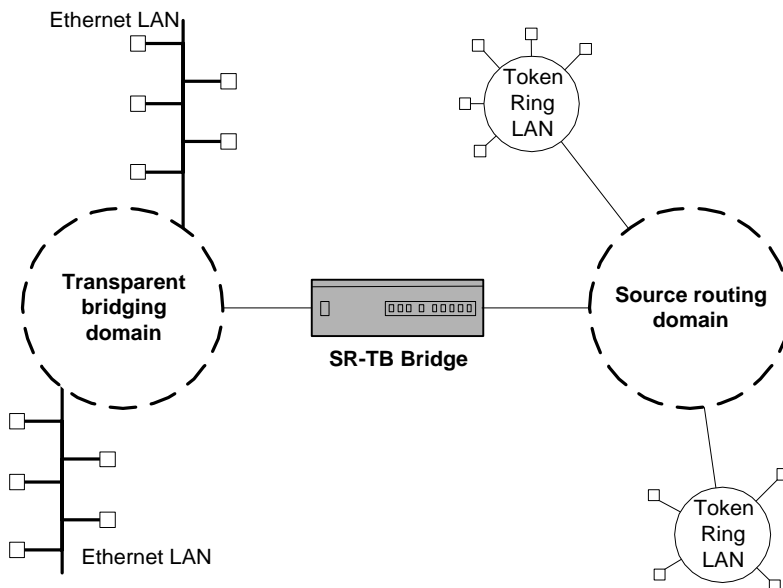
2.4.2 Source Routing – Transparent Bridge Operation

During SR-TB bridging, 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

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 shows an example of these domains.

Figure 2-11 SR-TB Bridge Connecting Two Domains



LKG-10585-97V

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

2.4.2.1 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 2-2.

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

Table 2-2 SR-TB Bridge Decision Table

Type of Frame Received	Action Taken by SR-TB Bridge
Non-routed frame received by the Source Routing station.	Does not copy or forward frame 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 destination address is in the transparent bridging table, bridge forwards the frame without routing information on the transparent bridging network. Otherwise, frame is not forwarded.
Single-route broadcast frame received by the Source Routing station. Bridge <i>is not</i> designated as single-route broadcast bridge.	Does not copy or forward the frame.
Single-route broadcast frame received by the Source Routing station. Bridge <i>is</i> 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 saved Routing Information Field and the LAN number for transparent bridging side. Changes broadcast indicator to nonbroadcast, complements D-bit, and stores this routing information for the source address of the frame.

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

Table 2-2 SR-TB Bridge Decision Table (Continued)

Type of Frame Received	Action Taken by SR-TB Bridge
Non-broadcast frame received by the source-routing station.	<p>If frame carries specific route, bridge examines the routing information.</p> <p>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.</p> <p>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.</p>
Frame received from the Transparent bridging side.	<p>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.</p> <p>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.</p> <p>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.</p>

2.4.2.2 SR-TB Bridging: Four Examples

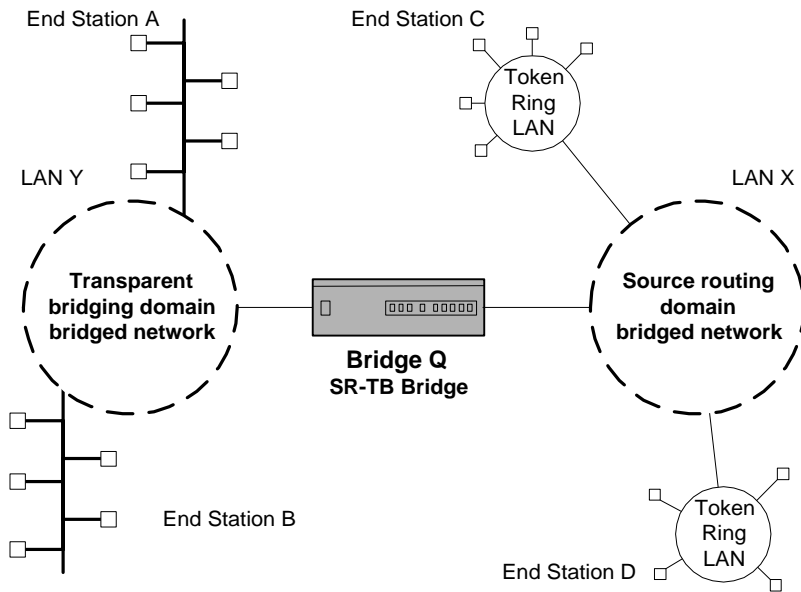
The SR-TB bridge interconnects source-routing domains with transparent bridging domains by transparently joining the domains. During operation, stations in both domains are not aware of the existence of each other or of the SR-TB bridge. From the end station's point of view, any station on the combined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

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

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 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



LKG-10586-97V

2.4.2.3 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.

2.4 Adaptive Source Routing Transparent Bridge (SR–TB) Conversion

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.

2.4.2.4 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 in the previous example. Since End Station C's address is definitely not in the transparent bridge address table, the SR–TB bridge forwards 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 uses this specific route, and because the route is marked as temporary, the frame is sent as a nonbroadcast 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 causes 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 (SMT) 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.

2.4 Adaptive Source Routing Transparent Bridge (SR–TB) Conversion

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

If the frame is sent as a nonbroadcast 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 does not forward the frame.

If the frame is sent as a single-route broadcast, the bridge discards 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 nonbroadcast 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.

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

If the frame is sent nonbroadcast, 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 nonbroadcast 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.

2.4.3 Source Routing – Transparent Bridge 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 nonrepeating route in the bridged LAN.

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

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 Spanning-Tree Explorer (STE) frames. ARE frames are forwarded by all ports while STE frames are forwarded only by ports assigned to forward them by the Spanning Tree protocol.

Ring Number

The unique number identifying a ring in a bridged network.

Route

A path through a series of LANs and bridges (for example, SRB 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.

2.5 ASRT Bridge Overview

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 travels.

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 stations. Transparent bridging interconnects local area network segments by bridges designated to forward data frames in a spanning tree algorithm.

2.5 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 functionality and source-routing functionality so that they function separately or can be combined as a single ASRT bridge. This extended functionality allows communication between a strict source-routing end station and a transparent end station through 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)

Modifications have been built into the ASRT bridge that provide users with extended functionality exceeding 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 functionality in some additional, critical ways described below.

2.5.1 Transparent and Source-Routing Compatibility – Issues and

2.5 ASRT Bridge Overview

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 a proposed 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. This means it functions as either a transparent bridge or a source-routing bridge by inserting or removing an RIF as necessary. With this functionality, packets can move between Ethernet and SRT token-ring LANs and still be compatible with an installed base of source-routing token-ring LANs.

2.5.1.1 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 since the Token Ring packet size allowance is much larger.

2.5.1.2 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 effects the interoperability 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, 802.5, and FDDI LANs. Two different standards 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) and FDDI LANs transmit source and destination addresses with the group bit first and LLC data fields transmitted Most Significant Bit (MSB) first.

2.5 ASRT Bridge Overview

Note: 802.3 and 802.4 bridges and LANs are now referred to as LSB bridges and LANs. 802.5 and FDDI bridges and LANs are referred to as MSB bridges and LANs.

The difference in the bit transmission standard means that a bridge from LSB to MSB LANs must 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 (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, since 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.

2.5.2 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 forms in an environment where different types of bridges exist. For example, a spanning tree of all bridges practicing IEEE802.1D protocol (such as STB and SRT) existing with another tree of IBM 8209 bridges. The loops forming from this configuration require you to correct the situation.

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

Bridging Features

This chapter describes bridging features that are available with the Adaptive Source Routing Transparent (ASRT) Bridge.

3.1 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 bridging distance limit encountered in source routing configurations. It also lets you connect source routing end stations across nonsource routing media, such as Ethernet networks.

The bridging tunnel overcomes several limitations of regular source routing that includes the following:

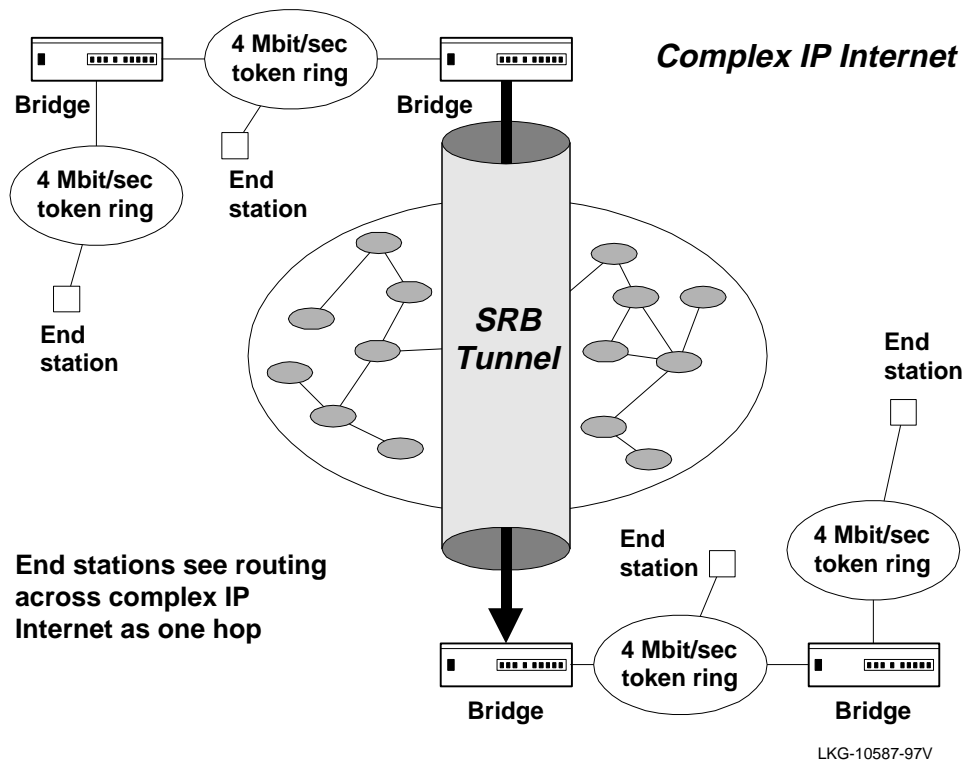
- Distance limitations of seven hops
- Large amounts of overhead that source routing causes in wide area networks (WANs)
- Source Routing's sensitivity to WAN faults and failures (if a path fails, all systems must restart their transmissions)

With the bridge tunnel feature enabled, the software encapsulates packets in IP packets. To the router, the packet looks like a 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.

Figure 3–1 shows an example of an IP internetwork using the tunnel feature in its configuration.

3.1 Bridging Tunnel

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 IP headers are automatically removed and the inner packet proceeds as a standard source routing packet.

3.1.1 Encapsulation and IP Routing Protocols

A major benefit of the encapsulation feature is the addition of a dynamic routing protocol to the routing process. Dynamic routing protocols such as OSPF and Integrated IS-IS offer the following benefits when used with encapsulation:

- **Least-Cost Routing** – Dynamic routing protocols access the shortest path (tunnel), allowing network administrators to distribute traffic over the least expensive route.
- **Dynamic Routing** – Dynamic routing protocols find the least-cost path, detect failures, and reroute traffic with low overhead.

3.2 TCP/IP Host Services (Bridge-Only Management)

With dynamic routing, tunnels automatically manage paths inside the internetwork. If a line or bridge fails along the path, the tunnel bridge automatically reroutes traffic along a new path. If a path is restored, the tunnel automatically updates to the best path. This rerouting is completely transparent to the end stations.

3.2 TCP/IP Host Services (Bridge-Only Management)

The Bridging Router also supports TCP/IP Host Services that let you configure and monitor a bridge when routing functions are disabled. This option gives you the following capabilities:

- Management through SNMP
- Telnet server functionality
- Downloading and uploading of configuration through the TFTP protocol
- TFTP neighbor boot functionality
- IP diagnostic tools of ping and traceroute
- 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 through the **protocol** command in the `Config>` and `+ (GWCON)` consoles.

Bridge-only management functionality is activated by assigning an IP address to the bridge and enabling TCP/IP Host Services. 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. Default gateway assignments may 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 functionality but do not require IP routing to be present.

3.3 Bridge-MIB Support

3.3 Bridge-MIB Support

Bridge Management through SNMP supports the Bridge-MIB as specified by RFC 1286. The entire Bridge-MIB is implemented except for the following:

- The “forwarding database table” for transparent bridges (dot1dTpFdbTable).
- The “static (Destination-Address Filtering) database table” (dot1dStaticTable).
- The “newRoot” and “topologyChange” traps.

The entire Bridge-MIB is read-only.

3.4 Pseudo Serial Ethernet

Pseudo Serial Ethernet is an optional mode of operation that provides for the encapsulation of any routed protocol on a bridging router proprietary serial line, to be forwarded within an Ethernet encapsulated frame over the same serial line. This allows the protocol to communicate with a pure bridge on the opposite end of the serial line.

When enabled, this mode makes the serial lines appear as an Ethernet interface to the configured routing protocols. The handler uses Ethernet (or IEEE 802.3, as appropriate) encapsulations, thus limiting the protocols to the maximum Ethernet frame size. These Ethernet frames are then sent and received as *bridged* Ethernet frames on the serial line. Any frames arriving on the routed protocol code points from the serial lines are ignored, and bridged Ethernet frames are passed to the bridging or routing forwarders as appropriate.

This encapsulation is normally not necessary with the bridging routers at both ends of the serial line, since both can be configured to route the same set of protocols over the same serial line.

3.5 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.

3.5 Multiple Spanning Tree Protocol Options

3.5.1 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, FDDI, and so forth). 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 SRB bridges to determine a loop-free topology. The algorithm sends Hello BPDUs in a transparent frame to well-known functional addresses. Since TCN BPDUs 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 are sent as transparent frames to the same group address. 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 and Hello BPDUs sent by SRBs are filtered. There is no way to prevent the 8209 from becoming the root bridge. If the 8209 bridge is selected as the root, then traffic between two Transparent Bridge domains may have to pass through Token Ring / SRB domains.

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

3.5.2 Spanning Tree Protocol and 8209 Bridging (STP/8209)

The STP/8209 bridging feature is available to allow you to further extend the spanning tree protocol. Previously, SRB bridges 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.

3.5 Multiple Spanning Tree Protocol Options

- **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 SRB spanning tree protocol is used for SRB.
- **ST-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. Since both BPDUs carry the same information, there is 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.

3.5.3 Logical Link Class 2 Support

In LANs, the data link layer is comprised of two sublayers: the medium access control (MAC) and the link layer control (LLC). LLC provides two types of services:

- **LLC1 (Type 1)** – An unacknowledged connectionless service
- **LLC2 (Type 2)** – A connection-oriented service

LLC2 provides the following capabilities:

- Initiating new data link connections
- Managing data link connections
- Exchanging data in sequential order (in a guaranteed fashion)
- Executing a level of flow control on the established connections

3.6 Threading

- Terminating link connections upon request from the service user or unrecoverable link errors.

The LLC sublayer adheres to the IEEE 802.5 standard.

3.6 Threading

Threading is a process where the network protocol (IPX, DNA, IP, and AppleTalk) of the Token Ring end station discovers a route over segments of a Source Routing Bridge Network.

Threading is similar to the Source Routing Bridge operation. It is how threading is implemented by the end station that is different. The following sections describe threading for IP, DECnet, IPX, and AppleTalk.

3.6.1 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 Explore) or an STE (Spanning Tree 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.

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 receives any other ARP REQUEST packets from the same source, that packet is dropped.

3.6 Threading

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.

3.6.2 DECnet (DNA) Threading

DECnet end stations use ARE to discover a route. Both the DNA end stations and the bridges participate in the route discovery process and forwarding. The following steps describe the DECnet threading process:

1. If there is no entry in the RIF table for the MAC address, an entry is created with the state NO_ROUTE. When this occurs the end station sends the data packet out with an STE attached. The STE is used for discovery without attempting to flood the network with ARE.
2. The end station then transmits an ARE in a loop-back frame for the destination MAC address.
3. All bridges on the local segment capture the STE and loop-back frame and send it over their connected networks.

As the packets continue their search for the destination end station, each bridge that forwards it adds its own bridge number and segment number to the RIF in the STE and the ARE. 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 STE and loop-back frame finally reaches the destination, it contains the exact sequence of bridge and segment numbers from the source to the destination.

4. When the destination end station receives the loop-back frame, it places the MAC address and the RIF of the source station into its own RIF table. If a RIF already exists for that entry, it either updates the RIF if that previous entry is an ST_ROUTE (refer to step 7) or it ignores the RIF. In any case, the entry state is changed to HAVE_ROUTE.
5. The destination end station then sends the loop-back reply frame including the specific RIF back to the source end station.
6. The source end station receives the learned specific route path. The RIF is entered into the RIF table and the entry is changed to HAVE_ROUTE.

3.6 Threading

7. Packets destined for a functional address are sent with an STE. DNA end stations can create a RIF entry using this STE frame. When this happens, the state of the entry is changed to ST_ROUTE. This type of route is less desirable and is replaced as described in step 4.

The DNA end stations contain an independent RIF timer. When this timer expires for a specific RIF entry, an ARE in a loop-back packet is sent out to that specific destination. When the loop-back frame returns, the RIF entry is updated. If the destination end station is on the same ring and the loop-back frame contains no RIF, the loop-back packet is returned with no RIF entry.

3.6.3 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 is not reentered until another packet arrives containing a RIF for that entry.

3.6.4 AppleTalk 1 and 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.

3.6 Threading

3. When the destination end station receives the frame, it places the MAC address and its RIF into its own ARP and RIF tables. The state of the entry is designated as HAVE_ROUTE. If the destination end station receives any other ARP REQUEST packet 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. 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.

Configuring and Monitoring ASRT

This chapter describes how to configure and monitor the Adaptive Source Routing Transparent (ASRT) Bridge protocol and how to use the ASRT configuration and console (monitoring) commands.

4.1 Accessing the ASRT Environments

Use the **protocol asrt** command at the `Config>` prompt or `GWCON` prompt (+) to access the ASRT configuration or console environments respectively. For more information on how to access the ASRT configuration and console environments, see the Chapter 1 in the *System Software Guide*.

4.1.1 ASRT Configuration Commands

ASRT configuration commands allow you to specify network parameters for the ASRT bridge and its network interfaces. These commands also allow you to enable and configure the bridge IP Tunnel, NetBIOS Filtering, and LAN Network Manager features.

Note: After you access the bridging configuration process, you may begin entering configuration commands. Whenever you make a change to a user-configurable interface parameter, you must restart the router for this change to take effect.

Enter the ASRT configuration commands at the `ASRT config>` prompt.

Enter configuration commands for IP tunnels at the `tnl config>` prompt. The tunnel prompt, `TNL config>`, is a subset of the major ASRT commands and is accessed by entering the ASRT **tunnel** command.

Configuration commands for the LNM (LAN Network Manager) process are entered at the `LNM config>` prompt. This prompt is a subset of the major ASRT commands and is accessed by entering the ASRT **LNM** command.

4.1 Accessing the ASRT Environments

Configuration commands for NetBIOS Filtering are entered at the `NetBIOS Filter config>` prompt. This prompt is a subset of the major ASRT commands and is accessed by entering the ASRT **netbios-filter** command. NetBIOS and NetBIOS filtering are described in detail in Chapter 9, Chapter 10 and Chapter 11.

4.1.2 ASRT Console Commands

ASRT console commands allow you to view and modify parameters from the active console. Information you modify with the console commands is not permanent. The configuration is reset to that stored in nonvolatile memory when you restart the bridging router.

You can use the console commands to temporarily modify the configuration without losing configuration information in the bridge memory. The `ASRT>` prompt displays for all ASRT console commands.

Note: For commands requiring you to enter MAC Addresses, the addresses can be entered in the following formats:

IEEE 802 canonical bit order - 00-00-00-12-34-56
IEEE 802 canonical bit order - 000000123456 (shorthand format)
IBM token ring native bit order - 00:00:00:12:34:56 (non-canonical)

Table 4–1 lists the ASRT configuration and console commands.

Table 4–1 ASRT Configuration and Console Command Summary

Command	Task	Function
? (Help)	Configure/ Monitor	Lists all of the ASRT configuration or console commands, or lists the options associated with specific commands.
Add	Configure/ Monitor	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.
Cache	Monitor	Displays cache entries for a specified port.
Change	Configure	Allows the user to change bridge and segment numbers.

4.1 Accessing the ASRT Environments

Table 4–1 ASRT Configuration and Console Command Summary (Continued)

Command	Task	Function
Delete	Configure/ Monitor	Deletes station address entries, specific address mapping, LAN/WAN ports, protocol filters, and a tunnel between end stations across an IP internetwork. As a console command, deletes MAC address entries from the bridging router database.
Disable	Configure	Disables bridging functionality, 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 frame to transparent frame, transparent (spanning tree) bridging functionality on a given port, and a tunnel between bridges.
Enable	Configure	Enables bridging functionality, 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 frame to transparent frame, transparent (spanning tree) bridging functionality on a given port, and a tunnel between bridges.
Flip	Monitor	Flips MAC address from canonical to 802.5 (non-canonical or IBM) bit order.
List	Configure/ Monitor	Displays information about the complete bridge configuration or about selected configuration parameters and options.
LNM	Configure	Allows access to the LNM (LAN Network Manager) configuration prompt so that LNM configuration commands can be entered.
Netbios-filtering	Configure/ Monitor	Allows access to the NetBIOS filtering prompt so that NetBIOS filtering configuration or console commands can be entered.
Set	Configure	Sets 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, and filtering database size.

4.1 Accessing the ASRT Environments

Table 4–1 ASRT Configuration and Console Command Summary (Continued)

Command	Task	Function
Tunnel	Configure	Allows access to the tunnel configuration prompt so that tunnel configuration commands can be entered.
Exit	Configure/ Monitor	Exits the ASRT configuration or console process and returns to the previous prompt level.

? (Help)

C M

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

C

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

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 one hop between the end stations, no matter how complex the path through the IP internet.

Syntax: add

address . . .
mapping . . .
port . . .
prot-filter . . .
tunnel . . .

4.1 Accessing the ASRT Environments

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.

Example:

```
add address
Address (in 12-digit hex) []?
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):
  continue to another input port? (Yes or No):
  Input Port Number [2]?
  Bridge to all ports?(Yes or No): n
  Bridge to port 1 - Yes or No: y
  Bridge to port 2 - Yes or No: n
  Bridge to port 3 - Yes or No: y
  continue to another input port? (Yes or No): y
  Input Port Number [3]? 3
  Bridge to all ports?(Yes or No): y
  continue to another input port? (Yes or No): y
Source Address Filtering Applies? (Yes or No): y
```

Note: For any Yes or No question in the prompts, No is the default value. Press the Return key to enter the default value.

<i>Exclude destination address...</i>	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.
---------------------------------------	--

4.1 Accessing the ASRT Environments

Use same output mapping...

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

Input Port 1, Port 2

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

Bridge to all ports?

Answering Yes to this prompt creates an outgoing port map that includes all ports. 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) then 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 to include that port 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.

4.1 Accessing the ASRT Environments

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 is 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.

Example 1. Destination Address Filtering Enabled for Entry

This example shows how to answer the command prompts to select Destination Address Filtering for an entry:

Example:

```
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 example below shows that no port map exists for that entry (in **bold**) and that Destination Address Filtering (DAF) was turned on.

Example:

```
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
```

4.1 Accessing the ASRT Environments

Example 2. 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 has more than one input port.

Example:

```
ASRT config> add address 000000123456
Filter exclusively, no matter what input port?(Yes or No): n
Single output port map for all input Ports?(Yes or No): n
Input Port Number [1]? 1
All Ports?(Yes or No): n
Port[1] - Yes or No: y
Port[2] - Yes or No: y
Port[3] - Yes or No: n
Port[4] - Yes or No: n
continue? (Yes or No): y
Input Port Number [2]? 2
All Ports?(Yes or No): n
Port[1] - Yes or No: n
Port[2] - Yes or No: n
Port[3] - Yes or No: y
Port[4] - Yes or No: y
continue? (Yes or No): n
Source Address Filtering Applies? (Yes or No): n
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) is also enabled.

Example:

```
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, 4
```

4.1 Accessing the ASRT Environments

Example 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.

Example:

```
ASRT config> add address 000000556677
Filter exclusively, no matter what input port?(Yes or No): n
Single output port map for all input Ports?(Yes or No): y
All Ports?(Yes or No): n
Port[1] - Yes or No: y
Port[2] - Yes or No: y
Port[3] - Yes or No: n
Port[4] - 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 an entry (in bold) that has a single port map for all incoming ports. Source Address Filtering (SAF) is also enabled.

Example:

```
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, 4
```

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/FDDI or vice versa.

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

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

4.1 Accessing the ASRT Environments

<i>type-field</i>	Protocol type field: <ul style="list-style-type: none">• 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.
<i>ga-address</i>	6-byte (12-digit hexadecimal) group/multicast address.
<i>fa-address</i>	Functional address in non canonical format. Functional addresses are locally administered group addresses. These are most commonly used in token ring networks.

Note: The most commonly used values for DECnet group address-to-functional address mapping are listed as follows:

<u>Ethertype</u>	<u>Group Address</u>	<u>Functional Address</u>
6002	ab-00-00-02-00-00	C0:00:20:00:00:00
6003	ab-00-00-03-00-00	C0:00:10:00:00:00
6003	ab-00-00-04-00-00	C0:00:08:00:00:00

<u>SNAP</u>	<u>Group Address</u>	<u>Functional Address</u>
00-00-00-6002	ab-00-00-02-00-00	C0:00:20:00:00:00
00-00-00-6003	ab-00-00-03-00-00	C0:00:10:00:00:00
00-00-00-6003	ab-00-00-04-00-00	C0:00:08:00:00:00

Example:

```
add mapping dsap
Protocol Type in hex (1 - FE) [1]?
Group-Address (in 12-digit hex) [ ]?
Functional address (in non-canonical format) [ ]?
```

4.1 Accessing the ASRT Environments

Example:

```
add mapping ether
Protocol Type in hex (5DD - FFFF) [0800]?
Group-Address (in 12-digit hex) []?
Functional address (in non-canonical format) [ ]?
```

Example:

```
add mapping snap
Address (in 10-digit hex) [0000000800]?
Group-Address (in 12-digit hex) []?
Functional address (in non-canonical format) [ ]?
```

port interface# port#

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

Example:

```
add port 0 4
Interface Number [0]?
Port Number [5]?
```

prot-filter snap ether dsap

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

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

<i>SNAP Packets</i>	Subnetwork Access Protocol with protocol type entered in 10-digit hexadecimal format.
<i>Ether Packets</i>	Ethernet Type with the protocol type entered in a range of 05DD to FFFF (hexadecimal).
<i>DSAP Packets</i>	Destination Service Access Point protocol with the protocol type entered in a range of 01 to FE (hexadecimal).

4.1 Accessing the ASRT Environments

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

DSAP Types

<u>Protocol</u>	<i>SAP (hexadecimal value)</i>
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

<u>Protocol</u>	<i>SNAP OUI/IP (10-digit)</i>
AppleTalk Phase 2	08-00-07-80-9B
AppleTalk ARP Phase 2	00-00-00-80-F3
Proprietary AppleTalk Phase 1 for FDDI	00-00-93-00-02
Proprietary AppleTalk ARP Phase 1 for FDDI	00-00-93-00-03

Ethernet Types

<u>Protocol</u>	<i>Ethernet type (hexadecimal value)</i>
IP	0800
ARP	0806
CHAOS	0804
DECnet MOP Dump/Load	6000
DECnet MOP Remote Console	6002
DECnet	6003

4.1 Accessing the ASRT Environments

<u>Protocol</u>	<i>Ethernet type (hexadecimal value)</i>
DEC LAT	6004
DEC LAVC	6007
XNS	0600
Maintenance Packet Type	7030
Apollo Domain	8019 (Ethernet)
Novell NetWare IPX	8137 (Ethernet)
AppleTalk Phase 1	809B
AppleTalk ARP Phase 1	80F3
Loopback assistance	9000

Example:

```
add prot-filter dsap (used for DSAP packets)
Protocol Type in hex (1 - 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:
```

Example:

```
add prot-filter ether (used for Ethernet packets)
Protocol Type in hex (600 - 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:
```

Example:

```
add prot-filter snap (used for SNAP packets)
Address (in 10-digit hex) [0000000800]?
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:
```

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 one hop

4.1 Accessing the ASRT Environments

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 you use a port number that is not used for any other LAN/WAN 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 by using the **enable source-routing** command.

Example:

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

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

Add **M**

Add static address entries to the bridging router’s permanent database.

Syntax: add
 static-entry . . .
 destination-address-filter

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 2 3
MAC address [00-00-00-00-00-00]?
Input port, 0 for all [0]?
Output port, 0 for none [0]? 2
Output port, 0 to end [0]?
```

4.1 Accessing the ASRT Environments

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]?
```

Cache **M**

Display the contents of a selected bridging port routing cache. If the port does not possess a cache, the following message appears:

Port X does not have a cache.

Syntax: cache *port#*

Example:

```
cache 1
Port number [1]? 3
```

MAC Address	MC*	Entry Type	Age	Port(s)
00-00-93-00-C0-D0		PERMANENT	0	3 (FDDI/0)
00-00-00-11-22-33		STATIC	0	3 (FDDI/0)

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 that survive power cycles or system resets.
- **Static** – Entries entered by the user in the console process that do not survive power cycles or system resets and are not effected by the aging timer.

4.1 Accessing the ASRT Environments

- Entry Type (continued)*
- **Dynamic** – Entries dynamically learned by the bridge that do not survive power cycles or system resets and that have an age associated with the entry.
 - **Free** - Locations in database that are free to be filled by address entries.
 - **Unknown** - Entry types unknown to the bridge. May be possible bugs and illegal addresses.

Age Age in seconds of each dynamic entry. Age is decremented at each resolution interval.

Port(s) Specifies the port number associated with that entry and displays the interface name (the interface containing the cache).

Change

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

Delete the following information from your bridging configuration:

- Station address entries to the permanent database
- Specific address mapping for a given protocol

4.1 Accessing the ASRT Environments

- 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 receive the message:

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

Example:

```
delete address
```

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

Deletes specific address mapping for given protocol.

<i>dlh-type</i>	(Data-link-header type) is a choice for DSAP, Ether-type, or SNAP.
<i>type-field</i>	Protocol type field. <ul style="list-style-type: none">• 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.
<i>ga-address</i>	6-byte (12-digit hexadecimal) group/multicast address.

4.1 Accessing the ASRT Environments

Example:

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

port port#

Removes a port from a bridging configuration. Since the **enable bridge** command by default configures all LAN devices to participate in bridging, this command allows you to determine which devices do or do 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 number (*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:

<i>SNAP Packets</i>	Subnetwork Access Protocol with protocol type entered in 10-digit hexadecimal format.
<i>Ether Packets</i>	Ethernet Type with the protocol type entered in a range of 5DD-FFFF (hexadecimal).
<i>DSAP Packets</i>	Destination Service Access Point protocol with the protocol type entered in a range of 1-FE (hexadecimal).

Example:

```
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:
```

4.1 Accessing the ASRT Environments

Example:

```
delete prot-filter ether (used for Ethernet packets)
Protocol Type in hex (600 - 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:

```
delete prot-filter dsap (used for DSAP packets)
Protocol Type in hex (1 - 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:
```

Delete **M**

Delete MAC address entries from the bridging router's filtering database. This also includes dynamic and static address entries. After entering the command, you are prompted for the desired MAC address. You may also follow the command with the MAC address you want to delete and skip the prompt.

Syntax: delete *MAC-address*

Example:

```
delete 00-00-93-10-04-15
MAC address [00-00-00-00-00-00]?
```

Disable **C**

Disable the following bridge functions:

- Bridging functionality entirely
- Creating duplicate frames for mixed bridging environments (network traffic management)
- Mapping between group address and functional address
- Propagating spanning tree explorer frames
- Source routing on a given port
- Receiving spanning tree explorer frames over a tunnel
- Converting source routed frames to transparent frames and vice versa

4.1 Accessing the ASRT Environments

- Transparent (spanning tree) bridging functionality 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 . . .
 fa-ga-mapping
 IBM8209_Spanning_Tree
 spanning-tree-explorer . . .
 source-routing . . .
 sr-tb
 transparent . . .

bridge

Disables bridging functionality entirely. This command does not remove previously configured bridging values.

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.

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 are sent out. Disabling transparent bridging on the interface also disables the creation of transparent frames.

4.1 Accessing the ASRT Environments

Example:

```
disable duplicate TSF
```

fa-ga-mapping

Disables group address-to-functional address (and vice versa) mapping. You can 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 enables an already participating bridge interface to discontinue source-routing.

Example:

```
disable source-routing 2
```

sr-tb

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

Example:

```
disable sr-tb
```

transparent *port#*

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

4.1 Accessing the ASRT Environments

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

Example:

```
disable transparent 2
```

Enable **C**

Enable the following bridging functions:

- Bridging functionality entirely
- Creating duplicate frames for mixed bridging environments (network traffic management)
- Mapping between group address and functional address
- Propagating Spanning Tree Explorer Frames
- Source routing on a given port
- Receiving Spanning Tree Explorer Frames over a tunnel
- Converting source routed frame to transparent frame
- Transparent (spanning tree) bridging functionality 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  
fa-ga-mapping  
IBM8209_Spanning_Tree  
spanning-tree-explorer . . .  
source-routing . . .  
sr-tb . . .  
transparent . . .
```

bridge

Enables transparent bridging functionality on all the LAN devices (interfaces) configured in the bridging router. The port numbers are assigned to each interface as

4.1 Accessing the ASRT Environments

the previous interface number plus 1. For example, if interface 0 is a LAN device its respective port number is 1.

Example:

```
enable bridge
```

duplicate

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.

Example:

```
enable duplicate
```

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 token ring arenas, 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 are 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.

4.1 Accessing the ASRT Environments

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 is disabled. For the first instance of the command, entering the bridge number is required. For subsequent times, this input is not required.

<i>port#</i>	Valid port participating in the bridge configuration.
<i>segment#</i>	12-bit number which 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 operations of source-routing functionality, it is very important that all the bridges attached to this LAN/WAN have the same perspective of the LAN/WAN identification value.
<i>bridge#</i>	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 <i>bridge#</i> be unique on the segment.

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

Example:

```
enable source-routing 2 1
```

sr-tb

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

4.1 Accessing the ASRT Environments

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 functionality 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.

Example:

```
enable sr-tb 1 1 1470
TB-Domain Segment Number in hex (1-FFF) [1]?
Bridge Virtual Segment Number in hex (1-FFF) [1]?
TB-Domain's MTU [1470]?
```

transparent port#

Enables transparent bridging functionality on the given port. Under normal circumstances, this command is not necessary. This command toggles the **disable transparent** command.

Flip **M**

View specific MAC addresses in the canonical and non-canonical format by “flipping” the address bit order. This command is useful for translating IEEE 802.5 addresses in their typical non-canonical format to the canonical format universally used by the bridge console and ELS (and vice versa).

Syntax: flip *MAC-address*

Example:

```
flip 00-00-00-33-44-55
MAC address [00-00-00-00-00-00]? 00000334455
IEEE 802 canonical bit order: 00-00-00-33-44-55
IBM Token-Ring native bit order: 00:00:00:CC:22:AA
```

4.1 Accessing the ASRT Environments

List **C**

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.

Example:

```
list address 00000123456
00-00-00-12-34-56      PERMANENT    Input Port:  ALL PORTS
                                      Output ports:  1, 2, 3, 4
```

Example:

```
list address 00000112233
00-00-00-11-22-33      PERM/SAF    Input Port:  1
                                      Output ports:  1, 2
```

Address

Address entry in 12-digit hexadecimal format.

Entry Type

PERMANENT indicates that the entry is permanent in nature and survives power cycles or system resets. RESERVED indicates that the entry is reserved by the IEEE802.1d 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 is configured.

4.1 Accessing the ASRT Environments

<i>Input Port</i>	Displays the numbers of input port(s) associated with that address entry.
<i>Output Port</i>	Displays the numbers of output port(s) associated with that address entry. Displays NONE/DAF to indicate that Destination Address Filtering applies because no ports were 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 Behaviour: SRT
Bridge Address: 00-00-00-00-00-09             Bridge Priority:(dec) 32768,
(hex):8000
Source Routing Bridge Number:  A             No. of Source routing segments: 3
SRB: Max ARE Hop cnt: 14                     Max STE Hop cnt: 14
SR-TB Conversion: DISABLED TB-Virtual Segment:0x0 MTU for TB-Domain: 0
1:N Source Routing: ACTIVE                   Internal-Virtual Segment:0x1
SRB LF-bit interpretation: EXTENDED FA <=> GA Conversion: ENABLED

Spanning Tree Protocol Participation: IEEE802.1d on all ports
DLS for the bridge: DISABLED

Number of ports added: 3
Port Number: 1 Interface Number: 0 Port Behaviour: SRB Only
Port Number: 2 Interface Number: 1 Port Behaviour: STB & SRB
Port Number: 3 Interface Number: 2 Port Behaviour: STB & SRB
```

<i>Bridge</i>	Indicates current state of bridge. Values are ENABLED or DISABLED.
<i>Bridge Behavior</i>	Indicates method of bridging being used by that bridge. The values include STB for Transparent, SRB for Source Routing, and SR-TB for Source Routing-Transparent conversion bridging.
<i>Bridge address</i>	Bridge address specified by the user (if set).

4.1 Accessing the ASRT Environments

<i>Bridge priority</i>	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.
<i>Source Routing Bridge Number</i>	The unique number identifying a bridge. It is used to distinguish between multiple bridges connecting the same two rings.
<i>Number of Source Routing Segments</i>	Indicates the number of Source Routing bridge segments configured for the Source Routing domain.
<i>SRB: Max ARE/STE Hop cnt</i>	The maximum hop count for frames transmitting from the bridge for a given interface associated with source-routing bridging.
<i>SR-TB</i>	Indicates whether the source-routing/transparent bridge frame conversion function is enabled or disabled.
<i>TB-Virtual Segment</i>	Indicates the segment number of the transparent bridging domain.
<i>MTU for TB-Domain</i>	Specifies the maximum frame size (maximum transmission units) the transparent bridge can transmit and receive.
<i>1:N Source Routing</i>	Indicates the current state of 1:N Source Routing as ACTIVE or NOT ACTIVE.
<i>Internal Virtual Segment</i>	Displays the virtual segment number configured for 1:N SRB bridging.
<i>SRB LF-bit interpretation</i>	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.
<i>FA-GA conversion</i>	Indicates whether FA-GA conversion is enabled or disabled.
<i>spanning tree Protocol Participation</i>	Displays the types of spanning tree protocols that the bridge participates in.
<i>DLS for the bridge</i>	Indicates whether data links for the bridge are enabled or disabled.
<i>Number of ports added</i>	Displays the number of bridge ports added to the bridging configuration.

4.1 Accessing the ASRT Environments

<i>Port Number</i>	Specifies user-defined number assigned to an interface by the Add Port command.
<i>Interface Number</i>	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.
<i>Port Behaviour</i>	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

Displays general information about the bridge's filtering database.

Example:

```
list filtering
Filtering Database Size:  5120
Aging Time (in seconds):  300
Resolution (in seconds):  5
```

<i>Filtering Database Size</i>	The maximum number of entries that can be held in the filtering database.
<i>Aging Time</i>	Amount of time (in seconds) specified for aging out (discarding) dynamic entries in the filtering database.
<i>Resolution</i>	How often dynamic entries are scanned to look for expiration according to the aging timer.

mapping *add-type type-field*

Lists specific address mapping for given protocol.

Example:

```
list mapping SNAP
```

<i>add-type</i>	Choice of either DSAP, Ether (Ethernet), or SNAP.
<i>type-field</i>	Protocol type field.

4.1 Accessing the ASRT Environments

- 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: 1, (hex): 80-01
Port State        : ENABLED
Port Supports: Transparent Bridging Only
Assoc Interface   : 0
Path Cost         : 0
*****
Port Id (dec)      : 128: 2, (hex): 80-02
Port State        : ENABLED
Port Supports: Transparent Bridging Only
Assoc Interface   : 1
Path Cost         : 0
*****
Port Id (dec)      : 128: 3, (hex): 80-03
Port State        : ENABLED
Port Supports: No Bridging
Assoc Interface   : 2
Path Cost         : 0
*****
Port Id (dec)      : 128: 4, (hex): 80-04
Port State        : ENABLED
Port Supports: Source Routing Bridging Only
SRB: Segment Number: 0x188      MTU: 4399      STE: ENABLED
Assoc Interface   : 3
Path Cost         : 0
```


4.1 Accessing the ASRT Environments

```
*****
Port Id (dec)      : 128: 5, (hex): 80-05
Port State        : ENABLED
Port Supports:    Transparent Bridging and Source Routing
SRB: Segment Number: 0x199      MTU: 4399      STE: ENABLED
Duplicates Frames Allowed:      STE: No, TSF: Yes
Assoc Interface   : 4
Path Cost         : 0
*****
```

<i>Port ID</i>	The ID consists of two parts: the port priority and the port number. In the example, 128 is the priority and 1, 2, and 3 is the port number. In hexadecimal format, the low order byte denotes the port number and the high order byte denotes the priority.
<i>Port state</i>	Displays current state of specified port(s). This can be either ENABLED or DISABLED.
<i>Port supports</i>	Displays bridging method supported by that port (for example, transparent bridging, source-routing bridging).
<i>SRB</i>	Displayed only when SRB is enabled and lists source-routing bridging information. This includes the SRB segment number (in hexadecimal), the Maximum Transmission Unit size, and whether the transmission of Spanning Tree Explorer frames is enabled or disabled.
<i>Duplicate Frames Allowed</i>	Displays a breakdown and count of the types of duplicate frames allowed.
<i>Assoc interface</i>	Displays interface number associated with the displayed port.
<i>Path Cost</i>	Cost associated with the port which is use for possible root path cost. The range is 1 to 65535.

prot-filter port#

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

4.1 Accessing the ASRT Environments

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: 8000.000000010203
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 Chapter 3.

- Bridge Identifier* 8-byte value in ASCII format. If you did not set the bridge address prior to displaying this information, the low order six bytes are displayed as zero denoting that the default MAC address is being used. When a bridge was selected as the root bridge, the bridge max age and bridge hello time are transmitted by it to all the bridges in the network through the HELLO BPDUs.

4.1 Accessing the ASRT Environments

<i>Bridge Maximum Age</i>	Maximum age (period of time) that is used to time out spanning tree protocol related information.
<i>Bridge Hello Timer</i>	Time interval between HELLO BPDUs.
<i>Bridge Forward Delay</i>	Time interval used before changing to another state (if this bridge becomes 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 one 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  [5] ?   6

ADDRESS              ENTRY TYPE              PORT MAP
=====
00:00:00:12:13:11   REGISTERED                ALL PORTS
00:00:00:12:13:12   RESERVED                  NONE/DAF
00:00:00:12:13:13   RESERVED                  NONE/DAF
00:00:00:12:13:14   RESERVED                  NONE/DAF
00:00:00:12:13:15   PERMANENT                 NONE/DAF
00-00-00-12-34-56   PERMANENT

Input Port:  ALL PORTS
Output ports: 1,2,3,4
```

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.

4.1 Accessing the ASRT Environments

- **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 that survive power cycles or system resets.
- **Static** – Entries entered by the user in the console process that do not survive power cycles or system resets and are ageless.
- **Dynamic** – Entries learned by the bridge dynamically that do not survive power cycles or system resets and that have an age associated with the entry.
- **Free** – Locations in database that are free to be filled by address entries.

Port Map

Displays outgoing port map for all incoming ports.

List **M**

Display information about the bridging router configuration or to display information about selected configuration or bridging options.

Syntax: list

aadaptive-bridge . . .
bridge . . .
conversion
database
filtering
source-routing
spanning-tree-protocol
transparent
tunnel

4.1 Accessing the ASRT Environments

adaptive-bridge *datagroup-option* [*sub-option*]

Lists all general information regarding the SR-TB bridge that converts between various types of bridging. There are a number of general datagroup options that are displayed under the **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 **list adaptive-bridge** display options.

Example:

list adaptive-bridge config

```
Adaptive bridge:          Disabled
Translation database size: 0
Aging time:               300 seconds
Aging granularity         5 seconds
```

Port	Segment	Interface	State	MTU
1	1	Eth /0	Up	0
2	3	Eth /1	Up	0

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.

4.1 Accessing the ASRT Environments

<i>State</i>	Indicates the current state of the conversion bridge port.
<i>MTU</i>	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
```

<i>Hash Collision Count</i>	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.
<i>Adaptive Database Overflow</i>	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 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.

4.1 Accessing the ASRT Environments

The following examples illustrate each of the **list adaptive-bridge database** command options.

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

```
list adaptive-bridge database address mac-address
list adaptive-bridge database all
list adaptive-bridge database port segment#
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  Interface  Port  Seg  Age  RIF: Type Direct  Length  LF
IBM MAC Address   RIF
```

<i>Canonical address</i>	Lists the MAC address of the node corresponding to this entry. This is displayed in IEEE 802 canonical (hexadecimal) format.
<i>Interface</i>	Displays the name of the network interface that learned this entry.
<i>Port number</i>	Displays the port number of the port that learned this address entry.
<i>Segment</i>	Displays the number of the segment that learned this address.
<i>Age</i>	Displays the entry age in seconds.
<i>RIF Type</i>	Displays the RIF type as SRF, STE, or ARE.
<i>RIF Direction</i>	Displays the RIF direction as Forward or Reverse.
<i>RIF Length</i>	Displays the RIF length in bytes.
<i>RIF LF</i>	Displays the largest frame value encoded in the RIF.
<i>IBM MAC Address</i>	Shows the MAC address of the node corresponding to this entry. This is displayed in the IBM token ring native bit order.
<i>RIF</i>	Displays the Routing Information Field learned from this node.

4.1 Accessing the ASRT Environments

bridge

Lists all general information regarding the bridge router configuration.

Example:

list bridge

```
Bridge ID (prio/add): 32768/08-00-2B-B1-E2-D1
Bridge state:        Enabled
Bridge type:         SRB
Bridge capability:   ASRT
Number of ports:     2
STP Participation:   IBM-SRB proprietary
```

Port	Interface	State	MAC Address	Modes	Maximum		LNM
					MSDU	Segment	
1	TKR /0	Up	08-00-2B-10-AC-E2	SR	2096	010	ENA
2	SL /1	Up	08-00-2B-00-00-00	SR	2054	002	

```
SR bridge number:    1
SR virtual segment:  000
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 hexadecimal address.

Bridge State Indicates whether bridging is enabled or disabled.

Bridge Type Displays the configured bridge type: NONE, SRB, TB, SRT, ADAPT, A/SRB, A/TB, or ASRT.

Number of Ports Displays the number of ports configured.

Port Specifies 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: UP or DOWN.

MAC address Displays the MAC address associated with that port in canonical bit order.

4.1 Accessing the ASRT Environments

<i>Modes</i>	Displays the bridging mode for that port. <ul style="list-style-type: none">• T – Indicates transparent bridging.• SR – Indicates source-routing.• A – Indicates adaptive bridging.
<i>MSDU</i>	Specifies the maximum frame size (including the MAC header but not the FCS field) the source-routing bridge can transmit and receive on this interface.
<i>Segment</i>	Displays the source-routing bridge segment number assigned to the port (if any).
<i>SR bridge number</i>	Displays the user-assigned source-routing bridge number.
<i>SR virtual segment</i>	Displays the source-routing bridge virtual segment number show (if any).
<i>Adaptive segment</i>	Displays the number of the segment which is used in the source-routing domain to route to the transparent domain (through 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 a specific 802.2 SAP type.
- **SNAP** – Displays rules for a specific 802.2 SNAP type.

The following examples break down each of the **list conversion** display options.

```
list conversion all
list conversion ethertype
list conversion SAP
list conversion SNAP
```

4.1 Accessing the ASRT Environments

database *datagroup-option*

Lists the contents of transparent filtering databases. There are a number of datagroups that can be displayed under the **list database** command. These include the following:

- **All-Ports** – 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 stop MAC address is given to define the range. All entries falling within this range are 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.

4.1 Accessing the ASRT Environments

Example:

list database all

MAC Address	MC*	Entry Type	Age	Port(s)
00-00-00-00-AA-AA		Dynamic	295	4 (Eth /0)
00-00-00-12-34-56		Perm/Source filter	2	(TKR /0) -> 3-4
00-00-00-22-33-44		Permanent		1-2
00-00-00-33-44-55		Perm Dest filter		1-2
00-00-00-55-66-77		Perm/Source filter		All
				1-2,4
08-00-2B-00-C0-D0		Registered		3 (FDDI/0)
08-00-2B-10-04-15		Registered		1 (Eth /0)
00-00-93-10-E4-F9		Dynamic	300	1 (Eth /0)
00-00-93-90-04-A6		Dynamic	300	1 (Eth /0)
00-00-A7-10-68-28		Dynamic	270	1 (Eth /0)
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 /0)
08-00-17-00-4D-DA		Dynamic	300	1 (Eth /0)

Note: The fields described below are displayed for all of the **list database** command options.

MAC Address Specifies the address entry in IEEE 802 canonical format.

*MC** An asterisk following an address entry indicates that the entry is flagged as a multicast address.

4.1 Accessing the ASRT Environments

<i>Entry Type</i>	<p>Specifies one of the following types:</p> <ul style="list-style-type: none">• 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 that survive power cycles or system resets.• Static – Entries entered by the user in the console process that do not survive power cycles or system resets and are ageless.• Dynamic – Entries dynamically learned by the bridge that do not survive power cycles or system resets and that have an age associated with the entry.• Free - This type is not used and is not normally seen except in occasional race conditions between the console and the bridge.• Unknown - Unknown entry type. May indicate a software bug. Report the hexadecimal entry type to Customer Service.
<i>Age</i>	<p>Refers to the age (in seconds) of each dynamic entry. Age is decremented at each resolution interval. The true age is calculated by subtracting this value from the max_age value.</p>
<i>Port(s)</i>	<p>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 is “5” for IP tunnel.</p>

Examples:

```
list database dynamic
list database local
list database permanent
list database port port#
list database static
```

4.1 Accessing the ASRT Environments

`list database range`

```
First MAC address [00-00-00-00-00-00]? 0-00-93-00-C0-D0
Last MAC address [FF-FF-FF-FF-FF-FF]? 1-80-C2-00-00-00
```

MAC Address	MC*	Entry Type	Age	Port(s)
00-00-93-00-C0-D0		Registered		3 (FDDI/0)
00-00-93-10-04-15		Registered		1 (Eth /0)
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 that can 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 802.2 SAP type entries.
- **SNAP** – Displays SNAP 802.2 SNAP type 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 that are passed to a routing forwarder to be forwarded.
- **Filtered** – Describes packets that are administratively filtered by user set protocol filters.
- **Bridged and routed** – Describes a protocol identifier containing a protocol entity within the system that 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.

4.1 Accessing the ASRT Environments

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 that can 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 the output from 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 /0    Enabled  4399   Yes           ENA
  3    002      TKR /1    Enabled  4399   Yes
  -    003      Adaptive  Enabled  1470   Yes
```

Bridge number Displays the bridge number (in hexadecimal) assigned to this bridge.

4.1 Accessing the ASRT Environments

<i>Bridge State</i>	Indicates whether bridging is enabled or disabled.
<i>Maximum STE hop count</i>	Displays the maximum hop count for Spanning Tree Explorer Frames transmitting from the bridge for a given interface associated with source-routing bridging.
<i>Maximum ARE hop count</i>	Displays the maximum hop count for All-Route Explorer frames transmitting from the bridge for a given interface associated with source-routing bridging.
<i>Virtual segment</i>	Displays the virtual segment number assigned for 1:N bridging.
<i>Port</i>	Lists the numbers of ports associated with source-routing bridging.
<i>Segment</i>	Lists the assigned segment numbers for networks associated with source-routing bridging.
<i>Interface</i>	Lists the associated interface names. Adaptive is listed for interfaces participating in the SR-TB feature.
<i>State</i>	Lists the current port state (Enabled or Disabled).
<i>MTU</i>	Lists the MTU size set for that port.
<i>STE Forwarding</i>	Indicates whether Spanning Tree Explorers received on this port are forwarded (Yes) and whether STEs from other ports go out this port.

The counters option has further subgroups of information that can 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.

4.1 Accessing the ASRT Environments

Example:

```
list source-routing counters all-ports
SRT>list source counters all-ports
Counters for port 2, segment 001, interface TKR /0:
SRF frames received:      0      sent:      0
STE frames received:      0      sent:      0
ARE frames received:     648      sent:      0
SR frames sent as TB:
TB frames sent as SR:
Dropped, input queue overflow:
Dropped, source address filtering:
Dropped, invalid RIF length:
Dropped, duplicate segment:
Dropped, segment mismatch:
Dropped, Duplicate LAN ID or tree error:
Dropped, STE hop count exceeded:
Counters for port 3, segment 002, interface TKR /1:
SRF frames received:      0      sent:      0
STE frames received:      0      sent:      0
ARE frames received:     825      sent:      0
SR frames sent as TB:
TB frames sent as SR:
Dropped, input queue overflow:
Dropped, source address filtering:
Dropped, invalid RI length:
Dropped, duplicate segment:
Dropped, segment mismatch:
Dropped, Duplicate LAN ID or tree error:
Dropped, STE hop count exceeded:
```

<i>Port</i>	Lists the numbers of ports associated with source-routing bridging.
<i>Segment</i>	Lists the source-routing segment numbers in hexadecimal.
<i>Interface</i>	Lists the name of the network interface.
<i>SRF Frames Received/Sent</i>	Lists the number of Specifically Routed frames received or sent on this bridge.
<i>STE Frames Received/Sent</i>	Lists the number of Spanning Tree Explorer Frames received or sent on this bridge.
<i>ARE Frames Received/Sent</i>	Lists the number of All Route Explorer frames received or sent on this bridge.

4.1 Accessing the ASRT Environments

<i>SR Frames Sent as TB</i>	Lists the number of source-routing frames received on this interface that were sent as Transparent Bridge frames.
<i>TB Frames Sent as SR</i>	Lists the number of Transparent Bridge frames received on this interface that were sent as source-routing frames.
<i>Dropped, input queue</i>	Lists the number of frames arriving on this interface that were dropped because the input queue to the forwarder overflowed.
<i>Dropped, source address filtering</i>	Lists the number of frames arriving on this interface that were dropped because this source address matched a source address filter in the filtering database.
<i>Dropped, destination address filtering</i>	Lists the number of frames arriving on this interface that were dropped because this destination address matched a destination address filter in the filtering database.
<i>Dropped, protocol filtering</i>	Lists the number of frames arriving on this interface that were dropped because their protocol identifier was one that is being administratively filtered.
<i>Dropped, duplicate RIF length</i>	Lists the number of frames arriving on this interface that were dropped because the RIF length is less than 2 or greater than 30.
<i>Dropped, duplicate segment</i>	Lists the number of frames arriving on this interface that were dropped because of a duplicate segment in the RIF. Note: Normal ARE frame behavior causes this counter to increase.
<i>Dropped, segment mismatch</i>	Lists the number of frames arriving on this interface that were dropped because the outgoing segment number does not match any in this bridge.

4.1 Accessing the ASRT Environments

Example:

```
list source-routing counters port port#
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, 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
```

Example:

```
list source-routing counters segment 2
Counters for port 3, segment 002, interface TKR /1:
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
```

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.

4.1 Accessing the ASRT Environments

- **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): 32768/08-00-2B-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
1	FDDI/0	128	10	Enabled
4	Eth /0	128	100	Enabled
128	Tunnel	128	65535	Enabled

Example:

list spanning-tree-protocol counters

```
Time since topology change (seconds)    0
Topology changes:                        3
BPDUs received:                          0
BPDUs sent:                              3866
```

Port	Interface	BPDUs received	BDPU input overflow	Forward transitions
1	FDDI/0	0	0	1
4	Eth /0	0	0	1
128	Tunnel	0	0	1

4.1 Accessing the ASRT Environments

Example:

```
list spanning-tree-protocol state
Designated root (prio/add): 32768/08-00-2B-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
  1 FDDI/0 Forwarding
  4 Eth /0 Forwarding
 128 Tunnel Forwarding
```

Example:

```
list spanning-tree-protocol tree
Port Interface Designated Desig. Designated Des.No.
Interface Root Cost Bridge Port
  1 FDDI/0 32768/00-00-93-00-84-EA 0 32768/00-00-93-00-84-EA 80-01
  4 Eth /0 32768/00-00-93-00-84-EA 0 32768/00-00-93-00-84-EA 80-04
 128 Tunnel 32768/00-00-93-00-84-EA 0 32768/00-00-93-00-84-EA 80-80
```

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.

4.1 Accessing the ASRT Environments

Example:

list transparent configuration

```
Filtering database size: 5141
Aging time: 300 seconds
Aging granularity 5 seconds
Port Interface State MTU
  1 FDDI/0 Enabled 0
  4 Eth /0 Enabled 0
 128 Tunnel Enabled 0
```

Example:

list transparent counters all-ports

```
Counters for port 4, interface Eth /0:
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
```

4.1 Accessing the ASRT Environments

Example:

```
list transparent counters port 4
Counters for port 4, interface Eth /0:
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

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.

```
list tunnel bridges
list tunnel config
```

4.1 Accessing the ASRT Environments

LNM **C**

Access the LNM configuration prompt. LNM configuration commands are entered at this new prompt.

Syntax: `lnm`

Example:
`lnm`

NetBIOS-filtering **C M**

Access the NetBIOS filtering configuration or console prompt. NetBIOS Filtering commands are entered at the new prompt.

Syntax: `netbios-filtering`

Example:
`netbios-filtering`

Note: If the NetBIOS filtering feature has not been purchased for your bridging router software load, you receive the following message if you try to use this command:

```
NetBIOS Filtering is not available in this load.
```

Set **C**

Set certain values, functions, and parameters associated with 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

4.1 Accessing the ASRT Environments

- Route Descriptor (RD) limit

Syntax: set

age
 bridge
 LF-bit-interpretation . . .
 port
 maximum-packet-size . . .
 protocol bridge
 protocol port . . .
 route-descriptor-limit . . .

age

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**. The default value for the aging timer is 300 seconds with a range of 1-1,000,000 seconds. The default value for the resolution parameter is 5 with a range of 1 to 60 seconds.

Example:

```
set age
seconds [300] ? 300
resolution [5] ? 5
```

bridge

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 Media 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.

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 has 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 the **Return** key. Do not use dashes or colons to separate each octet.

4.1 Accessing the ASRT Environments

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

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

Example:

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

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 three bits of the routing control field are used. This is the common practice in source-routing bridges that exist today. In extended mode, six 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. Obviously, MSDU value setting has no implication on traditionally transparent media. An MSDU value greater than the packet size configured in the router is 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.

Example:

```
set maximum-packet-size 1 4399
Port Number [1]? set ?
MSDU size [4399]?
MSDU reduced to 4399
```

4.1 Accessing the ASRT Environments

port status

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 is later 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
```

protocol option

Modifies the spanning tree protocol bridge or port parameters for a new configuration or to adjust 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.

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

$2 \times (\text{Bridge Forward Delay} - 1 \text{ second}) \geq \text{Bridge Maximum Age}$

$\text{Bridge Maximum Age} \geq 2 \times (\text{Bridge Hello Time} + 1 \text{ second})$

Example:

```
set protocol bridge
Bridge Max-Age [20] 20
Bridge Hello Time [2] 2
Bridge Forward Delay [15] 15
Bridge Priority [32768] 1
```

<i>Bridge Maximum Age</i>	Maximum age (period of time) used to time out spanning tree protocol related information.
<i>Bridge Hello Timer</i>	Time interval between HELLO BPDUs.
<i>Bridge Forward Delay</i>	Time interval used before changing to another state (if this bridge becomes the root).
<i>Bridge Priority</i>	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.

4.1 Accessing the ASRT Environments

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] ? 1
Port-Path-Cost [1] ? 1
Port Priority [128] ? 1
```

<i>Port Number</i>	Bridge port number; selects the port for which the path cost and port priority is changed.
<i>Path Cost</i>	Cost associated with the port that is used for possible root path cost. The range is 1 to 65535.
<i>Port Priority</i>	Identifies port priority for the specified port. The range is 0 to 255.

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
```

<i>Limit-type</i>	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.
<i>RD-limit-value</i>	Specifies the maximum number of RDs contained in the Routing Information Field (RIF) of the frame type specified by the RD limit type. This field takes values from 0 to 14. The default RD limit value for ARE and STE frames is 14.

4.2 LNM Configuration Commands

Tunnel **C**

Access the Tunnel configuration prompt for a specific tunnel. Tunnel configuration commands are entered at this new prompt. See Section 4.3 in this chapter for an explanation of each of these commands.

Syntax: tunnel *tunnel-id*

Example:
 tunnel 2

Exit **C M**

Return to the previous prompt level.

Syntax: exit

Example:
 exit

4.2 LNM Configuration Commands

This section summarizes and explains the LNM (LAN Network Manager) configuration commands. These commands allow you to configure network parameters for the LNM. Table 4–2 summarized the LNM configuration commands.

Configuration commands are entered at the LNM `config>` prompt. This prompt is accessed by entering the **Inm** command at the `ASRT config>` prompt.

Table 4–2 LNM Configuration Commands

Command	Function
? (Help)	Lists all of the LNM configuration commands, or lists the options associated with specific commands.
Disable	Disables all LNM agents on a specified port or disables specified LNM agents (RPS, CRS, or REM) on a specified port. Also disables the setting of certain LNM parameters from the remote LNM application linked to the bridge. This command applies globally to all instances of LNM within the bridge.

4.2 LNM Configuration Commands

Table 4–2 LNM Configuration Commands (Continued)

Command	Function
Enable	Enables all LNM agents on a specified port or enables specified LNM agents (RPS, CRS, or REM) on a specified port. Also enables the setting of the reporting link passwords from the remote LNM application linked to the bridge. This command applies globally to all instances of LNM within the bridge.
List	Displays the LNM agents that are enabled for the specified port. Also displays passwords that are configured for the reporting links of a specified bridge port.
Set	Sets the password to be used for the specified reporting link number.
Exit	Exits the LNM configuration process and returns to the ASRT environment.

? (Help)

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 ?`

Enable

Enable all LNM agents on a specified port or to enable specified LNM agents (RPS, CRS, or REM) on a specified port.

This command also enables the setting of the reporting link passwords from the remote LNM application linked to the bridge.

Syntax: enable

AGENT port#

lnm port#

configuration-remote-change

AGENT port#

Enables the specified LNM agent (either RPS, CRS, or REM) on the specified port. If the interface is not a token ring then the message `Port number XX is not token ring` is displayed. If this is the case, the command has no effect.

4.2 LNM Configuration Commands

If the port is not configured, the message `Port number XX does not exist` is displayed and the command has no effect.

If the specified agent is already enabled for the specified port, the message `Already enabled` is displayed.

Example:

```
enable REM
Port Number [1]? 1
```

lnm port#

Enables all LNM agents on the specified bridge port. If the interface is not a token ring, then the message `Port number XX is not token ring` is displayed. If this is the case, the command has no effect.

If the port is not configured, the message `Port number XX does not exist` is displayed and the command has no effect.

If the specified agent is already enabled for the specified port, the message `Already enabled` is displayed.

Example:

```
enable lnm
Port Number [1]? 1
```

configuration-remote-change

Enables the setting of the reporting link passwords from the remote LNM application linked to the bridge. The default setting is to not allow the setting of LNM configuration parameters remotely. This command applies globally to all instances of LNM within the bridge.

Example:

```
enable configuration-remote-change
```

Disable

Disable all LNM agents on a specified port or to disable specified LNM agents (RPS, CRS, or REM) on a specified port.

4.2 LNM Configuration Commands

This command also disables the setting of the reporting link passwords from the remote LNM application linked to the bridge.

Syntax: disable
 AGENT *port#*
 lnm *port#*
 configuration-remote-change

AGENT *port#*

Disables the specified LNM agent (either RPS, CRS, or REM) on the specified port. If the port is not configured then the message LNM not configured for Port XX is displayed. If this is the case, the command has no effect.

Example:

```
disable REM 1
```

lnm *port#*

Disables LNM on the specified bridge port. If the port is not configured, then the message LNM not configured for Port XX is displayed and the command has no effect.

If none of the LNM agents are enabled for the specified port, then the message LNM not configured for Port XX is displayed.

Example:

```
disable lnm 1
```

configuration-remote-change

Disables the setting of the reporting link passwords from the remote LNM application linked to the bridge. This command applies globally to all instances of LNM within the bridge.

Example:

```
disable configuration-remote-change
```

4.2 LNM Configuration Commands

List

Display the LNM agents that are enabled for the specified port. This command also displays passwords that are configured for the bridge.

Syntax: list
 password

password

Displays the passwords that are configured for the reporting links of the bridge. Information concerning whether or not the passwords may be changed by the remote LNM application is also displayed.

Example:

```
list password
Reporting Link      Password
0                   00000000
1                   00000000
2                   00000000
3                   00000000
LNM Remote Configuration Change: Enabled
```

Set

Set the password to be used for the specified reporting link number. The link number may be 0, 1, 2, or 3. Link 0 is used for the controlling link. Links 1, 2, and 3 are used for observing links.

The password must consist of six to eight characters, and must match the password used by LNM when it establishes a reporting link with the bridge. If the password is not set for a link, it defaults to the string "00000000."

Syntax: set
 password link# password

Example:

```
set password 1 1 guesswho
```

Exit

Return to the previous ASRT> prompt level.

Syntax: exit

Example:

```
exit
```

4.3 Tunnel Configuration Commands

4.3 Tunnel Configuration Commands

This section summarizes and 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 `Tunnel(#)config>` prompt. This prompt is accessed by entering the **tunnel** command at the `ASRT config>` prompt. Table 4–3 summarizes the Tunnel configuration commands.

Table 4–3 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.
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.
Exit	Exits the tunnel configuration process and returns to the ASRT environment.

4.3.1 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 X.25 or Frame relay configurations so when running multicast applications on the router (the MOSPF tunnel), care must be taken that one of the following conditions occur:

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

4.3 Tunnel Configuration Commands

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 is only used if some of the routers in the network do not like host addresses (occurs in mixed vendor networks).

? (Help)

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

Add the IP address of end stations participating in a unicast or multicast IP addressing configuration.

For IP unicast addressing, the tunneling configuration requires you supply IP addresses of destination bridges. This record is used by the router software to convert the segment number in the RIF (Routing Information Field) 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 Route Explorer (ARE) frames, the second group for sending Spanning Tree Explorer (STE) frames, and the third group for Specifically Routed Frames (SRF).

4.3 Tunnel Configuration Commands

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

Syntax: add
 address *IP-address*

Example:
 add address 128.185.144.37

Delete

Delete the IP address of bridges participating in a unicast or multicast IP addressing configuration.

Syntax: delete
 address *IP-address*

Example:
 delete address 128.185.144.37

List

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
 packet-size

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

4.3 Tunnel Configuration Commands

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
```

packet-size

Lists the frame size of the tunnel.

Example:

```
list packet-size
Frame size for the tunnel 2120
```

Exit

Return to the previous ASRT> prompt level.

Syntax: exit

Example:
exit

Basic Bridging Configurations

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

If you need more information about the ASRT bridge configuration commands, refer to the previous chapter. For more information about the ASRT bridge, refer to the *Routing Protocols Reference Guide*.

5.1 Accessing the ASRT Configuration Environment

For information on how to access the ASRT configuration environment, see Chapter 1 in the *Network Interface Operations Guide*.

Note: After you access the bridging configuration process, you may begin entering configuration commands. Whenever you make a change to a user-configurable interface parameter, you must restart the router for this change to take effect.

5.2 Basic Bridging Configuration Procedures

The philosophy behind the design of the ASRT bridge is to allow the basic configuration of bridging option by using as few commands as possible. 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 functionality beyond transparent bridging is then enabled on a “per interface” basis. Obviously, when source routing is enabled, the usual user input such as segment number, or bridge number is still required and must be entered beyond the basic commands that are explained.

5.2 Basic Bridging Configuration Procedures

5.2.1 Bridging Interfaces

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

- Ethernet
- Token Ring
- FDDI
- Serial line
- Virtual LAN (VLAN)

The Ethernet and FDDI 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 (using a proprietary protocol) for transparent and source routing traffic. It is important to note that a bridge configuration over a serial line must be consistent at both end points. This means that both endpoints are 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.

For information about configuring VLANs, refer to the *DECswitch 900EE Router Installation Guide* and the *DECswitch 900EF Router Installation Guide* (order numbers EK-DEBMP-DN and EK-DEFBA-DN).

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 is covered section 5.2. After completing these tasks, the router must be restarted for the new configuration to take effect.

5.2 Basic Bridging Configuration Procedures

5.2.2 Enabling the Transparent Bridge

Use the following commands to enable transparent bridging:

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

5.2.3 Enabling the Source Routing Bridge

Use the following commands to enable source routing bridging:

- Enable `bridge` to enable bridging on all Local Area Network interfaces. WAN interfaces (serial lines) can be included by using the **add port** command.
- Disable transparent `port#` to disable transparent bridging on all ports.
- 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.

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

Do 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.

5.2.4 Enabling the SR-TB Bridge

Use the following commands to enable SR-TB bridging:

- Enable `bridge` to enable bridging on all Local Area Network interfaces. WAN interfaces (serial lines) can be included by using the **add port** command.
- Disable transparent `port#` to disable transparent bridging on all underlying source routing interfaces.

5.3 ASRT Configuration Matrix

- 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 segment number to represent the *entire* transparent (Ethernet /FDDI) 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 4.

5.3 ASRT Configuration Matrix

With an ASRT Bridge, the collection of configuration parameters for the bridge and all connected interfaces produce a *bridge personality* for that bridge. Table 5–1 provides a guide to the configuration settings needed for each interface type to produce the desired bridge personality to handle your network.

Table 5–1 Bridge Configuration Settings Matrix

BRIDGE PERSONALITY	SR <-> TB CONVERSION ENABLED?	INTERFACE TYPE & BRIDGING METHOD SETTING					Serial Line or Tunnel
		Token Ring 1	Token Ring 2	Ethernet	FDDI		
STB	NO	TB	TB	TB	TB	TB	
SRB	NO	SR	SR	--	--	SR	
STB & SRB	NO	SR	SR	TB	TB	TB or SR	
SR <-> TB	YES	SR	SR	TB	TB	TB	
SR <-> TB	YES	SR	TB	TB	TB	SR	
SRT	NO	SR & TB	SR & TB	TB	TB	SR & TB	
ASRT	YES	SR & TB	SR	TB	TB	SR & TB	
ASRT	YES	SR	SR & TB	TB	TB	SR & TB	
ASRT	YES	SR or TB	SR or TB	TB	TB	SR & TB	

5.3 ASRT Configuration Matrix

Table 5–1 Bridge Configuration Settings Matrix (Continued)

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

Basic LNM Configuration

This section summarizes the procedure necessary for basic configuration of the LNM feature on your Bridging Router:

1. Obtain the MAC addresses required for network manager software.
2. Enable the LNM agents on the bridging router.
3. Check the configuration by displaying enabled LNM agents.

5.3.1 Obtain the MAC Address Required for Network Manager Software

Use the **list lnm status** command at the ASRT console prompt (ASRT>) to obtain the MAC addresses required by the Network Manager software running on the Network Manager Station. For example:

Example:

```
ASRT>list lnm status
Port Number [1]? 1
Port 1
LNM Agents Enabled: RPS CRS REM
Reporting Link      State      LNM Station Address
0                   ACTIVE    10:00:5A:F1:02:37
1                   AVAILABLE
2                   AVAILABLE
3                   AVAILABLE
MAC Addresses to use when configuring LNM Manager:
0:00:C9:08:35:47
40:00:D9:08:35:47
```

5.3 ASRT Configuration Matrix

The MAC addresses displayed on the bottom two lines (shown in bold the example) of the output are used by the Network Manager to configure it to the LNM agents present in the router.

Note: These addresses must be entered exactly as they appear in the output or LNM will not configure correctly.

5.3.2 Enable the LNM Agents on the Bridging Router

Use the **enable lnm** command at the `LNM config>` prompt to enable the LNM agents on the desired port of your bridging router. For example:

Example:

```
LNM config>enable lnm
Port Number [1]? 1
```

The default setting has all LNM agents enabled.

5.3.3 Check the Configuration by Displaying Enabled LNM Agents

Use the **list port** command at the `LNM config>` prompt to display which LNM agents are enabled on your configured port. For example:

Example:

```
LNM config>list port
Port Number [1]? 1
LNM Agents Enabled: RPS CRS REM
```

6

Configuring and Monitoring TCP/IP Host Services

This chapter describes how to configure and monitor the TCP/IP Host Services (TCP/IP Host) protocol and how to use the TCP/IP Host configuration and console (monitoring) commands.

6.1 Accessing the TCP/IP Host Configuration and Console Environments

For information on how to access the TCP/IP Host configuration environment, see the chapter entitled “Getting Started” in the *System Software Guide* or the *Protocol User’s Guide*.

6.2 Basic Configuration Procedures

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

6.2.1 Setting the IP Address

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

6.2.2 Adding a Default Gateway

To configure a default gateway for your bridging router, use the **add default gateway** command. After the bridging router is designated as the default gateway, it exists as the static (permanent) authoritative router that receives all packets with destinations that cannot be found in the routing table.

6.3 TCP/IP Host Configuration and Console Commands

This action generally occurs dynamically through resident routing protocols. The **add default gateway** command is used for instances when static routing is desired (for example, as a precaution against power failures) or in cases when routing information cannot be obtained dynamically.

The assigned IP address and default gateway information is saved immediately.

6.2.3 Enabling TCP/IP Host Services

After assigning the bridging router IP address and default gateway information, use the **enable services** command to enable TCP/IP Host Services.

6.3 TCP/IP Host Configuration and Console Commands

This section summarizes and explains all the TCP/IP Host configuration and console (monitoring) commands. The TCP/IP Host commands allow you to specify and monitor 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. Enter the TCP/IP Host console commands at the `HST>` prompt.

Table 6–1 summarizes the TCP/IP Host configuration and console commands.

Note: You can monitor HST only if you have previously configured it and have not configured IP.

Table 6–1 TCP/IP Host Configuration and Console Commands Summary

Command	Task	Function
? (Help)	Configure/ Monitor	Lists all of the TCP/IP Host configuration commands, or lists the options associated with specific commands.
Add	Configure	Adds a default-gateway.
Delete	Configure	Deletes a default-gateway.
Disable	Configure	Disables TCP/IP Host Services, router discovery processes, and RIP listening.
Dump	Monitor	Displays the current IP routing table. One line is printed for each destination.
Enable	Configure	Enables TCP/IP Host Services, router-discovery processes, and RIP listening.
Interface	Monitor	Displays the IP address configured for each interface.

6.3 TCP/IP Host Configuration and Console Commands

Table 6–1 TCP/IP Host Configuration and Console Commands Summary (Continued)

Command	Task	Function
List	Configure	Lists the current TCP/IP Host configuration.
Ping	Monitor	Continuously pings a given destination, printing a line for each response received.
Routers	Monitor	Displays the list of all IP routers known to the bridging router.
Set	Configure	Sets the bridging router's IP address.
Traceroute	Monitor	Displays the hop-by-hop route to a given destination.
Exit	Configure/ Monitor	Exits the TCP/IP Host configuration or console process and returns to the previous prompt level.

? (Help) **C M**

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 **C**

The default gateway is the IP address of another router in the network. All packets on this router whose destination address is not found in the routing table are routed to this destination.

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.

Syntax: add default-gateway *def-gateway-IP-address*

Example:
`add default-gateway 123.45.67.89`
Default-Gateway address [0.0.0.0]?

6.3 TCP/IP Host Configuration and Console Commands

Delete **C**

Delete default gateways from your bridging router 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 123.45.67.89  
Enter address to be deleted [0.0.0.0]?
```

Disable **C**

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 were 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
```

6.3 TCP/IP Host Configuration and Console Commands

services

Disables the TCP/IP Host Services protocol entirely. By default, TCP/IP Host Services are enabled.

Example:

```
disable services
```

Dump **M**

Display the current IP routing table. One line is printed for each destination. Many of the entries that 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 FFFFFFF0  1   0    BDG/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.

- **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 was 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.
- **EGP** – Indicates routes learned through the EGP protocol.

6.3 TCP/IP Host Configuration and Console Commands

	<ul style="list-style-type: none">• EGPR – Indicates routes learned through the EGP 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.
<i>Dest net</i>	Displays the IP address of the destination network/subnet.
<i>Mask</i>	Displays the IP address mask.
<i>Cost</i>	Displays the Route Cost.
<i>Age</i>	Displays the time that has elapsed since the routing table entry was last refreshed for RIP and EGP routes.
<i>Next Hop</i>	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.
<i>Default gateway</i>	Displays the IP address of the default gateway along with the route type, cost, age, and next hop information associated with that entry.
<i>Routing table size</i>	Displays the current size (in networks and kilobytes) of the current table. Also identifies the number of networks (nets) known to the host.

Enable **C**

Enable the following TCP/IP functions:

- TCP/IP Host Services
- Router discovery processes

6.3 TCP/IP Host Configuration and Console Commands

- RIP listening

Syntax: enable

 rip-listening
 router-discovery
 services

rip-listening

Enables the building of routing table entries that were 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. By default, TCP/IP Host Services are enabled.

Example:
 enable services

Interface **M**

List the IP addresses configured for each interface. 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
BDG/0     128.185.142.16       255.255.255.0
```

6.3 TCP/IP Host Configuration and Console Commands

<i>Interface</i>	Displays a single address as Bridge/0 when TCP/IP Host Services are running over the bridge. When services are disabled, interfaces with their corresponding numbers are displayed.
<i>IP Address</i>	Displays the IP address of the TCP/IP Host Services interface.
<i>Mask</i>	Displays the IP address subnet mask.

Ping **M**

Use the **ping** command to have 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 usually (depending on platform) around 20 milliseconds. The pinging process stops when a character is typed at the console. At that time, a summary of packet loss, round trip time, and number of ICMP destination unreachables received is 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.

Syntax: ping *interface-address*

Example:

```
ping 128.185.142.11
PING 128.185.142.11: 56 data bytes
64 bytes from 128.185.142.11: icmp_seq=0. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=1. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=2. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=3. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=4. time=0. ms
64 bytes from 128.185.142.11: icmp_seq=5. time=0. ms
```

6.3 TCP/IP Host Configuration and Console Commands

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

Routers **M**

Display the list of all IP routers that are known to the bridging router. Routers can be learned through:

- Static configuration (using the above **add default-gateway** command)
- 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 is declared invalid unless it is heard from again).

Syntax: routers

Example:
routers

List **C**

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.

6.3 TCP/IP Host Configuration and Console Commands

<i>Address mask</i>	Displays the current IP-Host IP subnet address mask.
<i>Default Gateway IP-address(es)</i>	Displays the current default gateway IP address.
<i>TCP/IP Host Services</i>	Displays whether TCP/IP Host Services is enabled or disabled.
<i>RIP-LISTENING</i>	Displays whether RIP-LISTENING is enabled or disabled.
<i>Router Discovery</i>	Displays whether Router Discovery is enabled or disabled.

Set **C**

Set the bridging router's IP address. You must assign the bridging router an IP address before enabling TCP/IP Host Services.

Syntax: `set`
 IP-Host address *IP-host-address*

Example:

```
set ip-host address 123.45.67.89
IP-Host address [0.0.0.0]?
Address mask [255.255.0.0]?
```

Traceroute **M**

Display the entire path to a given destination, hop by hop. For each successive hop, the **traceroute** command sends out three probes and prints the IP address of the responder along with the round trip time associated with the response. If a particular probe receives no response, an asterisk is printed. Each line in the display relates to this set of three probes, with the left most number indicating the distance from the router executing the command (in router hops).

6.3 TCP/IP Host Configuration and Console Commands

The traceroute is complete when either 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 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 * * ** Indicates that the router is expecting some form of response from the destination, but the destination is not responding.
*2 * * **

When a probe receives an unexpected result (see the above output example), several indicators can be printed. These indicators are explained in the following table:

IN Indicates that an ICMP Destination Unreachable (net unreachable) was received.

/H Indicates that an ICMP Destination Unreachable (host unreachable) was received.

/P Indicates that an ICMP Destination Unreachable (protocol unreachable) was received.

6.3 TCP/IP Host Configuration and Console Commands

!

Indicates that the destination is reached, but the reply sent by the destination was 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.

Exit

C M

Return to the previous prompt level.

Syntax: exit

Example:
 exit

Using MAC Filtering

This chapter explains MAC filtering, including filtering parameters and filtering tags. It also explains MAC filtering configuring and console (monitoring) commands.

Note: In this chapter, commands for configuring and monitoring, and the update subcommands are listed separately.

7.1 About MAC Filtering

MAC filtering lets you specify packet filters to be applied to packets during processing. Filters are a set of rules applied to a packet to determine how it should be handled during bridging.

Note: MAC filtering is allowed on tunnel traffic.

During the filtering process, packets are either processed, filtered, or tagged during bridging. The following explains these actions:

- **Processed** – Packets are permitted to pass through the bridge unaffected.
- **Filtered** – Packets are not permitted to pass through the bridge.
- **Tagged** – Packets are allowed to pass through the bridge but are marked via a number in the range of 1 to 64 based on a configurable parameter.

A MAC filter is made up of three objects:

- **Filter-item** – A single rule for the address field of a packet. The result is either TRUE (the match was successful) or FALSE (the match was not successful).
- **Filter-list** – Contains a list of one or more filter-items.
- **Filter** – Contains a set of filter-lists.

7.2 Using MAC Filtering Parameters

7.2 Using MAC Filtering Parameters

You can specify some or all of the following parameters in creating a filter:

- Source MAC address or destination MAC address
- Mask to be applied to the packet's fields to be filtered
- Interface number
- Input/output designation
- Include/Exclude/tag designation
- Tag value (if the tag designation is given)

7.2.1 Filter-Item Parameters

You specify the following parameters to construct a filter-item:

- Address Type: *source* or *destination*
- Tag: *Tag-value*
- Address Mask: *Hex-Mask*

Each filter-item specifies an address type (that is, source or destination) to match against the type in the packet with the tokens.

The *address mask* is a MAC address in hex comparing the packet's addresses. The mask is applied to the source or destination MAC address of the packet before comparing it against the specified MAC address.

The mask specifies the bytes that are to be logically ANDed with the bytes in the MAC address. It must be of equal length to the specified MAC address. If no mask is specified, it is assumed to be all 1s.

7.2.2 Filter-List Parameters

The following parameters are used to construct a filter list:

- Name: *ASCII-string*
- Filter-Item List: *filter-item1, ..., filter-itemn*
- Action: INCLUDE, EXCLUDE, TAG(*n*)

A filter-list is built from one or more filter-items. Each filter-list is given a unique name.

7.3 Using MAC Filtering Tags

Applying a filter-list to a packet consists of comparing each filter item in the order by which the filter items were added to the list. If any of the filter items in the list return TRUE then the filter-list returns its designated action.

7.2.3 Filter Parameters

The following parameters are used to construct a filter:

- Filter-list Names: *ASCII-string, ..., ASCII-string*
- Interface Number: *IFC-number*
- Port Direction: *input* or *output*
- Default Action: *include, exclude, or tag*
- Default Tag: *tag value*

A *filter* is constructed by associating a group of filter-list names with an interface number and assigning an input or output designation. The application of a filter to a packet means that each of the associated filter-lists should be applied to packets being received (input) or sent (output) on the specified numbered interface.

When a filter evaluates a packet to an *include* condition, the packet is forwarded.

When a filter evaluates a packet to an *exclude* condition, the packet is dropped.

When a filter evaluates to a *tag* condition, the packet being considered is forwarded with a tag.

An additional parameter of each filter is the default action which is the result of non-match for all of its filter-lists. This default action is include. It can be set to either include, exclude, or tag. In addition, if the default action is tag, a tag value is also given.

7.3 Using MAC Filtering Tags

- MAC Address filtering is handled by a joint effort between bandwidth reservation and the MAC filtering feature (MCF) using *tags*. A user with bandwidth reservation is able to categorize bridge traffic, for example, by assigning a tag to it.
- Tagging is done by creating a filter item at the MAC filtering configuration prompt and assigning a tag to it. This tag is used to set up a bandwidth class for all packets associated with this tag. Tag values must be in the range of 1 to 64.

7.4 Configuring and Monitoring MAC Filtering

DIGITAL Distributed Routing Software supports applying tags only to bridged packets and allows only the MAC address fields of the packet to be used in applying the tag.

- Up to five tagged MAC addresses can be set from 1 to 5. TAG1 will be searched for first, then TAG2, and so on.
- Once a tagged filter is created, it is assigned a class and priority in the Bandwidth Reservation configuration process. Use **tag** at the Bandwidth Reservation to reference the tag.

Tags can also refer to groups as in IP Tunnel. Tunnel end points can belong to any number of groups, and then packets are assigned to a particular group through the tagging feature of MAC address filtering.

7.4 Configuring and Monitoring MAC Filtering

The following sections explain how to access the MAC filtering prompts and the available commands.

Note: In this chapter, commands for configuring and monitoring, and the update subcommands are listed separately.

7.4.1 Accessing the MAC Filtering Prompts

Use the **feature** command from the configuration process to access the MAC filtering configuration commands or from the monitoring process to access the MAC filtering monitoring commands. The **feature** command lets you access configuration and monitoring commands for specific bridging router features outside of the protocol and network interface processes.

Type **?** after the **feature** command to see a listing of the features available for your software release. For example:

Example:

```
Config> feature ?
MAC filtering
Feature name or number [2]
```

7.4 Configuring and Monitoring MAC Filtering

To access the MAC filtering configuration prompt, enter the **feature** command followed by the *feature number (2)* or *short name (MCF)*. For example:

Example:

```
Config> feature mcf
MAC Filtering user configuration
Filter config>
```

To access the MAC filtering monitoring prompt, enter the **feature** command followed by the *feature number (2)* or *short name (MCF)*. For example:

Example:

```
+ feature mcf
MAC Filtering user console
Filter>
```

Once you access the MAC filtering configuration or monitoring prompt, you can begin entering specific commands. To return to the `Config>` prompt, type **exit** at the MAC filtering configuration prompt (`Filter config>`). To return to the * prompt, type **exit** at the filtering monitoring prompt (`Filter>`).

7.4.2 Accessing the MAC Filtering Configuration Commands

This section summarizes and explains the MAC filtering configuration commands. Enter configuration commands at the `Filter config>` prompt. Table 7-1 lists the MAC filtering configuration commands.

Table 7-1 MAC Filtering Configuration Command Summary

Command	Function
? (Help)	Displays the commands available from the current prompt level or the options available from a specific command.
Attach	Adds a filter list to a filter.
Create	Creates a filter list or an <i>input</i> or <i>output</i> filter.
Default	Sets the default action for the filter with a specified <i>filter-number</i> to <i>exclude</i> , <i>include</i> , or <i>tag</i> .
Delete	Removes all information associated with a filter list and frees an assigned string as a name for a new filter list. Also deletes a filter created using the create filter command.
Detach	Deletes a filter-list name from a filter.
Disable	Disables MAC filtering globally or on a per filter basis.
Enable	Enables MAC filtering globally or on a per filter basis.
Exit	Exits the MAC filtering configuration or monitoring process.

7.4 Configuring and Monitoring MAC Filtering

Table 7–1 MAC Filtering Configuration Command Summary (Continued)

Command	Function
List	Lists a summary of statistics and settings for each filter currently running in the router.
Move	Reorders the filter-lists attached to a specified filter.
Reinit	Re-initializes the entire MAC filtering system without affecting the rest of the router.
Set-Cache	Changes the cache size for a filter.
Update	Adds or deletes information from a specific filter-list. Brings you to a menu of appropriate subcommands.

Use the following commands to configure the MAC filtering feature.

? (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



Use the **attach** command to add a filter list to a filter.

A filter is constructed by associating a group of filter lists with an interface number. A filter list is built from one or more filter items.

Syntax: `attach filter-list-name filter-number`

Example:

```
attach atm_list 3
```

7.4 Configuring and Monitoring MAC Filtering

Create **C**

Use the **create** command to create a filter list or an *input* or *output* filter.

Syntax: `create`

```
list filter-list-name
filter input interface-number
filter output interface-number
```

list filter-list-name

Creates a filter list. Name a list by a unique string (*Filter-list-name*) of up to 16 characters. This name is used to identify a filter-list that is being built. This name is also used with other commands associated with the filter-list.

Example:

```
create list newyork
```

filter input/output interface-number

Creates a filter and places it on the network associated with the *input* or *output* direction on the interface given by an interface number. By default this filter is created with no attached filter-lists and has a default action of *include* and *enabled*.

Example:

```
create filter INPUT 2
```

Default **C**

Use the **default** command to set the default action for the filter with a specified filter-number to *exclude*, *include*, or *tag*.

Syntax: `default`

```
exclude filter-number
include filter-number
tag tag-number filter-number
```

exclude filter-number

Sets the default action for the filter with a specified filter-number to *exclude*.

Example:

```
default exclude 3
```

7.4 Configuring and Monitoring MAC Filtering

include *filter-number*

Sets the default action for the filter with a specified *filter-number* to *include*.

Example:

```
default include 3
```

tag *tag-number filter-number*

Sets the default action for the filter with the specified *filter-number* to *tag* and sets the associated tag value to *tag-number*.

Example:

```
default tag 3 15
```

Delete **C**

Use the **delete** command to remove all information associated with a filter-list and to free an assigned string as a name for a new filter-list. If filter-list is attached to a filter that has already been created by the user, then this command will display an error message on the screen without deleting anything. In addition, all filter-items belonging to this list are also deleted.

This command also deletes a filter created using the **create filter** command.

Syntax: `delete`

```
list filter-list  
filter filter-number
```

list *filter-list*

Removes all information associated with a *filter-list* and frees an assigned string as a name for a new filter-list. The *filter-list* must be a string entered by a previous **create list** command.

If the filter-list is attached to a filter that has already been created by the user, then this command will display an error message on the screen without deleting anything. All filter-items belonging to this list are also deleted when this command is used.

Example:

```
delete list newyork
```

filter *filter-number*

Deletes a filter created using the **create filter** command.

Example:

```
delete 3
```

7.4 Configuring and Monitoring MAC Filtering

Detach **C**

Use the **detach** command to delete a filter-list name (*filter-list* parameter) from a filter (*filter-number* parameter).

Syntax: detach
 list *filter-list*

Example:
 detach list newyork

Disable **C**

Use the **disable** command to disable MAC filtering entirely or to disable a particular filter.

Syntax: disable
 all
 filter *filter-number*

all

Disables MAC filtering entirely. Filters are still set as *enabled*, however, if they were enabled previously.

Example:
 disable all

filter *filter-number*

Disables a particular filter. The *filter-number* parameter corresponds to the numbers displayed with **list filters** command.

Example:
 disable filter 3

7.4 Configuring and Monitoring MAC Filtering

Enable **C**

Use the **enable** command to enable MAC filtering entirely or to enable a particular filter.

Syntax: `enable`
 `all`
 `filter filter-number`

all

Enables MAC filtering entirely although filters themselves may still set to disabled.

Example:
`enable all`

filter filter-number

Enables a particular filter. The *filter-number* parameter corresponds to the numbers displayed with **list filters**.

Example:
`enable filter 3`

Exit **C**

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

Syntax: `exit`

Example:
`exit`

List **C**

Use the **list** command to list all the filter-lists and filters configured by the user. A list of all the filter-lists attached to a filter is not given. Other information displayed includes:

- A list containing the state of the filtering system (enable, disable)
- The set of configured filter-list records
- Each of the configured filter records

7.4 Configuring and Monitoring MAC Filtering

In addition, the following information is displayed for each filter:

- Filter number
- Interface number
- Filter direction (input, output)
- Filter state (enable, disable)
- Filter default action (tag, include, exclude)

This command also generates a list of attached filter-lists for this filter and all subsequent information for the filter.

Syntax: `list`
 `all`
 `filter filter-number`

Example:

```
list all
```

filter filter-number

Generates a list of attached filter-lists for the specified filter and all subsequent information for the filter.

Example:

```
list filter 3
```

Move

Use the **move** command to reorder the filter-lists attached to a specified filter (given by the *filter-number* parameter). The list given by *Filter-list-name1* is moved immediately before the list given by *Filter-list-name2*.

Syntax: `move filter-list-name1 filter-list-name2 filter-number`

Example:

```
move newyork boston 13
```

7.4 Configuring and Monitoring MAC Filtering

Reinit **C**

Use the **reinit** command to reinitialize the entire MAC filtering system from an existing configuration without affecting the rest of the router.

Syntax: `reinit`

Example:
`reinit`

Set-Cache **C**

Use the **set-cache** command to change the default cache size (16) to a number between 4 and 32768.

Syntax: `set-cache cache-size filter-number`

Example:
`set-cache 32 13`

Update **C**

Use the **update** command to add information to or delete information from a specific filter-list. Using this command with the desired *filter-list-name* brings you to the `Filter filter-list-name Config>` prompt for that specific filter list. From this new prompt you can then change information in the specified list.

The new prompt level is used to add or delete filter-items from filter-lists. The order in which the filter-items are specified for a given filter-list is important as it determines the order in which the filter-items are applied to a packet.

Syntax: `update filter-list-name`

Example:
`update newyork`

Table 7–2 lists the MAC filtering update subcommands. Enter these commands at the `Filter filter-list-name Config>` prompt.

7.4 Configuring and Monitoring MAC Filtering

Table 7–2 MAC Filtering Update Subcommand Summary

Command	Task	Function
? (Help)	Configure	Displays all the update subcommands.
Add	Configure	Adds a hexadecimal number to compare against the source or destination MAC address. Adds filter items to a filter list. Adds a filter list to a filter.
Delete	Configure	Removes filter-items from a filter-list.
Exit	Configure	Exits the update subcommand configuration process.
List	Configure	Lists a summary of all the filter lists and filters configured by the user. Also generates a list of attached filter lists for this filter and all subsequent information for the filter.
Move	Configure	Reorders the filter lists attached to a specified filter.
Set-Action	Configure	Sets a filter item to evaluate either <i>include</i> , <i>exclude</i> or <i>tag</i> (with a <i>tag-number</i> option).

? (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



Use the **add** subcommand to add filter-items to a filter-list. This command specifically lets you add a hexadecimal number to compare against the source or destination MAC address.

The order in which the filter-items are added to a given filter-list is important as it determines the order in which the filter-items are applied to a packet.

Each use of the **add** subcommand creates a filter-item within the filter-list. The first filter-item is assigned *filter-item-number* 1, the next one is assigned number 2, and so forth. After an **add**, the router displays the number of the filter-item just added.

7.4 Configuring and Monitoring MAC Filtering

The first match that occurs stops the application of filter-items, and the filter-list evaluates to either *include*, *exclude* or *tag*, depending on the designated action of the filter-list. If none of the filter-items of a filter-list produce a match, then the default action (*include*, *exclude* or *tag*) of the filter is returned.

Syntax: add

source *hex-MAC-addr hex-Mask*
 destination *hex-MAC-addr hex-Mask*

source *hex-MAC-addr hex-Mask*

Adds a hexadecimal number (with no 0x in front, a maximum of 16 numbers, and an even number of hex numbers) to compare against the source MAC address.

The *hex-mask* parameter must be the same length as *hex-MAC-address* and is logically ANDed with the designated MAC address in the packet. The default *hex-mask* argument is all binary 1s.

The *hex-MAC-addr* parameter can be specified in canonical or non-canonical bit order. A canonical bit order is specified as just a hex number (e.g., 000003001234) It may also be represented as a series of hex digits with a dash between every two digits (e.g., 00-00-03-00-12-34).

A non-canonical bit order is specified as a series of hex digits with a colon between every two digits (e.g., 00:00:C9:09:66:49). MAC addresses of filter-items will always be displayed using either dash or colon to distinguish canonical from non-canonical representations.

Example:

```
add source 00-00-03-00-12-34 00-00-00-00-12-34
```

destination *hex-MAC-addr hex-Mask*

Acts exactly like **add source**, except that the match is made against the destination rather than source MAC address of the packet.

Example:

```
add destination 00-00-03-00-12-34 00-00-00-00-12-34
```

Delete

Use the **delete** subcommand to remove filter-items from a filter-list. You delete filter items by specifying the *filter-item-number* assigned to the item when it was added.

7.4 Configuring and Monitoring MAC Filtering

When the **delete** command is used, any gap created in the number sequence is filled in. For example, if filter-items 1, 2, 3, and 4 exist and filter-item 3 is deleted, then filter-item 4 is renumbered to 3.

Syntax: `delete filter-item-number`

Example:

```
delete 3
```

Exit **C**

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

Syntax: `exit`

Example:

```
exit
```

List **C**

Use the **list** command to print out a listing of all the filter-item records represented in canonical and non-canonical form. The following information about each filter item is displayed:

- MAC address and address mask in canonical or non-canonical form
- Filter-item numbers
- Address type (source or destination)
- Filter-list action is given

Syntax: `list`

`canonical`

`noncanonical`

canonical

Lists all the filter-item records in a filter-list, giving the item numbers, the address type (SRC, DST), the MAC address in canonical form, and the address mask in canonical form. In addition gives the filter-list action.

Example:

```
list canonical
```

7.4 Configuring and Monitoring MAC Filtering

non-canonical

Lists all the filter-item records in a filter-list, giving the item numbers, the address type (SRC, DST), the MAC address in non-canonical form and the address mask in non-canonical form. In addition gives the filter-list action.

Example:

```
list non-canonical
```

Move **C**

The **move** command reorders filter-items within the filter-list. The filter-item whose number is specified by *filter-item-name1* is moved and renumbered to be just before *filter-item-name2*.

Syntax: move *filter-item-name1* *filter-item-name2*

Example:

```
move 2 4
```

Set-Action **C**

The **set-action** command lets you set a filter-list to evaluate to either the *include*, *exclude* or *tag* (with a *tag-number* option) condition. If one of the filter-items of the filter-list matches the contents of the to be filtering, the filter-list will evaluate to the specified condition. The default setting is *include*.

Syntax: set-action
 include
 exclude
 tag *tag-number*

Example:

```
set-action EXCLUDE
```

7.5 Using the MAC Filtering Console Commands

7.5 Using the MAC Filtering Console Commands

This section explains the MAC filtering console (monitoring) commands. Enter these commands at the `filter>` prompt. Table 7-3 lists the MAC filtering commands.

Table 7-3 MAC Filtering Console Command Summary

Command	Function
? (Help)	Displays the commands available from the current prompt level or the options available from a specific command.
Clear	Clears the per filter statistics listed in the <code>list filter</code> command.
Disable	Disables MAC filtering globally or on a per filter basis.
Enable	Enables MAC filtering globally or on a per filter basis.
Exit	Exits the MAC filtering configuration or monitoring process.
List	Lists a summary of statistics and settings for each filter currently running in the router.
Reinit	Re-initializes the entire MAC filtering system without affecting the rest of the router.

Use the following commands to monitor the MAC filtering feature.

? (Help) **M**

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 **M**

Use the **clear** command to clear all the per filter statistics listed in the **list filter** *filter-number* command for all the filter objects and all the statistics listed for each filter-list.

7.5 Using the MAC Filtering Console Commands

The command also clears the per filter statistics listed in the **list filter** *filter-number* command for the filter associated with the *filter-number* plus all the statistics listed for each filter-list in this filter.

Syntax: `clear`
 `all`
 `filter filter-number`

all

Clears all statistics listed in the **list filter** *filter-number* command for each filter object and each filter-list.

Example:
`clear all`

filter *filter-number*

Clears the per filter statistics listed in the **list filter** *filter-number* command for the filter associated with the *filter-number* plus all the statistics listed for each filter-list in this filter.

Example:
`clear filter 6`

Disable

Use the **disable** command to disable MAC filtering entirely or to disable a particular filter.

Syntax: `disable`
 `all`
 `filter filter-number`

all

Disables MAC filtering entirely. Filters are still set as *enabled*, however, if they were enabled previously.

Example:
`disable all`

7.5 Using the MAC Filtering Console Commands

filter *filter-number*

Disables a particular filter. The *filter-number* parameter corresponds to the numbers displayed with **list filters** command.

Example:

```
disable filter 3
```

Enable **M**

Use the **enable** command to enable MAC filtering entirely or to enable a particular filter.

Syntax: enable
 all
 filter *filter-number*

all

Enables MAC filtering entirely although filters themselves may still set to disabled.

Example:

```
enable all
```

filter *filter-number*

Enables a particular filter. The *filter-number* parameter corresponds to the numbers displayed with **list filters**.

Example:

```
enable filter 3
```

Exit **M**

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

Syntax: exit

Example:

```
exit
```

List **M**

Use the **list** command to list all the filter-lists and filters configured by the user. A list of all the filter-lists attached to a filter is not given. Other information displayed includes:

7.5 Using the MAC Filtering Console Commands

- A list containing the state of the filtering system (enable, disable)
- The set of configured filter-list records
- Each of the configured filter records

In addition, the following information is displayed for each filter:

- Filter number
- Interface number
- Filter direction (input, output)
- Filter state (enable, disable)
- Filter default action (tag, include, exclude)

This command also generates a list of attached filter-lists for this filter and all subsequent information for the filter.

Syntax:

```
list
                all
                filter filter-number
```

Example:

```
list all
```

filter filter-number

Generates a list of attached filter-lists for the specified filter and all subsequent information for the filter.

Example:

```
list filter 3
```

Reinit **M**

Use the **reinit** command to reinitialize the entire MAC filtering system from an existing configuration without affecting the rest of the router.

Syntax: `reinit`

Example:

```
reinit
```

Configuring and Monitoring IP Tunneling

This chapter describes how to use the IP tunneling configuration and console (monitoring) commands.

8.1 Accessing the IP Tunnel Commands

Enter commands for specifically defined tunnels at the `Tunnel(#)config>` prompt. To get this prompt type **tunnel** at the `ASRT config>` or `ASRT>` prompt. Table 8–1 lists the tunnel commands.

Table 8–1 Tunnel Configuration and Console Command Summary

Command	Task	Function
? (Help)	Configure/ Monitor	Lists the commands available from the current prompt level. Depending on the prompt, Help lists either the configuration (<code>ASRT config></code>) or the monitoring (<code>ASRT></code>) commands.
Add	Configure	Adds the IP address of destination bridges participating in an IP unicast or multicast addressing configuration for bridging over IP.
Delete	Configure	Deletes the IP address of a destination bridge participating in an IP unicast or multicast addressing configuration for bridging over IP.
Join	Configure	Establishes the router as a member of one or more multicast groups.
Leave	Configure	Removes the router as a member of multicast groups.
List	Configure/ Monitor	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 bytes) of bridging packets being routed through an IP tunnel and whether or not multicast addressing is enabled or disabled.

8.1 Accessing the IP Tunnel Commands

Table 8–1 Tunnel Configuration and Console Command Summary (Continued)

Command	Task	Function
Set	Monitor	Sets a base multicast address for the router.
Exit	Configure/ Monitor	Exits the tunnel configuration process and returns to the ASRT <code>config></code> prompt. Exits the tunnel monitoring process and returns to the ASRT> prompt.

8.1.1 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 X.25 or Frame Relay configurations, so when running multicast applications on the router (e.g., the MOSPF tunnel) take care if one of the following conditions hold:

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

You can ensure the first condition by using the IP **set router-id** configuration command. You can ensure the second condition by using the IP **set internal-ip-address** configuration command.

In all cases, the second option is preferable. Use the first only if some of the routers in the network do not like host addresses (this might happen in mixed vendor networks).

8.1 Accessing the IP Tunnel Commands

? (Help) **C M**

Lists the commands available from the current prompt level. After a specific command lists its options.

Syntax: ?

Example:
list ?

Add **C**

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 you to supply IP addresses of destination bridges. The router software uses this record to convert the segment number in the RIF (Routing Information Field) in a source-routed frame to the corresponding IP address of the destination bridge. For transparent bridging frames, the record 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-route-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*

Example:
add address 128.185.144.37

8.1 Accessing the IP Tunnel Commands

Delete **C**

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`

Example:
`delete address 128.185.144.37`

Join **C**

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, and is defined by an integer tag. A bridge can belong to only one group type for each tag. A bridge cannot belong, for example, to both peer group 1 and server group 1.

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

server-group group-number

Makes the router a member of the server group with the given group number.

Example:
`join server-group 7`

client-group group-number

Makes the router a member of the client group with the given group number.

Example:
`join client-group 3`

peer-group group-number

Makes the router a member of the peer group with the given group number.

Example:
`join peer-group 5`

8.1 Accessing the IP Tunnel Commands

Leave **C**

Use the **leave** command to remove the router from membership in multicast groups.

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

server-group group-number

Removes the router from membership in the server group with the given group number.

Example:

```
leave server-group 7
```

client-group group-number

Removes the router from membership in the client group with the given group number.

Example:

```
leave client-group 3
```

peer-group group-number

Removes the router from membership in the peer group with the given group number.

Example:

```
leave peer-group 5
```

List **C**

Displays the IP addresses of bridges participating in an IP unicast or multicast addressing configuration for tunneling over IP. Displays the current size of IP packets being sent through the tunnels and whether or not IP is enabled or disabled.

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.

Syntax: `list`
 `address`
 `all`

8.1 Accessing the IP Tunnel Commands

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
```

List **M**

Displays the IP addresses of bridges participating in an IP unicast or multicast addressing configuration for tunneling over IP. Displays the current size of IP packets being sent through the tunnels and whether or not IP is enabled or disabled.

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.

Syntax: list tunnel
 bridges
 config

bridges

Use the *bridges* option to list the source routing segment and address of bridges.

Example:

```
list tunnel bridges
SR segment IP address
```


8.1 Accessing the IP Tunnel Commands

config

Use the *config* option to list the tunnel configuration.

Example:

```
list tunnel config
Tunnel IP addresses:
10.1.108.11
Frame size for the tunnel 2120
```

Set **M**

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.

Example:

```
set base-multicast-address 224.10.0.0
```

Exit **C M**

Use the **exit** command to return to the ASRT> prompt.

Syntax: exit

Example:

```
exit
```

Using NetBIOS

This chapter describes the DIGITAL implementation of NetBIOS over bridged networks.

9.1 About NetBIOS

NetBIOS was designed only for use on a LAN. It is not a routable protocol and is typically bridged.

NetBIOS relies on broadcast frames for most of its functions. While this may not present a problem in LAN environments, these broadcasts can be costly in internetwork environments by causing congestion, as well as increased costs for WAN links.

NetBIOS uses LLC type 1 (LLC1) and LLC type 2 (LLC2) services:

- LLC1 provides connectionless data transfer. It requires name conflict resolution, station status gathering flows, and circuit and connection setup flows.
- LLC2 provides a connection-oriented data transfer that uses I-frame traffic sent on established LLC2 connections.

9.1.1 NetBIOS Names

NetBIOS names are the key to communication between NetBIOS stations. A NetBIOS station must know its name in order to communicate with other NetBIOS stations.

NetBIOS names have 16 ASCII characters. IBM and Microsoft reserve the 16th character of the NetBIOS name.

There are two types of NetBIOS names:

- *Individual* names represent a single NetBIOS client or server and should be unique within the NetBIOS network.

9.2 Reducing NetBIOS Traffic

- *Group* names represent a group of NetBIOS stations (an OS/2 LAN Server domain, for example). These names should not be the same as any individual NetBIOS names in the network.

A single NetBIOS station can have multiple individual or group names. The NetBIOS application generates names based on the name or names the network administrator configures.

9.1.2 NetBIOS Name Conflict Resolution

Before a NetBIOS station uses an individual NetBIOS name, it makes sure that the name is unique. To do so, the station repeatedly broadcasts a Name Conflict Resolution frame to all NetBIOS stations. If the station does not receive a response, it assumes the name is unique and it uses the name.

9.1.3 NetBIOS Session Setup Procedure

To establish a NetBIOS session for data transfer types of operations, the NetBIOS client first determines the MAC address of the NetBIOS server. In Token Ring networks, the client also uses source routing techniques to determine the LLC route to the server.

The following is the process of establishing a session:

1. The client repeatedly broadcasts a Spanning Tree Explorer (STE) NetBIOS UI frame that contains the NetBIOS name of the server to all NetBIOS stations.
2. When the server receives the frame, it responds to the client with a corresponding All Routes Explorer (ARE) NetBIOS UI frame that contains the MAC address of and, for Token Ring, the route to the server.
3. The client can then do either of the following:
 - a. Establish an LLC2 connection to communicate with the server using I-frames.
 - b. Begin communicating with the server using specifically- routed NetBIOS UI frames.

9.2 Reducing NetBIOS Traffic

There are two ways to reduce the amount of broadcast NetBIOS traffic:

- Filter as many broadcast NetBIOS frames as possible.
- Forward unfiltered NetBIOS UI frames on as few bridge ports as possible.

9.2 Reducing NetBIOS Traffic

Table 9–1 lists the NetBIOS filters that DIGITAL provides.

Table 9–1 NetBIOS Filters

Filter Type	Filters . . .
MAC Address	Frames by either the source or destination MAC address.
Frame Type	Specific types of NetBIOS frames.
Duplicate Frame	Duplicate frames.
Response	Responses for which the router did not forward a NetBIOS broadcast frame.
Byte	Frames by byte offset and field length within a frame.
Name	Frames by NetBIOS source and destination names.

Once the router filters frames, name caching and route caching control how the router forwards the remaining frames.

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

9.2.1 Frame Type Filtering

Frame type filtering lets you filter the following types of frames:

- Name Conflict Resolution
- General Broadcast
- Trace Control

9.2.1.1 Name Conflict Resolution

NetBIOS stations use Name Conflict Resolution frames to make sure their name is unique. Name Conflict Resolution frames are Add-Name-Query, Add-Group-Name-Query, Add-Name Response, and Name-In-Conflict.

Use the following guidelines to determine when to filter Name Conflict Resolution frames:

- It is critical that the NetBIOS names of stations to which a NetBIOS session is established (typically a server) be unique.
- It is also usually critical that the individual NetBIOS names of stations within the same group (or domain) be unique.

9.2 Reducing NetBIOS Traffic

- It is often not critical that the NetBIOS names of stations from which a NetBIOS session is set up (typically a client) be unique, especially across domains.

For this reason, networks in which there is good control over server names may gain advantage by filtering name conflict resolution frames.

9.2.1.2 General Broadcast

NetBIOS stations use General Broadcast frames to send data to all NetBIOS stations in a network. Stations rarely use this frame, and you can typically filter it. The NetBIOS General Broadcast frame is Datagram-Broadcast.

9.2.1.3 Trace Control Frames

Trace Control frames terminate NetBIOS traces in all NetBIOS stations in a network. This frame is rarely used and you can typically filter it. The NetBIOS Trace Control frame is Terminate-Trace.

9.2.1.4 Configuring Frame Type Filtering

For bridged traffic, the router does not filter any of the above frame types as the default. However, if you are bridging NetBIOS traffic on WAN links, it may be beneficial to filter these frames. To turn frame type filtering on or off for bridging, enter **set filters bridge**.

For example:

```
NetBIOS config>set filters bridge
Filter Name Conflict frames? [No]: yes
Name conflict filtering is           ON
Filter General Broadcast frames? [No]: yes
General broadcast filtering is       ON
Filter Trace Control frames? [No]: yes
Trace control filtering is           ON
```

9.2.2 Duplicate Frame Filtering

When a station sends broadcast frames, it typically sends up to 10 (the default is 6) frames at fixed intervals (the default is 1/2 second).

Duplicate frame filtering causes the router to forward only one instance of each frame within a configurable amount of time.

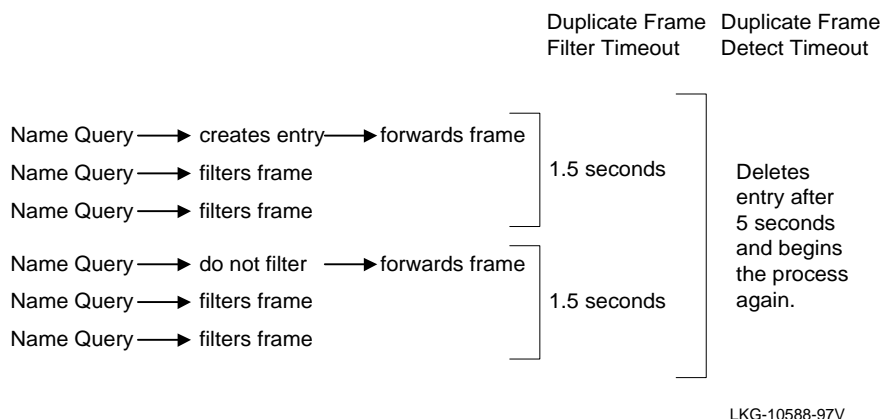
9.2 Reducing NetBIOS Traffic

9.2.2.1 How Duplicate Frame Filtering Works

Duplicate frame filtering works by keeping a database of NetBIOS command frames. These include: Name-Query, Status-Query, Datagram, Add-Name-Query, Add-Group-Name-Query, and Name-In-Conflict.

Figure 9–1 shows the duplicate frame filtering process for bridged traffic. In this example, the router receives six Name-Query frames in half-second intervals. The Duplicate Frame Filter Timeout is set to 1.5 seconds, and the Duplicate Frame Detect Timeout is set to 5 seconds.

Figure 9–1 Duplicate Frame Filtering Process for Bridged Traffic



Here are the steps for duplicate frame filtering:

1. When the router receives a new frame, it creates an entry for that frame in the duplicate frame database and forwards the frame.
2. The router filters any duplicate frames that it receives within the duplicate frame filter timeout (in this case, 1.5 seconds).
3. If the router receives a duplicate frame after the timer expires, it forwards the frame and resets the timer.

The router repeats this process until the duplicate frame detect timer expires.

9.2.2.2 Configuring Duplicate Frame Filtering

Duplicate frame filtering is disabled for bridged traffic as the default. You can enable or disable it for bridging using the **enable duplicate-filtering** and **disable duplicate-filtering** commands.

9.2 Reducing NetBIOS Traffic

To change the timers:

Example:

```
NetBIOS config>set general
WARNING! 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
```

9.2.3 Response Frame Filtering

NetBIOS stations expect a response frame to Name-Query and Status-Query frames. If a station does not receive a response, it continues to send queries.

If the router receives a response to a command frame that it did not forward, it drops the response and does not forward it.

You cannot disable response frame filtering on the router.

9.2.4 NetBIOS Name Caching and Route Caching

Once the router filters all possible NetBIOS broadcast frames, it uses NetBIOS name caching and route caching to reduce the number of frames that the router forwards.

With name caching, the router maintains a database of NetBIOS names and routes. Each time the router receives a Name-Recognized frame, it extracts the MAC address and route and enters that information into the database.

When the router receives a Name-Query or Status-Query, it checks to see if the name being queried is already in its database. If it is, route caching converts the frame from an STE frame to a Specifically-Routed Frame (SRF). A timer on the entry invalidates the database information if the server does not respond before the timer expires.

9.2.4.1 Enabling Caching

Name caching is always enabled. You cannot disable it. The default for route caching is disabled. Enter **enable route-caching** to enable it.

9.2 Reducing NetBIOS Traffic

9.2.4.2 Types of Name Cache Entries

There are three types of name cache entries:

- *Permanent* entries are those that you add at the NetBIOS configuration prompt (`NetBIOS config>`). The router saves permanent entries, and they are still available when you restart the router.
- *Static* entries are those that you enter at the NetBIOS monitoring prompt (`NetBIOS>`). The router does not save static entries, and they are not available after you restart the router.
- *Dynamic* entries are those that the router learns through Name-Query and Name-Recognized processing. A timer removes dynamic entries that have not been referenced within a configurable amount of time. The router does not save dynamic entries, and they are not available after you restart the router.

There are three types of NetBIOS names kept in the name cache:

- *Individual* is a NetBIOS individual name.
- *Group* is a NetBIOS group name.
- *Unknown* means the router does not yet have information about the name, indicating that a search for the name is not complete.

The router also distinguishes between local and remote entries:

- *Local* is an entry the router can reach locally via the bridge network. The router saves the MAC address associated with the name. If route caching is enabled, the router also saves the best LLC route between the router and the NetBIOS station.
- *Remote* is an entry the router can reach remotely. The router saves the best TCP sessions.

9.2 Reducing NetBIOS Traffic

9.2.4.3 Setting Cache Parameters

Use the **set cache-parms** command to change the following parameters:

Example:

```
NetBIOS config>set cache-parms
Significant characters in name [15]?
Best path aging timeout value in seconds [60.0]?
Reduced search timeout value in seconds [1.5]?
Unreferenced entry timeout value in minutes [5000]?
Max nbr local name cache entries [500]?
Max nbr remote name cache entries [100]?
```

```
Cache parameters set
```

See Chapter 11 for more information on the **set cache-parms** command.

9.2.4.4 Displaying Cache Entries

The router provides the following commands that let you view cache entries.

From the NetBIOS configuration prompt, you can use the **list cache** commands in Table 9–2.

Table 9–2 NetBIOS List Cache Configuration Commands

Command	Displays . . .
list cache all	All active entries in the router's name cache, including permanent, static, and dynamic entries.
list cache entry-number	A cache entry according to its entry number.
list cache name	A cache entry for a specific NetBIOS name.
list cache ip-address	A cache entry for a specific IP address.

From the NetBIOS monitoring prompt, you can use the **list cache** commands in Table 9–3.

Table 9–3 NetBIOS List Cache Monitoring Commands

Command	Displays . . .
list cache active	All active entries in the router's name cache, including permanent, static, and dynamic entries.
list cache config	Static and permanent entries. Does not show dynamic entries.
list cache group	Entries that exist for NetBIOS group names.
list cache local	Local cache entries. Local cache entries are those that the router learns over the bridge.
list cache name	A cache entry for a specific NetBIOS name.

9.2 Reducing NetBIOS Traffic

Table 9–3 NetBIOS List Cache Monitoring Commands (Continued)

Command	Displays . . .
list cache remote	Remote cache entries.
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.

9.2.5 NetBIOS Name Filtering

You can use name filters to filter NetBIOS packets that have specific NetBIOS host names. The router examines the source name or destination name field of the following NetBIOS UI packet types:

- Add-Group-Name-Query (source)
- Add-Name-Query (source)
- Datagram (destination)
- Name-Query (source and destination)

For information on how to create name filters, see Chapter 11.

9.2.6 NetBIOS Byte Filtering

Byte filtering lets you filter NetBIOS packets based on fields in the NetBIOS packet.

To build a byte filter, you specify

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

For information on how to create byte filters, see Chapter 11.

10

NetBIOS Filtering and Caching Commands

This chapter describes the NetBIOS filtering and caching configuration and console (monitoring) commands.

10.1 About NetBIOS Configuration and Console Commands

Enter NetBIOS configuration commands at the `NetBIOS config>` prompt. This chapter refers to changes you make at the configuration prompt as *permanent*. Changes you make at this prompt do not take effect immediately. They become part of the router's configuration memory when you restart the router.

Enter NetBIOS console commands at the `NetBIOS>` prompt. This chapter refers to changes you make at the console prompt as *static*. Console commands take effect immediately, but the router does not save them after you restart the router.

10.2 Displaying the NetBIOS Configuration Prompt

You can access the `NetBIOS config>` prompt from the ASRT configuration environment. Changes you make at the `NetBIOS config>` prompt affect bridging.

To display the `NetBIOS config>` prompt from the ASRT configuration environment, enter **protocol asrt** at the `Config>` prompt and **netbios** at the ASRT `config>` prompt.

Example:

```
Config>protocol asrt
Transparent Bridge user configuration
ASRT config>netbios

NetBIOS Support User Configuration

NetBIOS config>
```

10.3 Displaying the NetBIOS Console Prompt

10.3 Displaying the NetBIOS Console Prompt

You can access the `NetBIOS>` prompt from either the ASRT console environment. Changes you make at the `NetBIOS>` prompt affect bridging.

To display the `NetBIOS>` prompt from the ASRT console environment, enter **protocol asrt** at the `+>` prompt and **netbios** at the `ASRT>` prompt.

Example:

```
+ protocol asrt
ASRT>netbios

NetBIOS Support User Console

NetBIOS>
```

10.4 Configuring NetBIOS Filtering and Caching

You can configure the following NetBIOS filtering and caching parameters:

- To configure name caching parameters, enter **set cache-parms**.
- To configure duplicate frame filtering, enter **set general**.
- To configure frame type filtering, enter **set filters bridge**.

10.5 NetBIOS Commands

Table 10–1 lists the NetBIOS commands.

Table 10–1 NetBIOS Commands

Command	Task	Function
? (Help)	Configure/ Monitor	Lists available commands or options.
Add	Configure/ Monitor	Adds cache entries to the router's name cache.
Delete	Configure/ Monitor	Deletes cache entries that you added using the add cache-entry command.
Disable	Configure/ Monitor	Disables duplicate frame filtering and route caching.
Enable	Configure/ Monitor	Enables duplicate frame filtering and route caching.
Exit	Configure/ Monitor	Returns to the previous prompt.

10.5 NetBIOS Commands

Table 10–1 NetBIOS Commands (Continued)

Command	Task	Function
List	Configure/ Monitor	Displays various cache entries and configuration information depending on whether you are at the configuration prompt or the console prompt.
Set	Configure/ Monitor	Configures parameters for name caching, duplicate frame filtering, and frame type filtering. Also, displays the NETBIOS Filter config> prompt.

? (Help)

C M

Lists available commands or options.

Syntax: ?

Example:

```
set ?
  CACHE-PARMS
  FILTERS
  GENERAL
```

Add

C M

Adds a new name cache entry to the router's permanent or static configuration.

From the configuration prompt, adds a permanent entry. From the console prompt, adds a static entry.

Syntax: add
 cache-entry

cache-entry

Note: This command is not supported in this release.

Adds a new entry to the router's name cache. The router ignores entries that you add for ASRT traffic.

Delete

C M

Deletes name cache entries from the router's permanent or static configuration.

Note: This command is not supported in this release.

10.5 NetBIOS Commands

Delete **C**

Note: This command is not supported in this release.

Deletes name cache entries from the router's permanent configuration.

Syntax: delete
 cache-entry

Delete **M**

Deletes name cache entries from the router's static configuration or active cache.

Syntax: delete
 cache-entry

Disable **C M**

Disables duplicate frame filtering or route caching for the bridge.

Syntax: disable
 duplicate-filtering
 route-caching

duplicate-filtering

Disables duplicate frame filtering for bridging.

Example:

```
disable duplicate-filtering
Duplicate frame filtering is          OFF
```

route-caching

Disables route caching for bridging. *Route caching* is the process of converting broadcast frames to Specifically-Routed Frames (SRF) using the entries in the NetBIOS name cache.

Example:

```
disable route-caching
Route caching is                    OFF
```


10.5 NetBIOS Commands

Enable **C M**

Enables duplicate frame filtering or route caching for the bridge.

Syntax: `enable`
 `duplicate-filtering`
 `route-caching`

duplicate-filtering

Enables duplicate frame filtering for bridging.

Example:

```
enable duplicate-filtering  
Duplicate frame filtering is ON
```

route-caching

Enables route caching for bridging. Route caching is the process of converting broadcast frames to Specifically-Routed Frames (SRF) using the NetBIOS name cache.

Example:

```
enable route-caching  
Route caching is ON
```

Exit **C M**

Returns to the previous prompt.

Syntax: `exit`

Example:

```
exit
```

List **C M**

Displays various cache entry and configuration information, depending on whether you are at the configuration prompt or the console prompt.

10.5 NetBIOS Commands

List **C**

Displays all cache entries or displays cache entries by type of entry. Displays filter configuration information or general configuration information.

Syntax: `list`

`cache all`
`cache entry-number`
`cache name`
`cache ip-address`
`filters all`
`filters bridge`
`general`

cache all

cache entry-number *record#*

cache name *name*

cache ip-address

Display cache entries in the router's permanent name cache.

Note: These commands are not supported in this release.

filters all

Displays whether or not frame type filtering is on or off. Use the **set filters bridge** command 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. Enter **set filters bridge** to turn these filters on or off.

10.5 NetBIOS Commands

Example:

```
list filters bridge
Bridge name conflict filtering is      OFF
Bridge general bcast filtering is     OFF
Bridge trace control filtering is     OFF
```

filters dls

Note: This command is not supported in this release.

general

Displays the current NetBIOS caching and filtering configuration.

Example:

```
list general
Bridge-only Information:

Bridge duplicate filtering is          OFF
Bridge duplicate frame filter t/o     1.5 seconds

DLS-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 **M**

Displays various types of cache entries, filter configuration, general configuration information, or statistics on caching and filtering.

Syntax: list

```
cache active
cache config
cache group
cache local
cache name
cache remote
cache unknown
filters all
```

10.5 NetBIOS Commands

filters bridge
filters dlsw
general
statistics cache
statistics frames bridge
statistics frames dlsw
statistics general bridge
statistics general dlsw

cache active
cache config
cache group
cache local
cache name *name*
cache remote
cache unknown

Note: These commands are not supported in this release.

filters all

Displays whether or not frame type filtering is on or off. Use the **set filters bridge** command 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
```

filters dlsw

Note: This command is not supported in this release.

10.5 NetBIOS Commands

general

Displays the current NetBIOS caching and filtering configuration.

Example:

```
list general
Bridge-only Information:

Bridge duplicate filtering is      OFF
Bridge duplicate frame filter t/o  1.5 seconds

DLS-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
```

statistics cache

Lists 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 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
```

10.5 NetBIOS Commands

statistics frames dlsw

Note: This command is not supported in this release.

statistics general bridge

Displays frame counts for bridging.

Example:

```
list statistics general bridge
Frames received                1339
Frames discarded                0
Frames forwarded to bridge     1339
Frames forwarded to DLS       1339
```

statistics general dlsw

Note: This command is not supported in this release.

Set **C M**

Sets name caching parameters, turns frame type filtering on or off, and adjusts duplicate frame filtering timers and frame retry timers. Also, displays the NetBIOS name and byte filtering prompt.

Syntax: set
 cache-parms
 filters bridge
 filters byte
 filters dlsw
 filters name
 general

cache-parms

Sets name caching parameters.

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
```

10.5 NetBIOS Commands

<i>Significant characters in name</i>	<p>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 enter 16, the router includes the 16th character when it looks up cache entries.</p> <p>The default is 15.</p>
<i>Best path aging timeout</i>	<p>Amount of time in seconds the router considers the address and route for a local 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.</p> <p>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.</p> <p>The default is 60 seconds. The range is 1.0 to 100,000.0 seconds.</p>
<i>Reduced search timeout</i>	<p>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.</p> <p>If the router receives a duplicate frame after this timer expires, it assumes the previous route is no longer valid and it widens its search. The router forwards the duplicate frame bridges.</p> <p>The default is 1.5 seconds. The range is 1.0 to 100.0 seconds.</p>
<i>Unreferenced entry timeout</i>	<p>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.</p> <p>The default is 5,000 minutes. The range is 100 to 100,000 minutes.</p>
<i>Max nbr local name cache entries</i>	<p>Maximum number of local entries the router saves in the name cache. Local entries are those that the router learns over the bridge.</p> <p>The default is 500. The range is 1 to 30,000. To optimize memory usage, processor usage, and the amount of broadcast traffic, set this number as close as possible to the total number of NetBIOS stations (servers and clients) that are active on this router's local bridge network.</p>

10.5 NetBIOS Commands

Max nbr remote name cache entries Maximum number of remotely-learned entries, group name entries, and unknown entries.
The default is 100. The range is 1 to 30,000. To optimize memory usage, processor usage, and the amount of broadcast traffic, set this number to the number of remote NetBIOS clients on this router's local bridge network, plus about 25%.

filters bridge

Turns on or off frame-type filtering for bridging.

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>`).

From the `NetBIOS>` prompt, displays the NetBIOS filtering console prompt (`NETBIOS Filter>`).

These prompts allow you to set up NetBIOS name and byte filters.

See Chapter 11 for more information about the commands available at this prompt.

Example:

```
set filters byte
NETBIOS Filtering configuration
NETBIOS Filter config>
```

filters dlsw

Note: This command is not supported in this release.

10.5 NetBIOS Commands

filters name

From the `NetBIOS config>` prompt, displays the `NETBIOS Filter config>` prompt.

From the `NetBIOS>` prompt, displays the `NETBIOS Filter>` prompt.

These prompts allow you to set up NetBIOS name and byte filters.

See Chapter 11 for more information on the commands available at this prompt.

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 section 9.2.2 for more information on how duplicate frame filters work.

Example:

```
set general
WARNING! 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
```

<i>Duplicate frame filter timeout</i>	Applies only 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.
<i>Duplicate frame detect timeout</i>	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.

11

Configuring and Monitoring NetBIOS Name and Byte Filters

This chapter describes the NetBIOS Name and Byte filtering configuration and monitoring commands.

11.1 Displaying the NetBIOS Filtering Prompts

This section describes how to access the NetBIOS Name and Byte filter configuration and monitoring command prompts.

Enter configuration commands at the `NetBIOS Filter config>` prompt. Display this prompt as follows:

Example:

```
Config>protocol asrt
Transparent Bridge user configuration
ASRT config>netbios

NetBIOS Support User Configuration

NetBIOS config>set filter name
NETBIOS Filtering configuration
NETBIOS Filter config>
```

Enter monitoring commands at the `NetBIOS Filter>` prompt. Display this prompt as follows:

Example:

```
ASRT>netbios
NetBIOS Support User Console

NetBIOS>set filter name
NETBIOS Filter>
```

11.2 Setting Up NetBIOS Name and Byte Filters

11.2 Setting Up NetBIOS Name and Byte Filters

A name or byte filter is made up of:

- Filter lists, which are made up of one or more filter items
- Filter items, which specify the NetBIOS names you want to filter

The router compares each filter item against a packet in the order in which you enter the filter items.

You configure NetBIOS name and byte filters for each port and specify whether the filter applies to input or output packets.

The following sections provide examples of how to set up a host name filter and a byte filter. Section 11.3 describes the commands used in these examples.

11.2.1 Example 1: Creating a Name Filter

Use the following procedure as a guideline to create a name filter. Before you begin, display the `NETBIOS Filter config>` prompt.

Example:

```
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>
```

1. Create an empty name filter list.

Enter **create name-filter-list**. The software prompts you to name your filter list.

Example:

```
NetBIOS Filter config>create name-filter-list
Handle for Name Filter List []? boston
```

2. Display the configuration prompt for the filter list you just created.

Enter **update**. The router prompts you for the name of the filter list.

11.2 Setting Up NetBIOS Name and Byte Filters

Example:

```
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.

When you add a filter item, you must specify the following parameters in this order:

- *Inclusive* (bridged) or *exclusive* (dropped).
- *ASCII* or *hex* is how you enter the name.
- *Hostname* is the actual name in either an ASCII or hex format. This entry is case sensitive.
- *Special 16th character* is an optional parameter for use with ASCII strings containing fewer than 16 characters.

The following example adds a filter item to the filter list *boston*, which allows packets containing the name *westboro* (an ASCII string) to be bridged (configured as *inclusive*). No *Special 16th character* is configured.

Example:

```
NetBIOS Name boston config>add inclusive ascii
Hostname []? westboro
Special 16th character in ASCII hex (<CR> for no special char) []?
```

If you do not want to be prompted, enter all parameters as one string on the command line. Use a space between each parameter.

4. Verify the filter item entry.

Enter **list** to verify your entry.

Example:

```
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 step 3 to add filter items to the filter list.

11.2 Setting Up NetBIOS Name and Byte Filters

The order in which you enter filter items is important. 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:

Example:

```
NETBIOS Name boston config>default exclusive
```

6. When you finish adding filter items to the filter list, enter **exit** to return to the NetBIOS Filter config> prompt.

Example:

```
NetBIOS Name boston config>exit  
NetBIOS Filter config>
```

7. Add the filter list to your configuration.

Use the **filter-on** command. When you turn on a name filter, you must specify the following parameters in this order:

- *Input* filters incoming packets or *output* filters outgoing packets.
- *Port#* is the desired configured bridging port number on the router.
- *Filter-list* is the name of the filter list (containing filter items) that you want to be included in this filter.
- Optionally add additional filter lists to the filter. Enter **AND** or **OR** in upper-case letters followed by a filter list name.

The following example adds a name filter comprised of the name filter list boston. The router evaluates all packets input on port 3 according to the filter items in the filter list boston. This means the router bridges all packets input on port 3 that contain the name westboro.

Example:

```
NetBIOS Filter config>filter-on input  
Port Number [1]? 3  
Filter List []? boston
```

11.2 Setting Up NetBIOS Name and Byte Filters

8. Enter **list** to verify the new filter.

Example:

```
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 name and byte filtering on the router.

Enter **enable netbios-filtering**.

Example:

```
NetBIOS Filter config>enable netbios-filtering
```

11.2.2 Example 2: Creating a Byte Filter

Use the following procedure as a guideline for creating a byte filter. Before you begin, display the `NetBIOS Filter config> promt`.

Example:

```
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>
```

11.2 Setting Up NetBIOS Name and Byte Filters

1. Create an empty byte filter list. Use the **create byte-filter-list** command.

Example:

```
NetBIOS Filter config>create byte-filter-list
Handle for Byte Filter List []? westport
```

2. Display the configuration prompt for the filter list you just created.

Enter **update**. The router prompts you for the name of the filter list.

Example:

```
NetBIOS Filter config>update
Handle for Filter List []? westport
Byte Filter List Configuration
NetBIOS Byte westport config>
```

3. Add filter items to the byte filter list.

When you add a filter item, you must specify the following parameters in this order:

- *Inclusive* (bridged) or *exclusive* (dropped).
- *Byte offset* is the number of bytes (in decimal) to offset into the packet the router is filtering. This starts at the NetBIOS header of the packet. Zero specifies that the router examines all bytes in the packet.
- *Hex pattern* is a hexadecimal number the router uses to compare with the bytes starting at the byte offset. See Section 11.3 for syntax rules.
- *Hex mask* if present, must be the same length as hex pattern. It is logically ANDed with the bytes in the packet, starting at byte offset, before the router compares the result with the hex pattern. If you omit the hex mask, the router considers it to be all binary 1s.

The following example adds a filter item to the byte filter list westport that causes the router to bridge packets with a hex pattern 0x12345678 at a byte offset of 0 (configured as *inclusive*). No hex mask is present.

Example:

```
NetBIOS Byte westport config>add inclusive
Byte Offset [0]? 0
Hex Pattern []? 12345678
Hex Mask (<CR> for no mask) []?
```

4. Verify the filter item entry with the **list** command.

11.2 Setting Up NetBIOS Name and Byte Filters

Example:

```
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
```

5. Add additional filter items to the filter list.

Repeat step 3 to add filter items to the filter list.

The order in which you enter filter items is important. 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:

Example:

```
NETBIOS Byte westport config>default exclusive
```

When you have finished adding filter items to the list, enter **exit** to return to the NetBIOS Filter config> prompt.

Example:

```
NetBIOS Byte westport config>exit
NetBIOS Filter config>
```

6. Add the filter to your configuration.

Use the **filter-on** command. When you turn on a byte filter, you must specify the following parameters in this order:

- *Input* filters incoming packets or *output* filters outgoing packets.
- *Port Number* is the desired configured bridging port number.
- *Filter List* is the name of the filter list (containing filter items) that you want included in this filter.
- Optionally add additional filter lists to the filter. Enter **AND** or **OR** in upper-case letters followed by a filter list name.

11.2 Setting Up NetBIOS Name and Byte Filters

The following example adds a byte filter to packets output on port 3. It is comprised of the byte filter list westboro. The router evaluates all packets output on port 3 according to filter items contained in the filter list westport.

Example:

```
NetBIOS Filter config>filter-on output
Port Number [1]? 3
Filter List []? westport
```

7. Verify the new filter.

Enter **list** to verify the filter.

Example:

```
NetBIOS Filter config>list
NetBIOS Filtering: Disabled
```

NetBIOS Filter Lists

```
-----
Handle          Type
newyork         Name
HELLO           Byte
westport        Byte
```

NetBIOS Filters

```
-----
Port #          Direction      Filter List Handle(s)
3               Output        nlist
1               Input         newyork OR HELLO
3               Output        westport
```

8. Globally enable NetBIOS name and byte filtering on the router.

Enter **enable netbios-filtering**.

Example:

```
NetBIOS Filter config>enable netbios-filtering
```

11.3 NetBIOS Name and Byte Filter Commands

11.3 NetBIOS Name and Byte Filter Commands

Table 11–1 lists the NetBIOS name and byte filtering commands.

Table 11–1 NetBIOS Name and Byte Filter Commands

Command	Task	Function
? (Help)	Configure/ Monitor	Lists available commands or options.
Create	Configure	Creates byte filter and name filter lists for NetBIOS filtering.
Delete	Configure	Deletes byte filter and name filter lists for NetBIOS filtering.
Disable	Configure	Disables NetBIOS name and byte filtering on the router.
Enable	Configure	Enables NetBIOS name and byte filtering on the router.
Exit	Configure/ Monitor	Returns you to the previous prompt.
Filter-on	Configure	Assigns a filter to a specific port. You can then apply this filter to NetBIOS packets input or output on the specified port.
List	Configure/ Monitor	Displays all information concerning created filters.
Update	Configure	Adds information to or deletes information from a name or byte filter list.

? (Help)

C M

Lists available commands or options.

Syntax: ?

Example:

```
create ?  
BYTE-FILTER-LIST  
NAME-FILTER-LIST
```

11.3 NetBIOS Name and Byte Filter Commands

Create **C**

Creates a byte filter list or a name filter list.

Syntax: `create`
 `byte-filter-list`
 `name-filter-list`

byte-filter-list *filter-list*

Creates a byte filter list. Give the list a unique name of up to 16 characters. You use this name to identify the filter list.

Example:

```
create byte-filter-list
Handle for Byte Filter List []? newyork
```

name-filter-list *filter-list*

Creates a name filter list. Give the list a unique name of up to 16 characters. You use this name to identify the filter list.

Example:

```
create name-filter-list
Handle for Name Filter List []? boston
```

Delete **C**

Deletes byte filter lists, host name filter lists, and filters. **Delete** removes all information associated with byte and host-name filter lists.

Syntax: `delete`
 `byte-filter-list`
 `name-filter-list`
 `filter input`
 `filter output`

byte-filter-list *filter-list*

Deletes a byte filter list.

Example:

```
delete byte-filter-list
Handle for Byte Filter List []? newyork
```

11.3 NetBIOS Name and Byte Filter Commands

name-filter-list *filter-list*

Deletes a host-name filter list.

Example:

```
delete name-filter-list newyork
Handle for Name Filter List []? boston
```

filter input *port#*

Deletes a filter created with the **filter-on input** command.

Removes all information associated with the filter and fills any resulting gap in filter numbers.

Example:

```
delete filter input
Port Number [1]?
```

filter output *port#*

Deletes a filter created with the **filter-on output** command.

Removes all information associated with the filter and fills any resulting gap in filter numbers.

Example:

```
delete filter output
Port Number [1]?
```

Disable **C**

Globally disables NetBIOS name and byte filtering on the router.

Syntax: **disable** **netbios-filtering**

Example:

```
disable netbios-filtering
```

11.3 NetBIOS Name and Byte Filter Commands

Enable **C**

Globally enables NetBIOS name and byte filtering on the router.

Syntax: enable
 netbios-filtering

Example:
enable netbios-filtering

Exit **C M**

Returns to the previous prompt.

Syntax: exit

Example:
exit

Filter-on **C**

Assigns one or more previously configured filter lists to the input or output of a specific port.

Syntax: filter-on
 input
 output

input port# filter-list AND or OR filter-list

Assigns one or more filter lists to incoming packets on a port. The router applies the resulting filter to all NetBIOS packets input on the specified port.

Port# is a configured bridging port number on the router. The port number identifies this filter. Enter **list** to see a list of port numbers. Use the **create** command to make a filter list. To add additional filter lists to this port, enter AND or OR in all capital letters followed by the filter list name.

The router applies the filter you create with this command to all incoming NetBIOS packets on the specified port. The router evaluates each filter list on the command line from left to right. If a packet matches an inclusive filter, the router bridges the packet. If a packet matches an exclusive filter, the router drops the packet.

If the packet is not one of the types that NetBIOS name or byte filtering supports, the router bridges the packet.

11.3 NetBIOS Name and Byte Filter Commands

Example:

```
filter-on input
Port Number [1]? 2
Filter List []? newyork AND boston
```

output port# filter-list AND or OR filter-list

Assigns one or more filter lists to outgoing packets on a port. The router applies this filter to all NetBIOS packets output on that port.

Port# is a configured bridging port number on the router. The port number identifies this filter. Enter **list** to see a list of port numbers. Use the **create** command to make a filter list. To add additional filter lists to this port, enter AND or OR in all capital letters followed by the filter list name.

The router applies the filter you create with this command to all outgoing NetBIOS packets on the specified port. The router evaluates each filter list on the command line from left to right. If a packet matches an inclusive filter, the router bridges the packet. If a packet matches an exclusive filter, the router drops the packet.

If the packet is not one of the types that NetBIOS name or byte filtering supports, the router bridges the packet.

Example:

```
filter-on output
Port Number [1]? 2
Filter List []? newyork OR boston
```

List **C M**

Displays filter information.

11.3 NetBIOS Name and Byte Filter Commands

List **C**

Displays information on all name and byte filters.

Syntax: list

Example:

```
list
NETBIOS Filtering: Enabled

NETBIOS Filter Lists
-----

      Handle          Type
      -----
      boston          Name
      newport         Byte

NETBIOS Filters
-----
      Port #         Direction      Filter List Handle(s)      1
Input          boston AND newport
      2             Output         boston
```

NetBIOS Filtering Displays whether NetBIOS filtering is enabled or disabled.

NetBIOS Filter Lists Shows the name (handle) of the filter lists, as well as the type, either Name or Byte.

NetBIOS Filters Assigned port number and direction (input or output) of each filter. *Filter List Handle(s)* displays the name(s) of the filter list(s) making up the filter.

11.3 NetBIOS Name and Byte Filter Commands

List **M**

Displays information on all filters, on byte filters, or on name filters.

Syntax: `list`
 `byte-filter-lists`
 `name-filter-lists`
 `filters`

byte-filter-lists

Displays all of the byte filter lists that you created.

Example:

```
list byte-filter-lists
BYTE Filter List Name: newport
BYTE Filter List Default: Inclusive

Filter Item #   Inc/Ex   Byte Offset  Pattern   Mask
              1   Inclusive    2     0x22    0x11
              2   Exclusive    0     0x22    0x22

BYTE Filter List Name: test1
BYTE Filter List Default: Inclusive

Filter Item #   Inc/Ex   Byte Offset  Pattern   Mask
              1   Inclusive    2     0x22    0xFF
```

name-filter-lists

Displays all of the name filter lists that you created.

Example:

```
list name-filter-lists
NAME Filter List Name: boston
NAME Filter List Default: Exclusive

Filter Item #   Type     Inc/Ex     Hostname   Last Char
              1     ASCII   Inclusive  westboro

NAME Filter List Name: newyork
NAME Filter List Default: Inclusive

Filter Item #   Type     Inc/Ex     Hostname   Last Char
              1     ASCII   Inclusive  newyork1
```

11.4 Update Byte-Filter-List Commands

filters

Lists all of the filters that you created and the number of packets the router filtered as a result of those filters.

Example:

```
list filters
```

```
NETBIOS Filtering: Enabled
```

Port #	Direction	Filter List Handle(s)	Pkts Filtered
1	Input	newyork OR boston	0
1	Output	boston AND newport	

Update **C**

Displays the NETBIOS Byte (or Name) **filter-list** config> prompt, which lets you update the specified filter list. At this prompt you can add, delete, list, or move items in byte and name filter lists. You can also set the default of each filter list to *inclusive* or *exclusive*.

Syntax: uppdate *filter-list*

Example:

```
update newyork
```

```
NETBIOS Byte newyork Config>
```

At this new prompt, you can enter several commands. The following sections describe these commands.

11.4 Update Byte-Filter-List Commands

This section describes the commands available at the NETBIOS Byte **filter-list** config> prompt.

11.4 Update Byte-Filter-List Commands

Add

Adds a filter item to the filter list. When you add a filter item, the router numbers the item and displays the number of the filter item you just added.

Syntax: `add`
 `inclusive byte-offset hex-pattern hex-mask`
 `exclusive byte-offset hex-pattern hex-mask`

Note: Adding filter items to filter lists adds to processing time due to the time it takes to evaluate each item in the list. It can affect performance in heavy NetBIOS traffic.

The order in which you enter filter items is important as this determines how the router applies filter items to a packet. The router stops comparing the packet to a filter when it finds the first match.

- *Inclusive* (bridged) or *exclusive* (dropped).
- *Byte offset* is the number of bytes (in decimal) to offset into the packet the router is filtering. This starts at the NetBIOS header of the packet. Zero specifies that the router examines all bytes in the packet.
- *Hex pattern* is a hexadecimal number used to compare with the bytes starting at the byte offset. 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. It is logically ANDed with the bytes in the packet, starting at byte offset, before the router compares the result with the hex pattern. If you omit the hex mask, the router considers it 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 (for example, if the packet is shorter than was intended when setting up a byte-filter list), the router does not apply the filter to the packet. If you use a series of byte filter items to set up a single NetBIOS filter list, then a packet is not 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) []?
```

11.4 Update Byte-Filter-List Commands

The following example shows how to filter Datagram Broadcast Packets.

Example:

```
add exclusive
Byte Offset [0]? 4
Hex Pattern []? 09
Hex Mask (<CR> for no mask) []?
```

Default

Changes the default setting of the filter list to *inclusive* or *exclusive*. If no filter items match the contents of the packet the router considers for filtering, the router forwards or drops the packet, depending on this setting.

Syntax: default
 inclusive
 exclusive

Example:

```
default inclusive
```

Delete

Deletes a filter item from the filter list. The software immediately renumbers the list. To see a list of item numbers, enter **list**.

Syntax: delete *filter-item*

Example:

```
delete
Filter Item Number [1]? 2
```

List

Displays information related to filter items in the filter list.

Example:

```
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

11.5 Update Name-Filter-List Commands

Move

Re-orders filter items within the filter list. To see a list of item numbers, enter **list**.

Syntax: `move filter-item1 filter-item2`

Example:

```
move
Source Filter Item Number [1]? 3
After Destination Filter Item Number [0]? 1
```

Exit

Returns to the previous prompt.

Syntax: `exit`

Example:

```
exit
```

11.5 Update Name-Filter-List Commands

This section lists the commands available at the NETBIOS Name *filter-list* config> prompt.

Add

Adds a filter item to the name filter list. The router compares the following frames and fields with the information you enter with this command:

- ADD_GROUP_NAME_QUERY: Source NetBIOS name field
- ADD_NAME_QUERY: Source NetBIOS name field
- DATAGRAM: Destination NetBIOS name field
- NAME_QUERY: Destination NetBIOS name field

Syntax: `add`

```
                  inclusive or exclusive
                  ASCII host-name or HEX hexstring
                  special-16th-char
```

add inclusive or exclusive ASCII hostname Special-16th-character

Enter the following information with this command:

- *Inclusive* (bridged) or *exclusive* (dropped).

11.5 Update Name-Filter-List Commands

- *Hostname* is an ASCII string up to 16 characters. It can contain any character but the following: . / \ [] : | < > + = ; , space. Use ? to indicate a single character wildcard. Use * as the final character of the name to indicate a wildcard for the remainder of the name. If the name contains fewer than 15 characters, it is padded to the 15th character with ASCII spaces.
- *Special-16th-character* can be used if *hostname* 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 you do not specify a 16th character on a name less than 16 characters, the router uses a ? wildcard for the 16th character.

Example:

```
add inclusive ascii
Hostname []? newport
Special 16th character in ASCII hex (<CR> for no special character)[]?
```

add inclusive or exclusive HEX *hexstring*

Adds a filter item to the name filter list. This command is functionally the same as **add inclusive ascii**. However, you enter the name as a series of hexadecimal numbers (with no 0x in front).

Hexstring must consist of an even number of hexadecimal numbers. Specify a wildcard for a single byte by ?. If you do not supply a full 32 hexadecimal numbers, the router pads ASCII blanks to the 29th and 30th numbers and supplies a wildcard as the 31st and 32nd (16th byte) numbers.

Example:

```
add inclusive hex
Hex String []?
```

Default

Changes the default setting of the filter list to *inclusive* or *exclusive*. If no filter items match the packet the router considers for filtering, the router forwards or drops the packet, depending on this setting.

Syntax: default
 inclusive
 exclusive

Example:

```
default inclusive
```

11.5 Update Name-Filter-List Commands

Delete

Deletes a filter item from the filter list. To see a list of item numbers, enter **list**.

Syntax: `delete filter-item`

Example:

```
delete
Filter Item Number [1]? 4
```

List

Displays information related to items in the specified filter list.

Syntax: `list`

Example:

```
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

Re-orders filter items within the filter list. To see a list of item numbers, enter **list**.

Syntax: `move filter-item1 filter-item2`

Example:

```
move
Source Filter Item Number [1]?
After Destination Filter Item Number [0]?
```

Exit

Returns to the previous prompt.

Syntax: `exit`

Example:

```
exit
```

Index

Page reference numbers in bold type indicate a reference to a command description.

Symbols

? (Help)

- ASRT Bridge configuration command, **4-4**
- IP tunnel configuration command, **8-3**
- LNM configuration command, **4-59**
- MAC filtering configuration
 - command, **7-6**
- MAC filtering monitoring command, **7-17**
- MAC filtering update subcommand, **7-13**
- NetBIOS configuration command, **10-3**
- NetBIOS filtering configuration
 - command, **11-9**
- NetBIOS filtering monitoring
 - command, **11-9**
- NetBIOS monitoring command, **10-3**
- TCP/IP Host Services configuration
 - command, **6-3**
- Tunnel configuration command, **4-64**

Numerics

8029 bridges, 3-5

A

Abbreviating commands, xx
Adaptive Source Routing Transparent Bridge (ASRT), 2-1, 3-1, 4-1

- basic configuration procedures, 5-1
- bit ordering in STB and SRB bridges, 2-37
- bridge-only management, 3-3
- bridging basics, 1-1
- bridging tunnel, 3-1
 - encapsulation and OSPF, 3-2
- configuration matrix, 5-4
- configuring, 2-38, 4-1, 5-1
- description of, 2-36
- eliminating packet size problems, 2-37
- Ethernet packet format translation, 2-7
- MIB support, 3-4
- monitoring, 4-1
- overview, 1-1
 - complex bridges, 1-4
 - CSMA/CD MAC frames, 1-9
 - local bridges, 1-5
 - MAC bridge frame formats, 1-8
 - operation and protocol
 - architecture, 1-5
 - point-to-point links, 1-6
 - remote bridges, 1-5
 - simple bridge, 1-4, 1-5
 - token-ring MAC frames, 1-10
- protocol filtering, 2-20
- pseudo serial ethernet, 3-4

- source routing bridge (SRB), 2–12
 - operation, 2–14
 - source routing frames, 2–14
 - spanning tree explore option, 2–18
 - spanning tree bridges, 2–7
 - spanning tree explore option
 - balancing traffic loads, 2–19
 - simulating a network, 2–19
 - SRB terminology and concepts, 2–21
 - bridge instance, 2–21
 - bridge number, 2–21
 - explorer frames, 2–22
 - interface number, 2–22
 - route, 2–22
 - route discovery, 2–22
 - segment number, 2–22
 - source routing, 2–22
 - SR–TB bridging examples, 2–31
 - SR–TB conversion
 - description of, 2–26
 - operation, 2–28, 2–29
 - TCP/IP Host Services, 3–3
 - terminology and concepts, 2–8, 2–34
 - aging time, 2–8
 - all routes broadcast, 2–34
 - all stations broadcast, 2–35
 - bridge, 2–8, 2–35
 - bridge address, 2–8
 - bridge hello time, 2–8
 - bridge identifier, 2–9
 - bridge maximum age, 2–9
 - bridge number, 2–35
 - bridge priority, 2–9
 - designated port, 2–10
 - destination bridge, 2–10
 - explorer frames, 2–35
 - filtering and permanent databases, 2–10
 - parallel bridges, 2–11
 - path cost, 2–11
 - port, 2–11
 - port ID, 2–11
 - port number, 2–11
 - port priority, 2–11
 - resolution, 2–12
 - ring number, 2–35
 - root bridge, 2–12
 - root port, 2–12
 - route, 2–35
 - route designator, 2–35
 - route discovery, 2–35
 - segment number, 2–35
 - single route broadcasting, 2–36
 - source routing bridging, 2–36
 - spanning tree, 2–12, 2–36
 - transparent bridging, 2–12, 2–36
 - transparent bridge (STB), 2–1
 - network requirements, 2–2
 - operation of, 2–3
 - routers and transparent bridges, 2–2
 - shaping the spanning tree, 2–4
 - transparent-source routing
 - compatibility, 2–37
- Add
- ASRT Bridge configuration command, **4–4**
 - ASRT Bridge console command, **4–14**
 - IP tunnel configuration command, **8–3**
 - MAC filtering update subcommand, **7–13**
 - NetBIOS configuration command, **10–3**
 - NetBIOS monitoring command, **10–3**
 - TCP/IP Host Services configuration command, **6–3**

- Tunnel configuration command, **4-64**
- Address entries
 - dynamic, 4-16, 4-34, 4-42
 - free, 4-16, 4-34
 - permanent, 4-15, 4-34, 4-42
 - registered, 4-15, 4-34, 4-42
 - reserved, 4-33
 - static, 4-34
- ARP
 - with AppleTalk threading, 3-9
 - with IP threading, 3-7
- ASRT Bridge
 - LNM feature prompt, 4-1
 - NetBIOS Filtering feature prompt, 4-2
 - tunnel feature prompt, 4-1
- ASRT Bridge configuration commands
 - ? (Help), **4-4**
 - add, **4-4**
 - and IP tunnel, 4-58
 - change, **4-16**
 - delete, **4-16**
 - disable, **4-19**
 - enable, **4-22**
 - exit, **4-58**
 - functional address to group address
 - mapping, 4-9
 - IP tunnel commands, 4-63
 - list, **4-26**
 - LNM, **4-53**
 - LNM configuration commands, 4-58
 - ? (Help), **4-59**
 - disable, **4-60**
 - enable, **4-59**
 - exit, **4-62**
 - list, **4-62**
 - set, **4-62**
 - netbios-filtering, **4-53**

- port maps explained, 4-7
- set, **4-53**
- summary of, 4-1
- tunnel, **4-58**
- tunnel configuration commands
 - ? (Help), **4-64**
 - add, **4-64**
 - delete, **4-65**
 - exit, **4-66**
 - list, **4-65**
- ASRT Bridge console commands
 - add, **4-14**
 - cache, **4-15**
 - delete, **4-19**
 - flip, **4-25**
 - list, **4-34**
 - netbios-filter, **4-53**
 - summary of, 4-2
- ASRT. *See* Adaptive Source Routing
Transparent Bridge
- Attach, MAC filtering configuration
command, **7-6**

B

- Bridge
 - MAC frame formats, 1-8
 - point-to-point links, 1-6
- Bridging tunnel
 - description of, 3-1
 - encapsulation and OSPF, 3-2
- Brouter, 2-2
- Byte filters, NetBIOS, 9-9

C

- Cache
 - ASRT Bridge console command, **4-15**
 - TCP/IP Host Services console

- command, **6–7**
- Change, ASRT Bridge configuration
 - command, **4–16**
- Clear, MAC filtering monitoring
 - command, **7–17**
- Command summary
 - IP tunnel, 8–1
 - MAC filtering, 7–5
 - MAC filtering console, 7–17
 - NetBIOS, 10–2
 - NetBIOS filtering, 11–1
- Commands
 - abbreviating, xx
 - entering, xx
 - explanation of, xix
- Configuration commands
 - IP tunnel, 8–1
 - MAC filtering, 7–5
 - NetBIOS, 10–2
 - NetBIOS filtering, 11–1
- Console commands, MAC filtering, 7–17
- Conventions, documentation, xvii
- Create
 - MAC filtering configuration
 - command, **7–7**
 - NetBIOS filtering configuration
 - command, **11–10**

D

- Database, permanent, 4–15, 4–42
- Delete
 - ASRT Bridge configuration
 - command, **4–16**
 - ASRT Bridge console command, **4–19**
 - IP tunnel configuration command, **8–4**
 - MAC filtering configuration
 - command, **7–8**
 - MAC filtering update subcommand, **7–14**

- NetBIOS configuration command, **10–4**
- NetBIOS filtering configuration
 - command, **11–10**
- NetBIOS monitoring command, **10–4**
- TCP/IP Host Services configuration
 - command, **6–4**
- Tunnel configuration command, **4–65**
- Detach, MAC filtering configuration
 - command, **7–9**
- Disable
 - ASRT Bridge configuration
 - command, **4–19**
 - LNLM configuration command, **4–60**
 - MAC filtering configuration
 - command, **7–9**
 - MAC filtering monitoring command, **7–18**
 - NetBIOS configuration command, **10–4**
 - NetBIOS filtering configuration
 - command, **11–11**
 - NetBIOS monitoring command, **10–4**
 - TCP/IP Host Services configuration
 - command, 6–4

- Documentation
 - conventions, xvii
 - related, xiv
- Dump, TCP/IP Host Services console
 - command, **6–5**
- Duplicate frame filters for NetBIOS, 9–4
 - configuring, 9–5
 - for bridged traffic, 9–5
 - how they work, 9–5
- Dynamic NetBIOS name cache entries, 9–7

E

- Enable
 - ASRT Bridge configuration
 - command, **4–22**

LNM configuration command, **4-59**

MAC filtering configuration
command, **7-10**

MAC filtering monitoring command, **7-19**

NetBIOS configuration command, **10-5**

NetBIOS filtering configuration
command, **11-12**

NetBIOS monitoring command, **10-5**

TCP/IP Host Services configuration
command, **6-6**

Exit

ASRT Bridge configuration
command, **4-58**

IP tunnel configuration command, **8-7**

LNM configuration command, **4-62**

MAC filtering configuration
command, **7-10**

MAC filtering monitoring command, **7-19**

MAC filtering update subcommand, **7-15**

NetBIOS configuration command, **10-5**

NetBIOS filtering configuration
command, **11-12**

NetBIOS filtering monitoring
command, **11-12**

NetBIOS monitoring command, **10-5**

TCP/IP Host Services configuration
command, **6-12**

Tunnel configuration command, **4-66**

F

Filter-on, NetBIOS filtering configuration
command, **11-12**

Finding information, xii

Flip

ASRT Bridge console command, **4-25**

Frame type filtering for NetBIOS, 9-3

Functional address to group address mapping

ASRT Bridge configuration command, 4-9

I

Information, locating, xii

IP tunnel configuration commands

? (Help), **8-3**

add, **8-3**

delete, **8-4**

exit, **8-7**

join, **8-4**

client-group, **8-4**

peer-group, **8-4**

server-group, **8-4**

leave, **8-5**

client-group, **8-5**

peer-group, **8-5**

server-group, **8-5**

list, **8-5**

address, **8-6**

all, **8-6**

summary of, 4-63

IP tunnel feature and ASRT bridge

description of, 4-1

IP tunnel monitoring commands

list, **8-6**

bridges, **8-6**

config, **8-7**

set, **8-7**

base-multicast-address, **8-7**

J

Join, IP tunnel configuration command, **8-4**

L

LAN Network Manager

See LNM

Leave, IP tunnel configuration command, **8-5**

List

ASRT Bridge configuration
command, **4-26**

ASRT Bridge console command, **4-34**

IP tunnel configuration command, **8-5**

IP tunnel monitoring command, **8-6**

LNLM configuration command, **4-62**

MAC filtering configuration
command, **7-10**

MAC filtering monitoring command, **7-19**

MAC filtering update subcommand, **7-15**

NetBIOS configuration command, **10-6**

NetBIOS filtering configuration
command, **11-14**

NetBIOS filtering monitoring
command, **11-15**

NetBIOS monitoring command, **10-7**

TCP/IP Host Services configuration
command, **6-9**

Tunnel configuration command, **4-65**

LLC2 support, 3-6

LNLM

accessing the configuration prompt, 4-1

ASRT Bridge configuration
command, **4-53**

basic configuration, 5-5

configuration commands

? (Help), **4-59**

disable, **4-60**

enable, **4-59**

exit, **4-62**

list, **4-62**

set, **4-62**

summary of, 4-58

Locating information, xii

M

MAC addresses, 4-54

MAC filtering

overview, 7-1

parameters, 7-2

tags, 7-3

MAC filtering configuration commands

? (Help), **7-6**

attach, **7-6**

create, **7-7**

filter, **7-7**

list, **7-7**

default, **7-7**

exclude, **7-7**

include, **7-8**

tag, **7-8**

delete, **7-8**

filter, **7-8**

list, **7-8**

detach, **7-9**

disable, **7-9**

all, **7-9**

filter, **7-9**

enable, **7-10**

all, **7-10**

filter, **7-10**

exit, **7-10**

list, **7-10**

filter, **7-11**

move, **7-11**

reinit, **7-12**

set-cache, **7-12**

update, **7-12**

MAC filtering monitoring commands

? (Help), **7-17**

all, **7-18**

- clear, **7-17**
 - all, **7-18**
 - filter, **7-18**
- disable, **7-18**
 - filter, **7-19**
- enable, **7-19**
 - all, **7-19**
 - filter, **7-19**
- exit, **7-19**
- list, **7-19**
 - filter, **7-20**
- reinit, **7-20**
- MAC filtering update subcommands
 - ? (Help), **7-13**
 - add, **7-13**
 - delete, **7-14**
 - exit, **7-15**
 - list, **7-15**
 - move, **7-16**
 - set-action, **7-16**
- MAC frames
 - CSMA/CD, 1-9
 - token-ring, 1-10
- Medium Access Control *See* MAC filtering
- Monitoring commands
 - IP tunnel, 8-1
 - MAC filtering, 7-5, 7-17
 - NetBIOS, 10-2
 - NetBIOS filtering, 11-1
- Move
 - MAC filtering configuration
 - command, **7-11**
 - MAC filtering update subcommand, **7-16**
- Multiple spanning trees, 3-4
 - problems, 3-4

N

- Name caching for NetBIOS, 9-6, 9-7
- Name filters, NetBIOS, 9-9
- NetBIOS
 - about configuring and monitoring
 - commands, 10-1
 - byte filters, 9-9
 - configuration commands, 10-2
 - configuring, 10-2
 - displaying configuration prompt, 10-1
 - displaying console prompt, 10-2
 - duplicate frame filters, 9-4
 - configuring, 9-5
 - for bridged traffic, 9-5
 - how they work, 9-5
 - frame type filtering, 9-3
 - monitoring commands, 10-2
 - name caching, 9-6
 - name filters, 9-9
 - names, 9-1
 - overview, 9-1
 - reducing NetBIOS traffic, 9-2
 - response frame filters, 9-6
 - route caching, 9-6
 - session setup, 9-2
 - types of cache entries, 9-7
- NetBIOS configuration commands
 - ? (Help), **10-3**
 - add, **10-3**
 - cache-entry, **10-3**
 - delete, **10-4**
 - cache-entry, **10-4**
 - disable, **10-4**
 - duplicate-filtering, **10-4**
 - route-caching, **10-4**
 - enable, **10-5**

- route-caching, **10-5**
- exit, **10-5**
- list, **10-6**
 - cache all, **10-6**
 - cache entry-number, **10-6**
 - cache ip-address, **10-6**
 - cache name, **10-6**
 - filters all, **10-6**
 - filters bridge, **10-6**
 - general, **10-7**
- set, **10-10**
 - cache-parms, **10-10**
 - filters bridge, **10-12**
 - filters byte, **10-12**
 - filters name, **10-13**
 - general, **10-13**
- NetBIOS filter
 - configuration prompt, **4-2**
- NetBIOS filtering
 - ASRT bridge, **4-1**
- NetBIOS filtering configuration commands
 - ? (Help), **11-9**
 - create, **11-10**
 - byte-filter-list, **11-10**
 - name-filter-list, **11-10**
 - delete, **11-10**
 - byte-filter-list, **11-10**
 - filter input, **11-11**
 - filter output, **11-11**
 - name-filter-list, **11-11**
 - disable, **11-11**
 - netbios-filtering, **11-11**
 - enable, **11-12**
 - netbios-filtering, **11-12**
 - exit, **11-12**
 - filter-on, **11-12**
 - input, **11-12**
 - output, **11-13**
 - list, **11-14**
 - update, **11-16**
- NetBIOS filtering monitoring commands
 - ? (Help), **11-9**
 - exit, **11-12**
 - list, **11-15**
- NetBIOS monitoring commands
 - ? (Help), **10-3**
 - add, **10-3**
 - cache-entry, **10-3**
 - delete, **10-4**
 - cache-entry, **10-4**
 - disable, **10-4**
 - duplicate-filtering, **10-4**
 - route-caching, **10-4**
 - enable, **10-5**
 - duplicate-filtering, **10-5**
 - route-caching, **10-5**
 - exit, **10-5**
 - list, **10-7**
 - cache active, **10-8**
 - cache config, **10-8**
 - cache group, **10-8**
 - cache local, **10-8**
 - cache name, **10-8**
 - cache remote, **10-8**
 - cache unknown, **10-8**
 - filters all, **10-8**
 - filters bridge, **10-8**
 - general, **10-9**
 - statistics cache, **10-9**
 - statistics frames bridge, **10-9**
 - statistics general bridge, **10-10**
 - set, **10-10**
 - cache-parms, **10-10**

- filters bridge, **10–12**
- filters byte, **10–12**
- filters name, **10–13**
- general, **10–13**
- Netbios-filtering
 - ASRT Bridge configuration
 - command, **4–53**
 - ASRT Bridge console command, **4–53**

P

- Permanent NetBIOS name cache entries, 9–7
- Ping, TCP/IP Host Services console
 - command, **6–8**
- Port map
 - cache, 4–16
 - displaying, 4–7
 - list range, 4–34
- Protocol filters
 - Ethernet type, 4–11, 4–18
 - SNAP packets, 4–11, 4–18
- Protocols
 - Adaptive Source Routing Transparent
 - Bridge (ASRT), 4–1, 5–1
 - TCP/IP Host Services, 6–1

R

- Reference documentation, xiv
- Reinit
 - MAC filtering configuration
 - command, **7–12**
 - MAC filtering monitoring command, **7–20**
- Related documentation, xiv
- Response Frame Filters for NetBIOS, 9–6
- Route caching for NetBIOS, 9–6
- Routers
 - TCP/IP Host Services console
 - command, **6–9**

S

- Set
 - ASRT Bridge configuration
 - command, **4–53**
 - IP tunnel monitoring command, **8–7**
 - LNLM configuration command, **4–62**
 - NetBIOS configuration command, **10–10**
 - NetBIOS monitoring command, **10–10**
 - TCP/IP Host Services configuration
 - command, **6–10**
- Set-action
 - MAC filtering update subcommand, **7–16**
- Set-cache
 - MAC filtering configuration
 - command, **7–12**
- Source routing
 - terminology and concepts
 - all routes broadcast, 2–34
 - all stations broadcast, 2–35
 - bridge, 2–35
 - bridge number, 2–35
 - explorer frames, 2–35
 - ring number, 2–35
 - route, 2–35
 - route designator, 2–35
 - route discovery, 2–35
 - segment number, 2–35
 - single route broadcasting, 2–36
 - source routing bridging, 2–36
 - spanning tree, 2–36
 - transparent bridging, 2–36
 - threading, 3–7
- Source Routing Bridge
 - description of, 2–12
 - frame types, 2–14, 2–18
 - operation of, 2–14

- routing information field (RIF), 2–16
- spanning tree explorer frame, 2–17
- terminology and concepts, 2–34
 - bridge instance, 2–21
 - bridge number, 2–21
 - explorer frames, 2–22
 - interface number, 2–22
 - route, 2–22
 - route discovery, 2–22
 - segment number, 2–22
 - source routing, 2–22
- Source Routing Transparent Bridge
 - architecture, 2–24
 - description of, 2–22
 - operation of, 2–24
 - terminology, 2–25
 - explorer frames, 2–25
 - routing information field (RIF), 2–25
 - routing information indicator (RII), 2–26
 - source routing, 2–26
 - spanning tree, 2–26
 - transparent bridging, 2–26
- Spanning Tree Bridge, 2–2, 2–7
 - explore option, 2–18
- Spanning tree network
 - balancing traffic loads, 2–19
 - simulation of, 2–19
- Spanning tree protocol with 8029 bridges, 3–5
- Static NetBIOS name cache entries, 9–7
- Syntax, explanation of, xix

T

- TCP/IP Host Services
 - basic configuration procedures, 6–1
 - configuring, 6–1
- TCP/IP Host Services configuration commands

- ? (Help), 6–3
- add, 6–3
- delete, 6–4
- disable, 6–4
- enable, 6–6
- exit, 6–12
- list, 6–9
- set, 6–10
- summary of, 6–2
- TCP/IP Host Services console commands
 - dump, 6–5
 - interface, 6–7
 - ping, 6–8
 - routers, 6–9
 - traceroute, 6–10
- Threading
 - by AppleTalk end stations, 3–9
 - by DNA end stations, 3–8
 - by IP end stations, 3–7
 - by IPX end stations, 3–9
- Traceroute
 - TCP/IP Host Services console command, 6–10
- Transparent Bridge (STB)
 - bridge ID, 2–3
 - description of, 2–1
 - Ethernet packet format translation, 2–7
 - network requirements, 2–2
 - operation of, 2–3
 - port ID, 2–3
 - root bridge ID, 2–3
 - routers and bridges, 2–2
 - shaping the spanning tree, 2–4
 - spanning tree bridges, 2–7
 - terminology and concepts, 2–8
 - aging time, 2–8
 - bridge, 2–8

- bridge address, 2–8
- bridge hello time, 2–8
- bridge identifier, 2–9
- bridge maximum age, 2–9
- bridge priority, 2–9
- designated bridge, 2–10
- designated port, 2–10
- filtering and permanent
 - databases, 2–10
- parallel bridges, 2–11
- path cost, 2–11
- port, 2–11
- port ID, 2–11
- port number, 2–11
- port priority, 2–11
- resolution, 2–12
- root bridge, 2–12
- root port, 2–12
- spanning tree, 2–12
- transparent bridging, 2–12

Tunnel configuration commands

- ? (Help), **4–64**
- add, **4–64**
- delete, **4–65**
- exit, **4–66**
- list, **4–65**

Tunnel process prompt, 4–1

Tunnel, ASRT Bridge configuration
command, **4–58**

Tunneling, bridge tunnel, 2–13

U

Update

- MAC filtering configuration
command, **7–12**

- NetBIOS filtering configuration
command, **11–16**

