Using NETBuilder® Family Software
Chapter 1 through Chapter 20

Software
Version 9.3

p/n 09-0861-002
Published May 1997

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

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

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

UNITED STATES GOVERNMENT LEGENDS:
If you are a United States government agency, then this documentation and the software described herein are provided to you subject to the following restricted rights:

For units of the Department of Defense:
Restricted Rights Legend: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) for Restricted Rights in Technical Data and Computer Software Clause at 48 C.F.R. 52.227-7013. 3Com Corporation, 5400 Bayfront Plaza, Santa Clara, California 95052-8145.

For civilian agencies:
Restricted Rights Legend: Use, reproduction, or disclosure is subject to restrictions set forth in subparagraph (a) through (d) of the Commercial Computer Software - Restricted Rights Clause at 48 C.F.R. 52.227-19 and the limitations set forth in 3Com Corporation's standard commercial agreement for the software. Unpublished rights reserved under the copyright laws of the United States.

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

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

3Com, Boundary Routing, NETBuilder, NETBuilder II, and SuperStack are registered trademarks of 3Com Corporation. 3TECH and OfficeConnect are trademarks of 3Com Corporation. 3ComFacts is a service mark of 3Com Corporation.

CompuServe is a registered trademark of CompuServe, Inc. IBM, AS/400, LAN Net Manager, OS/2, PS/2, Systems Network Architecture (SNA), and VTAM are registered trademarks of International Business Machines Corporation. Advanced Peer-to-Peer Networking and APPN are trademarks of International Business Machines Corporation. AppleTalk, Macintosh, and LaserWriter are registered trademarks of Apple Computer, Incorporated. XNS is a trademark of Xerox Corporation. VAX, DEC, and DECnet are registered trademarks of Digital Equipment Corporation. TeleVideo is a registered trademark of TeleVideo Corporation. NetWare and Novell are registered trademarks of Novell, Inc. Banyan and VINES are registered trademarks of Banyan Systems. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd. Telenet is a trademark of Telenet Communications Corporation. SPARCsystem is a trademark of SPARC International, Inc. licensed exclusively to Sun Microsystems, Inc. Sun and Solaris are registered trademarks of Sun Microsystems, Inc. SunOS is a trademark of Sun Microsystems, Inc. Link level compression uses Stac LZS compression software, copyrighted by Stac Electronics, (© Stac Electronics, 1991-1995) and protected by one or more patents, including U.S. patent 5,126,739. Stac and LZS compression are registered trademarks of Stac Electronics. Computer Library is a trademark of Ziff Communications. HP is a registered trademark of Hewlett-Packard Company. Hayes and Ultra are registered trademarks of Hayes Microcomputer Products, Inc. Motorola is a registered trademark of Motorola Corporation. Cisco is a registered trademark of Cisco Systems. Wellfleet is a registered trademark of Wellfleet Communications, Inc. NCS is a registered trademark of National Computer Systems.

Other brand and product names may be registered trademarks or trademarks of their respective holders.

Guide written by Lianne Card, Mike Figone, Andrew Mann, Patty Nunn, Francisco Razo, Patrick Sullivan, and Carol Tatick. Edited by Amy Guzules and Pat Vaughn. Technical illustration by Debra Knodel. Production by Debra Knodel and Ramona Boersma.

Printed on recycled paper.

CONTENTS

5 CONFIGURING SOURCE ROUTE BRIDGING

7 BUILDING INTERNET FIREWALLS

8 IP SECURITY OPTIONS

9 CONFIGURING IP MULTICAST ROUTING

10 CONFIGURING APPN INTERMEDIATE SESSION ROUTING

13 IPX ROUTING

15 CONFIGURING DECNET ROUTING

16 OSI ROUTING

17 CONFIGURING VINES ROUTING

20 CONFIGURING UDP BROADCAST HELPER

21 CONFIGURING THE LLC2 DATA LINK INTERFACE

22 CONFIGURING SYNCHRONOUS DATA LINK CONTROL CONNECTIVITY

23 CONFIGURING SDLC AND HDLC TUNNELING FOR SNA NETWORKS

24 CONFIGURING DATA LINK SWITCHING FOR SNA AND NETBIOS NETWORKS

37 CONFIGURING PORT BANDWIDTH MANAGEMENT

42 WIDE AREA NETWORKING USING FRAME RELAY

45 CONFIGURING WIDE AREA NETWORKING USING X.25

50 CONFIGURING CONNECTIONS FOR INCOMING CALLS

51 CONFIGURING LOCAL ACCESS CONTROL

52 MANAGING SESSIONS FOR INCOMING EXTENDED CALLS

D INTERNET ADDRESSING

E NSAP AND PSAP ADDRESSING

F SUPPORTED MIBS

G MACRO FEATURES

H STATISTICS DISPLAYS

ABOUT THIS GUIDE

This guide provides information you need to use NETBuilder® software to operate and configure your bridge/router. This guide includes procedures for configuring your software for bridging, routing, and wide area protocols, according to your network needs.

Supported bridge/routers include:

- NETBuilder II®
- SuperStack® II NETBuilder
- SuperStack II NETBuilder Boundary Router
- OfficeConnect™ NETBuilder

If the information in the release notes shipped with your product differs from the information in this guide, follow the release notes.

Before you use the information in this guide, you must first install the bridge/router according to the hardware installation guide. You must then install and configure NETBuilder software on the bridge/router. If you are upgrading, refer to *Upgrading NETBuilder Family Software*. For a new installation, refer to the appropriate guide for your platform:

Table 1 Software Installation Guides

Platform	Guide
NETBuilder II	*New Installation for NETBuilder II Software*
SuperStack II NETBuilder	*Using SuperStack II NETBuilder Software*
SuperStack II NETBuilder Boundary Router	*Using SuperStack II NETBuilder Boundary Router Software*
OfficeConnect NETBuilder	*Using OfficeConnect NETBuilder Software*

For a comprehensive description of NETBuilder software commands, refer to *Reference for NETBuilder Family Software.*

In this guide, the term bridge/router is used regardless of whether the NETBuilder is configured as a bridge or a router or both.

Audience Description

This guide is intended for network administrators who:

- Have experience planning, maintaining, and troubleshooting local or wide area networks.
- Are familiar with network protocols, bridging and routing, and network management.
- Are responsible for configuring and operating NETBuilder Bridge/Routers.

How to Use This Guide

This guide provides a comprehensive description of NETBuilder configuration and operation. Most users do not need to read the entire guide.

All users should read Chapter 1. Read other chapters if they apply to your network needs. If a chapter discusses a protocol or function you do not intend to use, you do not need to read that chapter.

Chapter 1 provides conceptual information on paths, ports, virtual ports, and logical networks, including port and path numbering, and describes how to configure them.

Chapter 2 describes how to configure ports for Fiber Distributed Data Interface (FDDI).

Chapter 33 describes how to configure a remote bridge/router and central site server so the server sends boot information to the bridge/router as part of the bridge/router automatic startup procedure.

Tabs divide the rest of the guide into sections. Each section consists of one or more chapters.

The **Bridging section** provides information on configuring transparent and translation bridging, bridge filters, source route bridging, source route transparent bridging, source route transparent bridging gateway (SRTG), and route discovery for end system source routing.

The **Routing section** describes how to configure your bridge/router for AppleTalk, Advanced Peer-to-Peer Networking (APPN), DECnet, Internet Protocol (IP) (including IP multicasting, User Datagram Protocol (UDP) Broadcast Helper, and IP security options), IP Firewall , Internetwork Packet Exchange (IPX), Open System Interconnection (OSI), VINES, and Xerox Network Systems (XNS)routing protocols.

The **IBM Internetworking section** describes how to configure the Logical Link Control, type 2 (LLC2) data link interface, data link switching to connect networks running IBM's Systems Network Architecture (SNA) and NetBIOS traffic over Transmission Control Protocol/Internet Protocol (TCP/IP), and tunnels to connect peer networks running SNA.

The **Boundary Routing section** describes Boundary Routing® system architecture and how to implement it.

The **Circuit Switched Services section** describes port bandwidth management of dial-up lines over wide area networks using the Point-to-Point

Protocol (PPP), Phone Line Gateway (PLG) Protocol, and Integrated Services Digital Network (ISDN). It also describes data compression, explains how to schedule recurring events, and how to prioritize packets to be forwarded over a wide area network.

The **Packet Switched Services section** describes wide area networking using packet and cell-switched services, including Frame Relay, Asynchronous Transfer Mode (ATM), Switched Multimegabit Data Service (SMDS), X.25, and local and global switching (X.25 tunneling over IP).

The **Cell Switched Services section** describes Internetworking using ATM for both WAN environments and LAN Emulation.

The **Connection Services section** describes X.25 connection services for outgoing and incoming calls.

The **Network Management section** describes network management activities, configuration, monitoring, alarms, messages, and remote access.

The **Appendixes section** provides additional information about NETBuilder software and related technology that will help you use the product more effectively.

Conventions

Table 2 and Table 3 list conventions that are used throughout this guide.

Table 2 Notice Icons

Icon	Notice Type	Alerts you to...
	Information note	Important features or instructions
	Caution	Risk of personal safety, system damage, or loss of data
	Warning	Risk of severe personal injury

Table 3 Text Conventions

Convention	Description
`Syntax`	The word "syntax" means you must evaluate the syntax provided and supply the appropriate values. Placeholders for values you must supply appear in angle brackets. Example: Enable RIPIP using: `SETDefault !<port> -RIPIP CONTrol = Listen` In this example, you must supply a port number for <port>.
`Commands`	The word "command" means you must enter the command exactly as shown in text and press the Return or Enter key. Example: To remove the IP address, enter: **`SETDefault !0 -IP NETaddr = 0.0.0.0`** *This guide always gives the full form of a command in uppercase and lowercase letters. However, you can abbreviate commands by entering only the uppercase letters and the appropriate value. Commands are not case-sensitive.*
`Screen displays`	This typeface represents information as it appears on the screen.
The words "enter" and "type"	When you see the word "enter" in this guide, you must type something, and then press the Return or Enter key. Do not press the Return or Enter key when an instruction simply says "type."
[Key] names	Key names appear in text in one of two ways: ■ Referred to by their labels, such as "the Return key" or "the Escape key" ■ Written with brackets, such as [Return] or [Esc]. If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press [Ctrl]+[Alt]+[Del].
Menu commands and *buttons*	Menu commands or button names appear in italics. Example: From the *Help* menu, select *Contents*.
Words in *italicized* type	Italics emphasize a point or denote new terms at the place where they are defined in the text.
Words in **bold-face** type	Bold text denotes key features.

1 CONFIGURING PORTS, PATHS, VIRTUAL PORTS, AND LOGICAL NETWORKS

To make full use of NETBuilder® software, you need to understand the concept of ports and paths. This chapter provides conceptual information on ports and paths, including numbering, and describes how to configure ports and paths on the NETBuilder II® bridge/router and SuperStack® II NETBuilder bridge/router platforms.

Some platforms support virtual ports, and the NETBuilder II bridge/router supports port groups (logical networks). Refer to Table 1-1 on page 1-4 for a list of platforms that support virtual ports. If your platform supports virtual ports or port groups, you also need to become familiar with these concepts. This chapter provides conceptual information on virtual ports and logical networks, including numbering, and describes how to configure them on these platforms.

Concepts

This section defines ports and paths and explains how they are numbered on the following platforms:

- NETBuilder II bridge/router
- SuperStack II bridge/router or boundary router

This chapter provides a definition for virtual paths, which are provided by adding a WAN Extender to a NETBuilder II bridge/router, and how they can be used by NETBuilder II ports.

This section also provides a definition of virtual ports and explains how they are numbered on NETBuilder platforms that support them, and defines and explains logical networks, which provide simultaneous bridging and routing for the same network protocol.

The concepts in this section apply regardless of whether the bridge/router is used as a bridge or as a router.

The local and wide area interfaces available to you depend on your hardware platform and its configuration. For information on the types of interfaces your platform offers, refer to its installation guide.

Paths

A *path* is a physical interface that connects a bridge/router to a physical network medium such as an Ethernet bus, a token ring, or a serial line. In an Integrated Services Digital Network (ISDN) environment, a path also represents the channel over which data is transmitted. All NETBuilder bridge/routers provide several paths; each path is associated with a connector, such as an AUI, BNC, RS-232, or RS-449 connector, or a variety of others.

For software purposes, paths are numbered 1, 2, 3, and so on. (The path number may be followed by a letter or a decimal and a channel number. For more information, refer to "Port and Path Numbering on NETBuilder II Multiport Modules" on page 1-13 and to "Port and Path Numbering on a SuperStack II Bridge/Router" on page 1-14. For all SuperStack II bridge/router platforms, the connector configuration and the path number for each connector are fixed. For the NETBuilder II bridge/router, a connector takes its path number from the slot in which its module is installed. For more information on NETBuilder II path numbers, refer to "Port and Path Numbering on a NETBuilder II Bridge/Router" on page 1-11 and to "Port and Path Numbering on NETBuilder II Multiport Modules" on page 1-13.

Ports

A *port* is a logical interface used by the software to represent a connection to a network. By default, there is a one-to-one correspondence between ports and paths, and they are usually numbered alike: for instance, port 1 is associated with path 1. All network traffic received on physical path 1 is treated by the software as arriving on logical port 1, and all traffic that the software transmits through logical port 1 passes through physical path 1. The same is true for the other ports and paths.

This default configuration is called a *static port and path binding*. A *static path* is a path that is mapped to a port. All paths are static by default.

You can redefine the default mapping through software commands. For example, you can redirect network traffic that is being routed through a particular logical port to a different physical path without manually switching cables on the connector.

Each logical port is usually associated with only one physical path. For token ring, Ethernet, Fiber Distributed Data Interface (FDDI), Frame Relay, Asynchronous Transfer Mode (ATM), X.25, and Switched Multimegabit Data Service (SMDS), the path-to-port ratio is always one to one. But for paths connected to serial lines, multiple paths can be associated with and statically bound to a single port if the Point-to-Point Protocol (PPP) or Phone Line Gateway (PLG) Protocol is running over the port.

Paths can also be unbound from their ports and placed in a dial path pool to be shared by more than one port. The paths in the dial pool are called *dynamic paths*. A path in the dial pool can be dynamically bound to a port running PPP when the path is needed for data transfer events associated with dial-up. A dynamic path can also be bound to a port for dial backup purposes such as bandwidth-on-demand or disaster recovery. For more information about the use of the dial path pool, refer to Chapter 37.

Virtual Paths

When you add a WAN Extender to a NETBuilder II bridge/router, it provides virtual paths that can be dynamically bound to a NETBuilder II physical or virtual port if PPP is running over the port. The NETBuilder II bridge/router can currently support up to 75 virtual paths. Because virtual paths are used by ports running PPP, multiple paths can be bound to a single port using the MultiLink Protocol.

Virtual paths can be used for WAN Extender ISDN and switch-56 dial-up lines and for WAN Extender T1 and E1 permanent leased channelized connections.

WAN Extender virtual paths are unbound to a port until a connection is established. While they are unbound, the number of virtual paths that are not configured to be used for channelized leased lines can serve as dynamic paths in a dial-up path pool. The paths in the dial-up path pool are used for calls going through a port running PPP.

On ISDN or switch-56 dial-up lines, a virtual path binds to a port when an outgoing call is started or when an incoming call is received by the port. The virtual path goes back into the dial pool after the call is ended.

Like other dynamic paths, specific virtual paths in the dial-up pool can be dynamically bound to a port for bandwidth-on-demand or disaster recovery. After the demand and recovery is completed, the virtual paths unbind and return to the dial pool.

For channelized connections, such as T1 and E1, the virtual path binds to the port when the NETBuilder bridge/router and the WAN Extender synchronize with each other and the PPP negotiation is completed. The virtual paths used by channelized connections do not increase the number of paths in the dial pool after the call is ended.

For more information about WAN Extender virtual paths, refer to Chapter 36.

Virtual Ports

You can configure multiple ports over one path on the platforms listed in Table 1-1 on page 1-4. To configure multiple ports, you create new logical interfaces called *virtual ports*. A virtual port is an object you define through software, and associate with a nonvirtual port, called the parent port (see Figure 1-1). A virtual port functions in the same way as a port, that is, as a logical interface that represents a connection to a network. The virtual port and its parent port share most of their properties, but can be referenced separately by port-oriented software features such as route policy and packet filtering, and can also be distinguished by distinct wide area addresses.

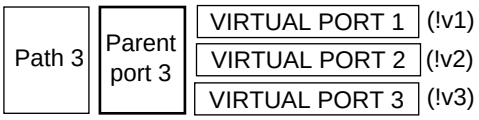


Figure 1-1 Parent Port and Virtual Port

For more information on virtual port numbering, refer to "Port and Path Numbering on a NETBuilder II Bridge/Router" on page 1-11 and to "Port and Path Numbering on a SuperStack II Bridge/Router" on page 1-14.

A virtual port can be connected to a network through a path providing a Frame Relay, ATM, or X.25 virtual circuit, or an SMDS Subscriber Network Interface (SNI). A connection can also be made using PPP with dial-up features to achieve multidestination dialing (modem pooling and WAN Extender virtual path pooling).

The sample Boundary Routing® topology in Figure 1-2 demonstrates the use of virtual ports. This topology shows a NETBuilder II bridge/router with two paths labeled path 1 and path 2. Path 1 is an Ethernet interface. Path 2 is connected to a Frame Relay network that interconnects multiple local area networks

through two SuperStack II boundary routers. Two virtual ports have been created on path 2. Each virtual port is a logical interface that represents a connection to one of the remote local area networks.

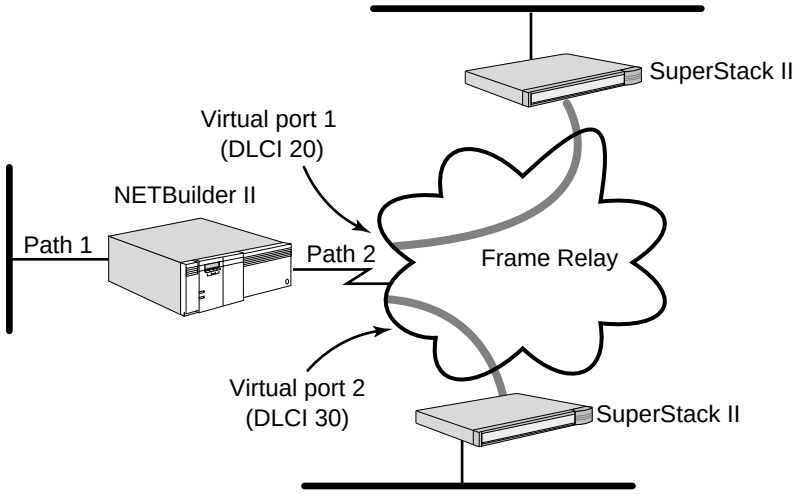


Figure 1-2 Topology Using Virtual Ports

Table 1-1 lists the bridge/routers that support virtual ports and the maximum number of virtual ports that can be configured on each bridge/router. There is no per-path limit, except that the total number of virtual ports configured on all paths cannot exceed the maximum for the bridge/router.

Table 1-1 Bridge/Routers That Support Virtual Ports

Bridge/Router	Number of Virtual Ports
NETBuilder II	40 or 75*
SuperStack II models 221, 222, 223, 224, 228, 323, 421, 422, 423, 424, and 523	28
SuperStack II models 227, 327, 427, 527	28†

* Software packages SW/NBII-CP and SW/NBII-FF support 75 virtual ports.
† These platforms can act as a central node in a Boundary Routing topology.

Virtual ports function in the same way as nonvirtual ports. Table 1-2 provides information on topologies that require virtual ports and the node in the topology on which the virtual ports should be created.

Table 1-2 Topologies that Require Virtual Ports

Topology	Virtual Ports Required?	Node That Virtual Ports Should Be Created On
Boundary Routing over Frame Relay, ATM DXI, or X.25	Yes.	Central node (NETBuilder II, SuperStack II models 227, 327, 427, or 527)
Traditional routed environment: partially meshed or nonmeshed Frame Relay, ATM DXI, and X.25 topologies	Depends on bridging or routing protocol. Refer to "Virtual Ports over Frame Relay, ATM DXI, and X.25" for more information.	"Hub" router (NETBuilder II or SuperStack II models 222, 224, 227, 228, 327, 422, 424, 427, or 527)
Traditional routed or bridged environment: fully, partially, and nonmeshed ATM topologies	Yes.	NETBuilder II nodes on both ends of serial line running ATM

(continued)

Table 1-2 Topologies that Require Virtual Ports (continued)

Topology	Virtual Ports Required?	Node That Virtual Ports Should Be Created On
SMDS Service where there are more than 127 routers or more than one logical network segment (or 32 segments under IP), or a need to selectively filter packets among groups	Yes.	Depends on configuration
Multidestination dialing (modem pooling) over PPP	Yes, for dynamic dial-up lines.	Central node (NETBuilder II or SuperStack II models 227,327, 427, or 527)
Multidestination dialing (WAN Extender virtual path pooling) over PPP	Yes, for dynamic dial-up lines.	Central node only (NETBuilder II)
Frame Relay topology with disaster recovery configured	Yes.	Nodes on both ends of serial line running Frame Relay

For more information on partially meshed and nonmeshed Frame Relay, SMDS, and X.25, and ATM topologies, refer to Chapter 42, Chapter 44, Chapter 45, and Chapter 47, respectively. Frame Relay topologies also apply to Asynchronous Transfer Mode Data Exchange Interface (ATM DXI).

Virtual Ports over Frame Relay, ATM DXI, and X.25

Frame Relay, ATM DXI, and X.25 are peer-to-peer protocols that connect two nodes on the network. Boundary Routing and bridging, Internet Protocol-Open Shortest Path First (IP-OSPF), DECnet IV, VINES, and Xerox Network Systems (XNS) require virtual ports because they do not provide a method for dealing with Frame Relay, ATM DXI, or X.25 topologies where bridge/routers are not directly connected to all others (full mesh). With Boundary Routing system architecture, when you create a virtual port over a particular path, each remote network attached to the Frame Relay, ATM DXI, or X.25 cloud is treated as a separate network.

Internet Protocol-Routing Information Protocol (IP-RIP), IP-Integrated Intermediate System-to-Intermediate System (IIS-IS) (NETBuilder II bridge/router only), Internetwork Packet Exchange (IPX), Intermediate System-to-Intermediate System (IS-IS), DECnet V, and AppleTalk can operate over partially meshed or nonmeshed Frame Relay, ATM DXI, or X.25 topologies without the use of virtual ports. The next-hop split horizon feature in IP-RIP, IPX, and AppleTalk allows communication between bridge/routers that are not directly connected to one another. To configure next-hop split horizon for these routing protocols, you must have a list of neighbors, which can be dynamically generated or manually configured in IP-RIP.

In IPX, you must manually configure neighbors for broadcast multi-access (BMA) networks. For nonbroadcast multi-access (NBMA) networks, for example, X.25 and Frame Relay, you can configure dynamic neighbor learning through the CONTrol parameter in the NRIP, SAP, and NLSP Services.

In AppleTalk, next-hop split horizon is configured by adding static mappings to the address mapping table.

You do not need to further configure IP-Integrated IS-IS and IS-IS to run over partially meshed or nonmeshed Frame Relay, ATM DXI, or X.25 topologies; you only need to configure neighbors.

Although it is not necessary to define virtual ports on IP-RIP, IPX, or AppleTalk routers in partially meshed or nonmeshed Frame Relay, ATM DXI, or X.25 topologies, virtual ports do provide the following additional benefits:

- A virtual port can be defined for each configured neighbor, allowing you to set up such features as filters and routing policies on a per-neighbor basis.
- Virtual ports provide greater control over your network.

If you want your NETBuilder II bridge/router or SuperStack II bridge/router to act as an Open System Interconnection (OSI) router in a Frame Relay, ATM DXI, or X.25 topology, you do not need to create virtual ports.

Table 1-3 summarizes each bridging and routing protocol and the technique you must use to deal with the lack of connectivity in partially meshed and nonmeshed Frame Relay, ATM DXI, and X.25 topologies.

Table 1-3 Connectivity in Partially Meshed and Nonmeshed Topologies

Protocol	Technique
Bridging	Virtual port
Boundary Routing	Virtual port
IP-RIP*	Next-hop split horizon
IP-OSPF	Virtual port
IP-Integrated IS-IS*	No special configuration required
IS-IS	No special configuration required
IPX*	Next-hop split horizon
APPN*†	No special configuration if sending APPN only over Frame Relay
DECnet IV	Virtual port
OSI/DECnet V	No special configuration required
VINES	Virtual port
XNS	Virtual port
AppleTalk*	Next-hop split horizon

* When configuring this protocol and another protocol that requires virtual ports over the same path, use virtual ports.
† The SuperStack II bridge/router does not support this protocol.

Virtual Ports over ATM

In an ATM environment, virtual ports are required in fully meshed and partially meshed topologies when bridging and routing. Nonmeshed topologies are supported but are not recommended. Each ATM virtual port has a unique media access control (MAC) address.

For ATM configuration information, refer to Chapter 47.

Virtual Ports over PPP

You can use virtual ports and WAN Extender virtual paths in a PPP environment to provide dial pooling at the central site router. With dial pooling, a set of dynamic paths is unbound from their default ports and waits in the dial pool for an incoming call. When a call is received, the dynamic path that answers is assigned to a virtual port, which is standing by with the appropriate configuration information for the calling network. Because not all sites using a

dial pool call the central site at the same time, it is possible to share a small group of paths with a larger group of sites. Each site that can potentially call into the dial pool has its own virtual port defined, so there are usually more virtual ports configured for the dial pool than dynamic paths assigned to it.

PPP virtual ports differ from Frame Relay, ATM, X.25, and SMDS virtual ports in the following ways:

- A PPP virtual port can potentially use any path in the dial pool.

  Frame Relay, ATM, X.25, and SMDS virtual ports are always associated with a particular path.

- PPP virtual ports do not have a parent port and operate independently. No parent port exists because the path was unbound from its port and placed into the dynamic dial path pool.

  Frame Relay, ATM, X.25, and SMDS virtual ports inherit the attributes of the path over which they are defined. For more information, refer to "Parent Ports."

- PPP virtual ports can be used with dial-up related parameters.

  Frame Relay, ATM, X.25, and SMDS virtual ports cannot be used with dial-up related parameters.

Virtual Ports over SMDS

Unlike Frame Relay, ATM, and X.25, SMDS provides a connectionless wide area network that also has multicast delivery capability, giving it LAN-like characteristics. Each attachment point to the SMDS network, the Subscriber Network Interface (SNI), can be assigned up to 16 individual addresses by the SMDS service provider. These addresses can be used to distinguish up to 16 distinct virtual SMDS ports over the same SNI. Unlike virtual ports for Frame Relay, ATM, or X.25, which connect to a single remote device, each virtual port in an SMDS environment connects to a distinct group of fully meshed devices. This allows the creation of a hierarchical, partially meshed structure that can exceed the SMDS address-screen-imposed limitation of 128 addresses in an SMDS network.

SMDS virtual ports provide additional points of control for configuring network and routing protocols, and for selectively applying port-level features such as filtering, route policy control, and route aggregation. Boundary Routing is not supported over SMDS.

For more information, refer to Chapter 44, in the "SMDS Addresses" section on page 44-19.

Parent Ports

When you configure an X.25, Frame Relay, ATM, or SMDS virtual port, it inherits the attributes of the path over which it is defined. It also inherits some of the attributes of the port associated with the path through which the virtual port is defined. This port is referred to as the *parent port*.

For PPP dial virtual ports, no parent port exists because the path was unbound from its port and placed into the dynamic dial path pool.

Unlike Frame Relay, ATM, X.25, and SMDS virtual ports, which are always associated with a particular path, PPP virtual ports can potentially use any path in the dynamic dial path pool. PPP virtual ports also can be used with dial-up-related parameters.

For example, if you create a Frame Relay, ATM, X.25, or SMDS virtual port associated with a wide area port, the virtual port inherits port attributes from the following sources:

- Default and configured values of PORT Service parameters specified for a wide area port, with the exception of the following PORT Service parameters that are not related to X.25, Frame Relay, ATM, and SMDS virtual ports:

AutoDial	DialRetryCount
COMPressType	DialRetryTime
DialCONFig	DialSamplPeriod
DialCONTrol	DialSTatus
DialDebouncTime	LinkCompStat
DialHistory	OWNer
DialIdleTime	PAths
DialInitState	PathPreference
DialRcvrState	

  The parameters in this list do apply to PPP virtual ports, such as SysCallerID virtual ports.

- Default and configured values of parameters from all other services specified for a wide area port.

To configure a virtual port, you must specify the virtual port and not the parent port. For example, if you are using the SETDefault !<port> -BCN CONTrol = Enabled syntax, you must specify the virtual port number instead of the parent port number for <port>. For complete information on the numbering convention of virtual ports, refer to "Port and Path Numbering on a NETBuilder II Bridge/Router" on page 1-11 or to "Port and Path Numbering on a SuperStack II Bridge/Router" on page 1-14.

Logical Networks

On the NETBuilder II bridge/router, the *multiple logical network* (MLN) feature allows you to:

- Group together multiple ports on a single bridge/router, and the LAN segments attached to them, to form a logical network. NETBuilder software can use groups in its network topology in the same way it uses virtual ports.
- Bridge network protocols, such as IP, among ports within a group.
- Route network protocols outside the group to other ports, virtual ports, or logical networks.
- Configure different MLN configurations for different protocols.
- Maintain configurations for protocols not configured for MLN, so that they bridge and route as usual, independent of the port groupings for other protocols.

Unlike a conventional bridge/router, MLN provides simultaneous bridging and routing for the same network protocol. It enables you to integrate a number of

bridged networks by routing from the bridged environments (configured as logical networks) across a LAN or WAN backbone. It also allows you to assign the same network number or subnet number to multiple physical paths. You can think of the logical network as a group of LAN segments that have been joined together to form a single network-level addressing domain.

When a conventional bridge/router is configured to bridge a particular protocol, all traffic for that protocol is bridged, and the router component is inactive, as shown in Figure 1-3. When it is configured to route that protocol, correctly addressed traffic for the protocol is routed, and the bridge component is inactive, as shown in Figure 1-4.

Bridging can occur even when the bridge/router is configured as a router. If a bridge/router receives packets of a protocol type that has not been configured on it, the bridge/router bridges the packets. If the -BRidge CONtrol parameter has been set to NoFireWall, incorrectly addressed routed packets are also bridged. The bridge/router can also be configured to bridge some protocols and route others. However, a conventional bridge/router without MLN cannot selectively bridge or route the same protocol, depending on destination.

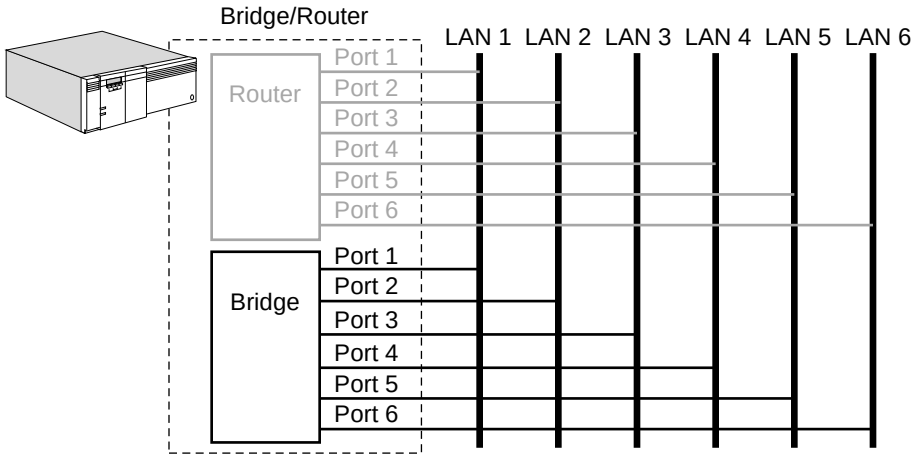


Figure 1-3 Bridge/Router in Bridging Mode

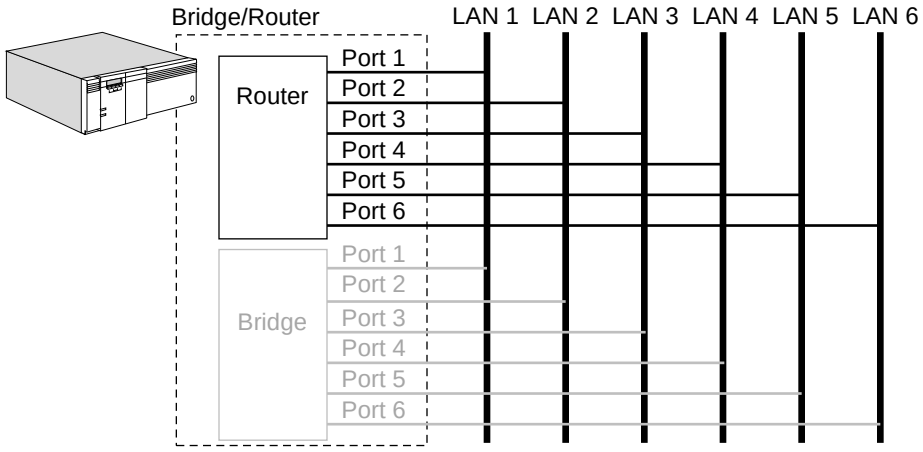


Figure 1-4 Bridge/Router in Routing Mode

Figure 1-5 is an example of the simultaneous bridging and routing capability provided by MLN. Six networks are attached to a NETBuilder bridge/router. Each of the six networks has IP nodes, IPX nodes, and AppleTalk nodes. Ports 1, 2, and 3, and the LANs attached to them, have been grouped together into one logical network or port group, called V1. The logical interface between NETBuilder software and this group is called a *group port*, and it is also identified as V1. The IP protocol has been configured on group port V1 (that is, V1 has been given an IP address). This IP address also applies to all ports in the group.

Group ports are numbered as if they were virtual ports.

Ports 5 and 6, and the LANs attached to them, have been grouped into another logical network, V2. IP has also been configured on this group. IP has been configured individually on port 4, which has not been assigned to a group (that is, port 4 has been given an IP address).

Port groups have not been defined for IPX and AppleTalk. The bridge/router has been configured to route IPX. It has not been configured to route AppleTalk.

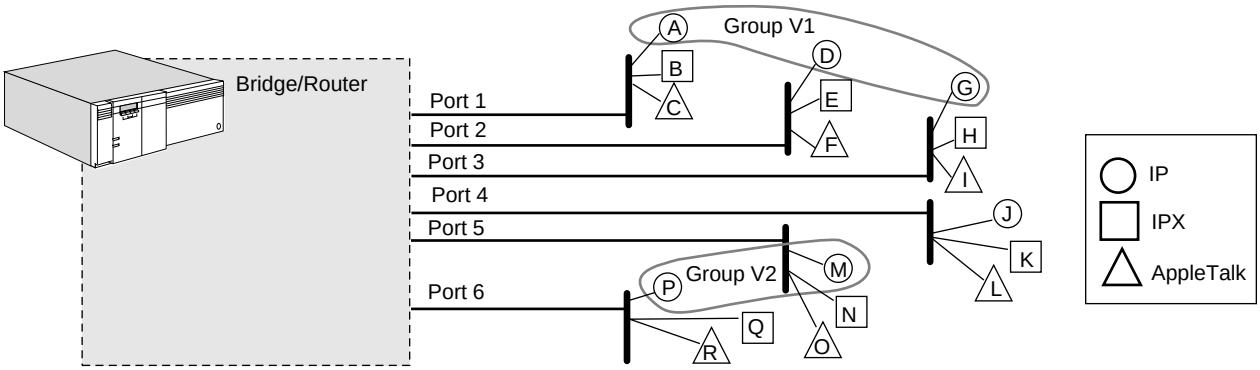


Figure 1-5 Multiple Logical Networks

Figure 1-6 shows how IP traffic is handled in this configuration. IP is bridged among ports 1, 2, and 3 (as indicated in the figure by the MLN Bridge, which is not a physical bridge but an internal software function). IP traffic is also bridged between ports 5 and 6. IP is routed between group V1 and all ports outside the group, including port 4 and group port V2. IP is also routed between group V2 and all ports outside the group, including port 4 and group port V1.

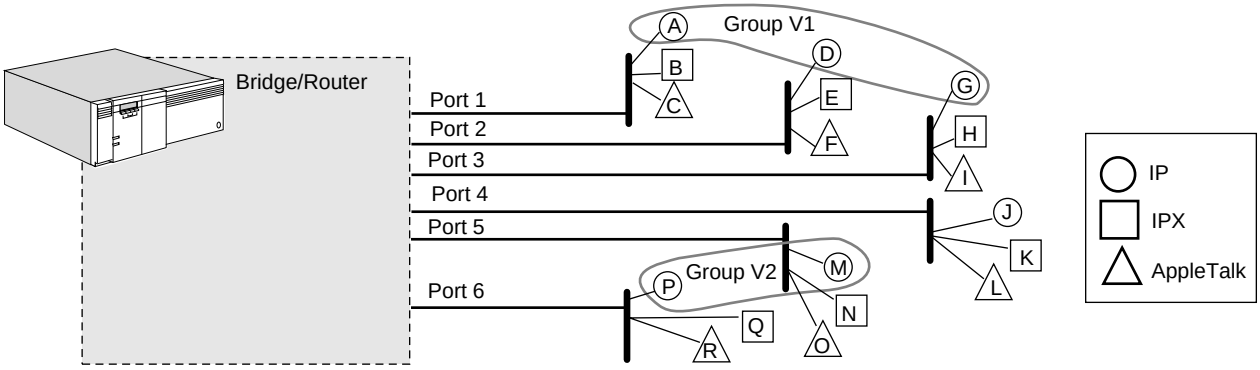


Figure 1-6 IP Configuration Under MLN

Figure 1-7 shows how the network looks to IPX. IPX traffic is routed among all ports, independent of the port groups defined for IP.

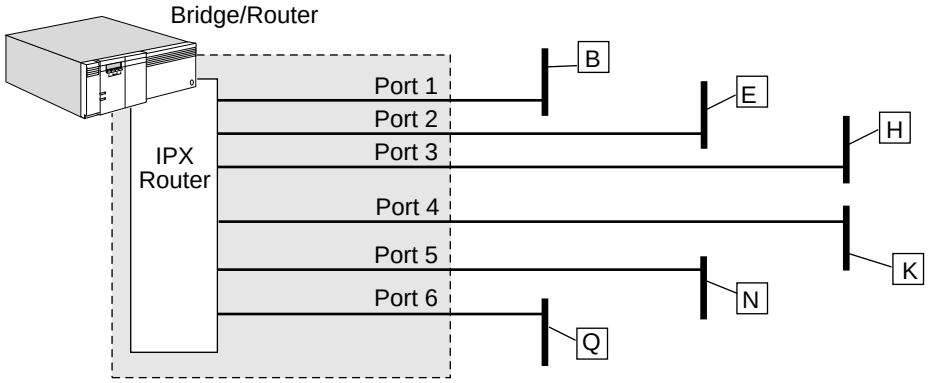


Figure 1-7 IPX Configuration Under MLN

AppleTalk routing is not enabled, so AppleTalk traffic is bridged among all six ports, as shown in Figure 1-8.

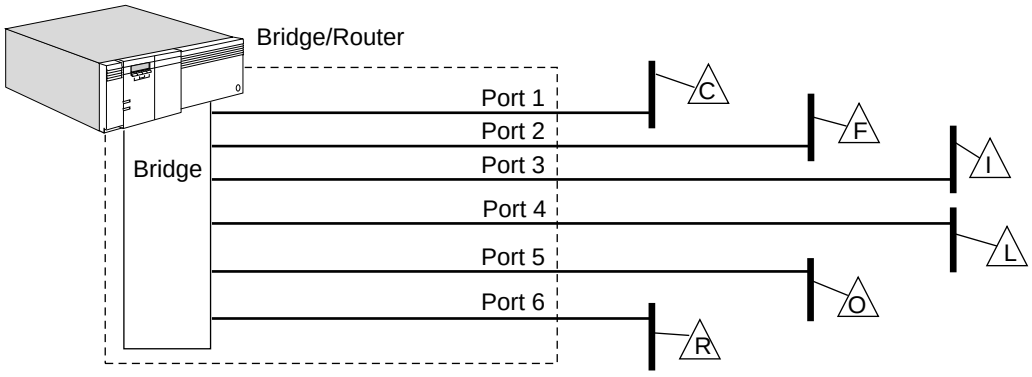


Figure 1-8 AppleTalk Configuration Under MLN

Only network protocols that configure a port group are affected by MLN. A protocol that does not participate in MLN can continue to configure its network topology at the port and virtual port level, including ports that belong to a port group for some other protocol. Bridged protocols such as NetBIOS and Logical Link Control, type 2 (LLC2) are also not affected by MLN.

MLN does not bridge between port groups, between a port group and a port, or between a port group and a virtual port. All of this type traffic is routed.

Software version 8.3 and later support MLN for IP routing and transparent bridging over Ethernet. To configure logical networks, refer to "Configuring Multiple Logical Networks" on page 1-22.

Port and Path Numbering on a NETBuilder II Bridge/Router

The configuration of ports and paths on your NETBuilder II bridge/router depends on the combination of I/O modules installed. For information on acceptable I/O module configurations for 4-, 8-, and 8-slot extended chassis NETBuilder II bridge/routers, refer to the chassis guide that came with your NETBuilder II bridge/router. Regardless of the number of slots, ports and virtual ports are numbered in the same way.

Figure 1-9 shows a possible port and path configuration on a NETBuilder II bridge/router connecting Ethernet, FDDI, and wide area networks. In this figure, the bridge/router labeled A uses four ports, three virtual ports, and six paths.

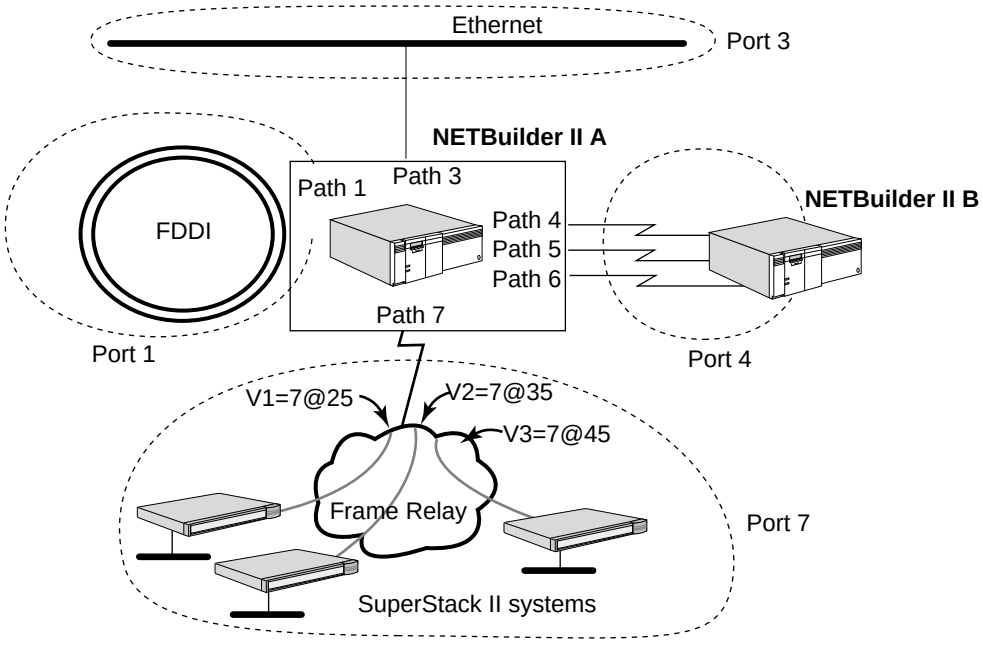


Figure 1-9 Ports, Virtual Ports, and Paths on a NETBuilder II Bridge/Router

Path 1, an FDDI network, is assigned to port 1. Path 3, an Ethernet network, is assigned to port 3. Path 7, a wide area network attached by a Frame Relay cloud, is assigned to port 7. Paths 4, 5, and 6, a wide area network, are assigned to port 4. These multiple paths are assigned to the port by entering:

```
ADD !4 -PORT PAths 4, 5, 6
```

This example does not include path 2, because the FDDI module set requires two consecutive slots in the NETBuilder II bridge/router.

Virtual ports are configured on the wide area network attached by a Frame Relay cloud. Virtual ports are numbered V*n*, where *n* is a number from 1 through the maximum supported on the bridge/router (refer to Table 1-1). In Figure 1-9, the virtual ports V1, V2, and V3 are created by entering:

```
ADD !V1 -PORT VirtualPort 7@25
ADD !V2 -PORT VirtualPort 7@35
ADD !V3 -PORT VirtualPort 7@45
```

You do not need to create virtual ports in numerical order. For instance, you can create virtual port V2 before V1.

If you are using an Ethernet 2-Port 10BASE-FL module, HSS V.35 3-Port module, or Ethernet 6-Port 10BASE-T module in your NETBuilder II bridge/router, additional port syntax for those ports is required. For more information, refer to "Port and Path Numbering on NETBuilder II Multiport Modules."

Port and Path Numbering on NETBuilder II Multiport Modules

Most I/O modules used on the NETBuilder II bridge/router have only one physical connector, with the following exceptions:

- The Ethernet 2-Port 10BASE-FL module has two physical connectors on each board.
- An external adapter cable maps the single high-density connector on the HSS V.35 3-Port module into three separate V.35 standard interface connectors.
- The Ethernet 6-Port 10BASE-T module has six connectors on each board.

Ports and paths on these multiport modules are labeled differently from ports and paths on other modules. To differentiate one physical path or one logical port from another, you append a letter to the path or port number.

For the Ethernet 2-Port 10BASE-FL module (Figure 1-10), this letter is an upper- or lowercase A for the first interface and an upper- or lowercase B for the second interface. The A designation, which is optional, corresponds to the physical connector on the left side of the module (Interface A), while the B designation corresponds to the connector on the right side (Interface B).

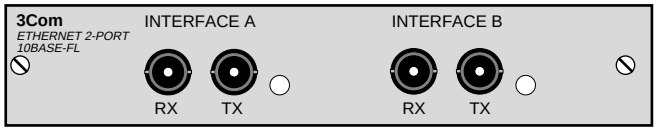


Figure 1-10 Ethernet 2-Port 10BASE-FL Module

For the HSS V.35 3-Port module, this letter is an optional upper- or lowercase A for the first interface, an upper- or lowercase B for the second interface, and an upper- or lowercase C for the third interface. These designations correspond to the markings on the V.35 adapter cable, as shown in Figure 1-11.

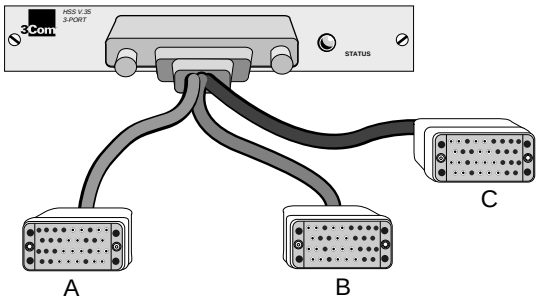


Figure 1-11 HSS V.35 3-Port Module

The Ethernet 6-Port 10BASE-T module connectors are designated A through F as shown in Figure 1-12. The A designation is optional.

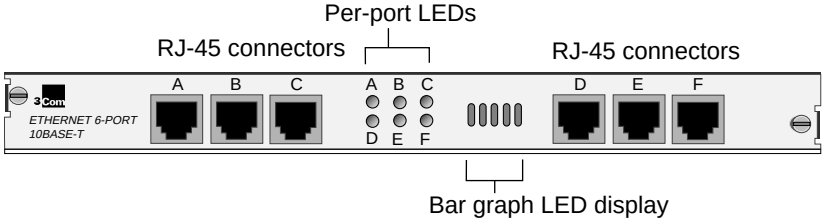


Figure 1-12 Ethernet 6-Port 10BASE-T Module

To configure port or path settings on a multiport module, you must identify both the slot and the letter. For example, to configure settings for the right-hand physical connector of the Ethernet 2-Port 10BASE-FL module in slot 4 of the NETBuilder II hardware, use the path 4B. To name this path, enter:

```
SETDefault !4B -PATH NAme = "sjose"
```

To configure settings for the left physical connector, you can use either 4 or 4A. For example, to name path 4A, enter one of the following commands:

```
SETDefault !4 -PATH NAme = "sfran"
SETDefault !4A -PATH NAme = "sfran"
```

For more information regarding multiport modules, refer to the *NETBuilder II Ethernet 2-Port 10BASE-FL Module Installation Guide*, the *NETBuilder II V.35 3-Port Module Installation Guide,* and *the NETBuilder II MP Ethernet 6-Port 10BASE-T Module Installation Guide*.

Port and Path Numbering on a SuperStack II Bridge/Router

Table 1-4, Table 1-5, and Table 1-6 outline the default port and path numbering for the model 22x, 42x, 32x, and 52x SuperStack II bridge/routers.

Table 1-4 Path and Port Numbering for Model 2xx Bridge/Routers

Path No.	Connector Mapped To	Port No. Mapped To
1	10BASE-T or AUI (Depends on which connector is cabled.)	1
2	V.35	2
3	RS-449	3
4	RS-232	4

Table 1-5 Path and Port Numbering for Model 42x Bridge/Routers

Path No.	Connector Mapped To*	Port No. Mapped To
1	10BASE-T or AUI (Depends on which connector is cabled.)	1
2.1	ISDN	2
2.2	ISDN	3
3	V.36/RS-449 or RS-232 (Depends on which connector is cabled. Use only one of these connectors at a time.)	4

* The connector and port associated with paths 2.1, 2.2, and 3 cannot be reconfigured.

Table 1-6 Path and Port Numbering for Model 32x Bridge/Routers

Path No.	Connector Mapped To	Connector Marking	Port No. Mapped To
1	UTP or STP (Depends on which connector is cabled.)	UTP or STP	1
2	V.35	A	2
3	Universal serial connector (USC)*	B	3
4	RS-232	C	4

* This connector can be converted to an X.21, V.35, V.36, RS-449, or RS-232 connector using cables. For more information, refer to your SuperStack II bridge/router or boundary router installation guide.

Table 1-7 Path and Port Numbering for Model 52x Bridge/Routers

Path No.	Connector Mapped To	Connector Marking	Port No. Mapped To*
1	UTP or STP (Depends on which connector is cabled)	UTP or STP	1
2.1	ISDN	ISDN	2
2.2	ISDN	ISDN	3
3	USC†	B	4
4	RS-232	C	5

* The connector and port associated with paths 2.1, 2.2, 3, and 4 cannot be reconfigured.

† This connector can be converted to an X.21, V.35, V.36, RS-449, or RS-232 connector using cables. For more information, refer to your SuperStack II bridge/router or boundary router installation guide.

In an ISDN environment, the path numbering convention differs from the convention in a non-ISDN environment. Instead of numbering only the physical interface, such as path 2, ISDN sometimes requires that you number the physical interface or connector and the multiple channels that transmit data (path 2.*n*). Valid channel numbers for model 42x and 52x bridge/routers are 1 and 2 (for example, paths 2.1 and 2.2). The physical interface number and channel number are separated by a decimal point.

Some parameters that support ISDN require that you specify a path consisting of a connector number only, while others require that you specify a path consisting of both a connector number and a channel number. If you do not specify a channel number for a parameter that requires it, the parameter is configured for channel 1 only. If you want to specify all channels associated with a physical interface, specify the connector number and an asterisk (for example, 2.*).

If you are unsure how to specify a path, refer to the description of the parameter in *Reference for NETBuilder Family Software*.

Before configuring ports, virtual ports, and paths for the ISDN interface on your SuperStack II bridge/router platform, you must decide how you want to use your ISDN interface. For more information, refer to Chapter 35.

You can reconfigure the software so that multiple paths are mapped to one wide area port by entering the ADD -PORT PAths command. If you assign multiple paths to a wide area port, the port must be running PPP or PLG.

If you have a WAN Extender attached to a NETBuilder II bridge/router, you can bind multiple WAN Extender virtual paths from the dial pool to a wide area port if the port is running PPP and Multilink Protocol.

Multiple paths can be mapped to one port to take advantage of the disaster recovery, bandwidth-on-demand features, and dial path pools if you are using ISDN or switch-56 circuit services directly or through a WAN Extender.

Figure 1-13 shows two sample topologies: the first topology has one path mapped to one port and the second topology has two paths mapped to one port (for disaster recovery or bandwidth-on-demand). In the first topology, path 2 is assigned by default to port 2. In the second topology, the software has been reconfigured so that paths 3 and 4 are mapped to port 3.

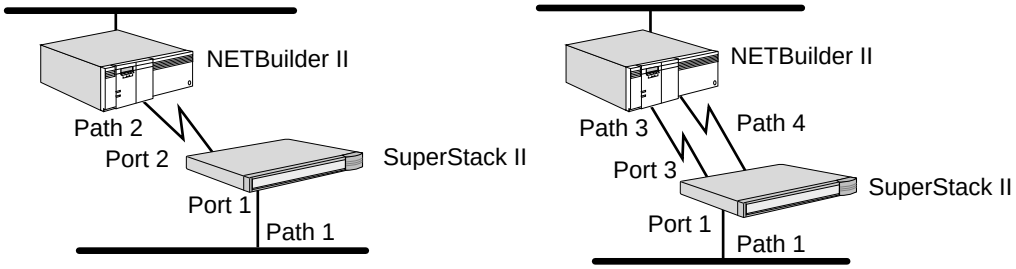


Figure 1-13 Possible Path-to-Port Assignments on a SuperStack II Bridge/Router

You can also assign one path to one wide area port and one path to another wide area port, as shown in Figure 1-14. In this figure, path 2 is assigned to port 2 and path 3 is assigned to port 3. Configure your wide area ports and paths as shown in this figure only if you plan to implement the Boundary Routing network resiliency feature on your SuperStack II bridge/router.

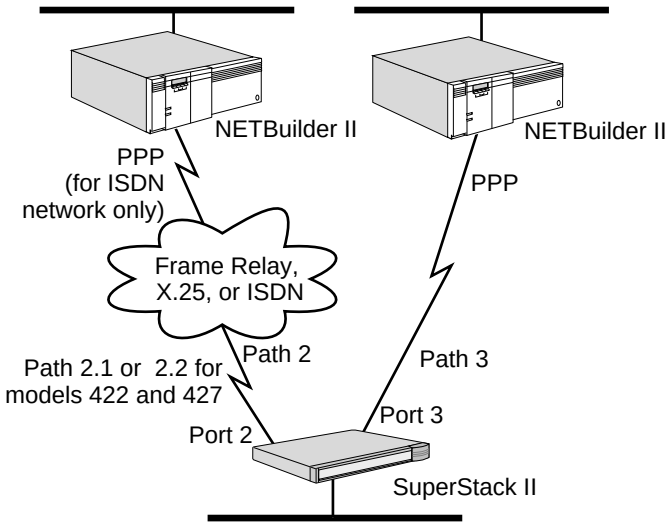


Figure 1-14 Setting Up Two Wide Area Ports on a SuperStack II Bridge/Router

Virtual ports are numbered V*n*, where *n* is a number from 1 through 28, which is the maximum supported on a SuperStack II bridge/router.

Figure 1-15 shows a sample topology using virtual ports. In this example, virtual ports are configured on the wide area ports of two SuperStack II bridge/routers on both sides of an X.25 network. In the figure, each bridge/router has a virtual port defined over the path directly connected to the X.25 network. On SuperStack II bridge/router #1, virtual port V1 is defined over path 2. On SuperStack II bridge/router #2, virtual port V2 is defined over path 3.

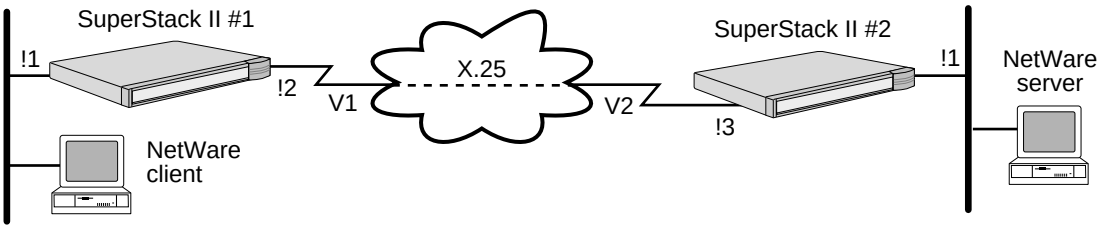


Figure 1-15 Virtual Ports on a SuperStack II Bridge/Router

Configuring Local Area Interfaces Only

To set up ports and paths on a bridge/router with local area interfaces, follow these steps:

1 Assign a name to path 1 (optional).

For example, to name path 1, enter:

```
SETDefault !1 -PATH NAme = "Floor_1"
```

Some restrictions apply to the name you assign using the -PATH NAme parameter. For more information, refer to Chapter 42 in *Reference for NETBuilder Family Software*.

2 If you have a model 32x or 52x SuperStack II bridge/router, configure the ring speed for the path using:

```
SETDefault !<port> -PATH BAud = 4000 | 16000
```

3 If necessary, enable the path.

Paths are enabled by default. If the path was previously disabled or you reconfigured the -PATH BAud parameter in the previous step, you must re-enable it.

For example, to enable path 1, enter:

```
SETDefault !1 -PATH CONTrol = Enabled
```

4 Assign a name to port 1 (optional).

Use a name that is easy to remember. For example, if port 1 is in Building 1, enter:

```
SETDefault !1 -PORT NAme = "Bldg_1"
```

Some restrictions apply to the name you assign using the -PORT NAme parameter. For more information, refer to Chapter 43 in *Reference for NETBuilder Family Software*.

5 If necessary, enable or disable the port.

All ports are enabled by default. If the port has been disabled, you must re-enable it.

3Com recommends that you disable any port you do not use. Disabling unused ports improves bridge/router performance.

To enable or disable the port, use:

```
SETDefault !<port> -PORT CONTrol = Enabled | Disabled
```

6 Repeat steps 1 through 5 for each local area port and local area path on your bridge/router.

This completes the setup procedure for local area ports and paths. To set up wide area ports and paths, refer to the next section. To configure bridging or routing protocols, refer to the bridging and routing chapters in this guide.

Configuring Wide Area Interfaces

This section describes the procedure for setting up ports and paths on a bridge/router with wide area interfaces. To set up ports and paths for local area interfaces, follow the procedure described in "Configuring Local Area Interfaces Only" on page 1-17.

Before configuring ports and paths for the ISDN interface on SuperStack II bridge/routers with ISDN interfaces, you must decide how you want to use your ISDN interface. For more information, refer to Chapter 35.

To set up ports and paths on a bridge/router with wide area interfaces, follow these steps:

1 Assign a name to the path (optional).

For example, to assign the path name SF-SJ, enter:

```
SETDefault !3 -PATH NAme = "SF-SJ"
```

Some restrictions apply to the name you assign. For more information, refer to Chapter 42 in *Reference for NETBuilder Family Software*.

2 If necessary, reconfigure the connector type for the path.

This step applies only to a NETBuilder II bridge/router with an high-speed serial (HSS) adapter card and to model 327 and 527 SuperStack II bridge/routers if you converted the serial connector marked "B" (also referred to as the universal serial connector (USC)) to X.21, V.35, V.36, or RS-232 using a cable.

Table 1-8 summarizes which connector type to select if you have converted the serial connector marked "B" on your model 327 and 527 SuperStack II bridge/routers.

Table 1-8 Connector Setting for Converted Connectors on Model 327 and 527

Connector Type Converted To	Setting of -PATH CONNector Parameter
X.21	X21
V.35	V35
V.36 or RS-449	RS449
RS-232	RS232

On a NETBuilder SuperStack II bridge/router with an RS-449 module installed, the software cannot distinguish between the RS-449 module and an HSS V.35 module. You must configure the -PATH CONNector parameter to RS-449; otherwise, the software assumes the module is a V.35.

For the model 42x SuperStack II bridge/router, 3Com recommends retaining the default setting of the -PATH CONNector parameter (Auto). When this parameter is set to Auto, detection of the DTE connector type takes place when the platform boots.

For more information on the CONNector parameter and the auto startup feature, refer to Chapter 42 in *Reference for NETBuilder Family Software*.

3 If necessary, reconfigure the transmit clock setting for the serial path.

The default setting is External. Reconfigure this setting using:

```
SETDefault !<path> -PATH CLock = TestMode | Internal
```

The TestMode value applies to all NETBuilder platforms except model 327 and 527 SuperStack II bridge/routers; the Internal value applies to model 327 and 527 SuperStack II bridge/routers only.

If you are configuring a NETBuilder II bridge/router with a WAN Extender, leave the transmit clock setting at External, the default.

You do not need to perform this step for the ISDN path for model 42x and 527 SuperStack II bridge/routers.

The bridge/router usually derives its clock from an external modem so you do not need to change the External default setting. If you have a SuperStack II bridge/router model 327 or 527, and connected a serial connector to an IBM cluster controller, specify the Internal value. If you have any of the other NETBuilder platforms, and you want it to derive the clock from the on-board oscillator, specify the TestMode value.

If you connect two NETBuilder II or SuperStack II bridge/routers to a NETBuilder II bridge/router with an HSS V.35 3-Port WAN interface, you must use a modem eliminator and set the CLock parameter to External on both devices. Contact your 3Com supplier for a suggested list of modem eliminators.

4 If necessary, reset the baud rate for the path.

The default baud rate setting is 64 kbps. For example, to set the baud rate of path 3 at 256 kbps, enter:

```
SETDefault !3 -PATH BAud = 256
```

For the range of baud rates available for each of the bridge/router platforms, refer to Chapter 42 in *Reference for NETBuilder Family Software*.

It is important to set the baud rate even if you use an external clock. The bridge/router uses the baud rate setting to allocate resources for a path, compute metrics, select a forwarding path, and so forth.

5 If necessary, assign a path or multiple paths to each port.

For example, to assign paths 3 and 4 to port 3, enter:

```
ADD !3 -PORT PAths 3,4
```

Assigning multiple paths to a port is supported only when PPP or PLG is the port owner.

To receive incoming calls from a remote site, you also can assign a dial path pool, WAN Extender dial-up lines, or WAN Extender channelized virtual paths to a port.

For example, the following command assigns all incoming calls from Boston using dial-up lines, WAN Extender dial-up lines, or channelized virtual paths to port 2:

```
ADD !2 -PORT PAths SCID “Boston”
```

For more information about setting up dial-up lines, refer to Chapter 37.

When you assign multiple paths to a port, a load-sharing algorithm is enabled. For more information, refer to “Load Sharing” on page 3-32.

6 If you have previously disabled the path, changed the value of the BAud, CLock, or CONNector parameters, or assigned multiple paths to one port, you must re-enable the path.

For example, to re-enable path 3 on your bridge/router, enter:

```
SETDefault !3 -PATH CONTrol = Enabled
```

If multiple paths are assigned to a port, you should enable all paths assigned to the port so the load-sharing algorithm takes effect.

7 Repeat steps 1 through 6 for each wide area path you configure.

8 If necessary, enable or disable the wide area port.

All ports are enabled by default. If the port was previously disabled, you must re-enable it.

3Com recommends that you disable any port you do not use. Disabling unused ports improves bridge/router performance.

To enable or disable the port, use:

```
SETDefault !<port> -PORT CONTrol = Enabled | Disabled
```

9 Assign a name to the port (optional).

For example, to assign wide area port 3 the name SanJose, enter:

```
SETDefault !3 -PORT NAme = "SanJose"
```

Some restrictions apply to the name you assign. For more information, refer to Chapter 43 in *Reference for NETBuilder Family Software*.

10 Refer to Table 43-8 in *Reference for NETBuilder Family Software* to determine the default port owner for your wide area ports. If necessary, change the default owner of a wide area port using:

```
SETDefault !<port> -PORT OWNer = PPP | PLG | FrameRelay | SMDS |
 X25 | WanExtender | SDLC | ATM | Auto
```

By default, the auto startup feature on the NETBuilder II and SuperStack II bridge/routers can provide an automatic PPP or Frame Relay data link connection.

Auto startup does not provide an automatic PLG, SMDS, X25, WAN Extender, SDLC, or ATM data link connection. If the owner of the wide area port is one of these protocols, you need to manually set the value of this parameter to PLG, SMDS, X25, WanExtender, Synchronous Data Link Control (SDLC), or ATM as appropriate.

For complete information on the -PORT OWNer parameter and the auto startup feature, refer to Chapter 43 in *Reference for NETBuilder Family Software* and Chapter 33 in this guide, respectively.

For information on WAN Extender, refer to the *WAN Extender 2T/2E Installation Guide*; the *WAN Extender Manager User Guide*, and Chapter 36 in this guide.

11 Repeat steps 8 through 10 for each wide area port you configure.

This completes the configuration of ports and paths. The new settings take effect immediately. If you need to configure virtual ports, go to the next section. To configure bridging or routing protocols, refer to the bridging and routing chapters in this guide.

Configuring Virtual Ports

This section explains how to configure virtual ports on your bridge/router. See Table 1-1 on page 1-4 to determine whether your platform supports virtual ports.

Before setting up virtual ports for the ISDN interface on a SuperStack II bridge/router with an ISDN interface, you must decide how you want to use your ISDN interface. For more information, refer to Chapter 35.

Before configuring virtual ports, make sure that the owner of the wide area port associated with the path through which the virtual ports will be defined is set appropriately. For instructions, refer to "Configuring Wide Area Interfaces" on page 1-17.

To set up virtual ports, follow these steps:

1 Create a virtual port for each remote network that is attached to a Frame Relay, X.25, SMDS, ATM, or ISDN cloud, or that is running the PPP Protocol, using:

```
ADD !<port> -PORT VirtualPort {<path> {<FR_DLCI> | <X.25 DTE> |
 SMDS | MPATM | ETHATM} | SysCallerID"<callerid>"}
```

Virtual ports are numbered V*n*, where *n* is a number from 1 through the maximum supported on the bridge/router (refer to Table 1-1). You do not need to create virtual ports in numerical order. For instance, you can create virtual port V2 before V1.

For example, if you have a remote network on interface 1 that uses Frame Relay data link connection identifier (DLCI) 35, add virtual port V1 by entering:

```
ADD !V1 -PORT VirtualPort 1@35
```

ATM DXI ports also use the FR_DLCI value.

If you have a remote network on interface 3 that uses X.25 DTE 31107551234, add virtual port V3 by entering:

```
ADD !V3 -PORT VirtualPort 3#31107551234
```

If you have a remote network on interface 5 that uses SMDS, add virtual port V4 by entering:

```
ADD !V4 -PORT VirtualPort 5 SMDS
```

The command syntax for SMDS virtual ports does not use an individual DTE address. The virtual port does not take effect until its SMDSIndivAddr parameter has been configured.

If you have a remote network on interface 4 that uses multiprotocol encapsulation for ATM, add virtual port V5 by entering:

```
ADD !V5 -PORT VirtualPort 4 MPATM
```

If you have a remote network on interface 4 that uses LAN Emulation for ATM, add virtual port V5 by entering:

```
ADD !V5 -PORT VirtualPort 4 ETHATM
```

To create a PPP dial virtual port that uses the dynamic dial pool for its path resources in initiating and receiving calls from a remote router called NewYork, enter:

```
ADD !V3 -PORT VirtualPort SysCallerID"NewYork"
```

This command builds a mapping table entry between the virtual port and a remote bridge/router identifier (the SysCallerID value of the remote NETBuilder site) and allows an incoming call to be mapped to a specific port. The remote router can be another NETBuilder II bridge/router or a SuperStack II bridge/router with an ISDN interface. Additional configuration steps are required to use the dial pool. For more information, refer to Chapter 37.

You can create multiple PPP virtual ports, but only one virtual port on a dynamic path can be active at a time.

2 If necessary, re-enable the virtual port.

Virtual ports are enabled by default. If virtual port V3 has been disabled, re-enable it by entering:

```
SETDefault !V3 -PORT CONTrol = Enabled
```

3 Assign a name to the virtual port (optional).

For example, to assign virtual port V3 the name First_St, enter:

```
SETDefault !V3 -PORT NAme = "First_St"
```

Some restrictions apply to the name you assign. For more information, refer to Chapter 43 in *Reference for NETBuilder Family Software*.

4 Repeat steps 1 through 3 for each virtual port you configure.

This completes the configuration of virtual ports. The new settings take effect immediately. To configure bridging or routing protocols, refer to the bridging and routing chapters in this guide.

Configuring Multiple Logical Networks

This section describes how to set up MLNs by creating port groups and assigning ports to them. In software version 8.2 through version 9.1, you can create port groups only for Ethernet ports. Because version 8.2 through 9.1 supports MLN only for the IP protocol, create port groups only for ports over which you intend to route IP.

The interface between a port group and NETBuilder software is a single port called the *group port*, which represents all ports in the port group.

To create port groups, follow these steps:

1 To assign ports to a port group, use:

```
ADD !<port> -PORT LogicalNET ETHernet <port> [,…]
 ["<string>"] (1-50 characters]
```

where the first <port> is the group port that interfaces to the logical network. This port is always numbered as if it were a virtual port (V*n*). The ports that follow the ETHernet parameter are assigned to the port group. These ports are called member ports, they cannot be virtual ports.

The last argument, "<string>", which must be enclosed in quotation marks, is an optional descriptive name for the group port. It is displayed by entering the SHow -PORT LogicalNET CONFiguration command.

For example, the following command adds ports 1 and 2 to port group V1:

```
ADD !V1 -PORT LogicalNET ETHernet 1,2 "Test Network B200 4th
 floor"
```

If port group V1 does not already exist, it is created and ports 1 and 2 are added to it. V1 also identifies the group port that references the group.

The following command adds ports 3 and 4 to port group V2:

```
ADD !V2 -PORT LogicalNET ETHernet 3,4
```

Port groups cannot overlap, that is, the same port cannot be configured as part of two different port groups.

2 If necessary, enable the group port.

Group ports are enabled by default. If group port V2 has been disabled, re-enable with this command by entering:

```
SETDefault !V2 -PORT CONTrol = Enabled
```

3 Assign a name to the group port (optional).

For example, to assign group port V2 the name "Bayfront," enter:

```
SETDefault !V2 -PORT NAme = "Bayfront"
```

Some restrictions apply to the name you assign. For more information, refer to the -PORT NAme parameter in *Reference for NETBuilder Family Software*.

4 Repeat steps 1 through 3 for each group port you configure.

This completes the configuration of group ports. The new settings take effect immediately.

In addition to the CONTrol, NAme, and LogicalNET parameters, you can use the -PORT CONFiguration parameter on group ports. To configure other port characteristics, configure them on member ports rather than the group port.

When you configure a logical network, you must enable global bridging and per-port transparent bridging on all member ports. For information, refer to , "Bridging over Multiple Logical Networks" section on page 3-2.

When a network routing protocol configures the group port in its network topology, it configures attributes for the entire port group. To configure the IP routing protocol on a logical network, refer to i"Configuring Logical Networks over IP" on page 6-9.

2 CONFIGURING FDDI

This chapter describes the following information on configuring the Fiber Distributed Data Interface (FDDI):

- Port configuration for FDDI usage
- FDDI maintenance and troubleshooting information

Configuring Ports for FDDI

When an FDDI board is installed in your system, the software automatically sets the port ownership for the corresponding port to FDDI. FDDI port configuration is transparent and no additional user-configuration activity is required.

Troubleshooting the Configuration

If you have problems making FDDI connections to other networks after setting up your router, review the following troubleshooting procedure. This procedure can help you diagnose various network and internal hardware problems. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier for assistance. For more information on the FDDI commands and parameters discussed in this chapter, refer to Chapter 1 and Chapter 22 in *Reference for NETBuilder Family Software*.

Diagnosing Internal Hardware Problems

If both PHY LEDs on the media access control (MAC) board do not light green after you initialize the system and connect to an operational ring, you need to perform a self-test.

To perform the self-test, follow these steps:

1. Remove the two connections from your station to the network, then loop PHY port A to PHY port B using a length of standard media interface connector/media interface connector (MIC/MIC) fiber-optic cable.

   This connection puts your station into loopback mode. If both PHY LEDs are green, the FDDI interface on your station is operating normally and the problem exists elsewhere, either with a neighbor station or with the line itself. Steps 3 through 5 describe how to perform line-state testing.

   If either of the two PHY LEDs are red while in loopback mode, your PHY board is defective and should be replaced.

2. Remove the loopback connection made in step 1 and reconnect your station to the network.

3. To perform line-state testing for each port, first set ports A and B to maintenance state using the PCControl parameters:

   ```
   SET !<path> -FDDI PCControlA = Maint [sets port A]
   SET !<path> -FDDI PCControlB = Maint [sets port B]
   ```

4 Display the current line states using the Maintenance Line State parameters:

```
SHow !<path> -FDDI MaintLineStateA
SHow !<path> -FDDI MaintLineStateB
```

These commands display the line state of the transmitter first, then the line state of the receiver.

5 Use the Idle and Halt values of the MaintLineState parameters to conduct additional tests of the lines:

```
SET !<path> -FDDI MaintLineStateA = Idle
SET !<path> -FDDI MaintLineStateB = Idle
SET !<path> -FDDI MaintLineStateA = Halt
SET !<path> -FDDI MaintLineStateB = Halt
```

These commands cause port A or B to transmit Idle or Halt symbols continuously.

If the line states being transmitted match the received line states at the other end of the fiber-optic cable for both directions, the fiber-optic transceivers at each end are in normal working order and are compatible. This is only a static test and does not diagnose an intermittent component failure.

Diagnosing Network Problems

Use the following FDDI parameters to help diagnose network problems:

- Use the PortNeighbor parameter to determine whether an undesirable connection is in operation, such as port A attempting to communicate with a port A neighbor.

  For normal dual-attachment operation, the neighbor for port A should be port B, and the neighbor for port B should be port A. For example, to display the port A and port B neighbor types for path 2, enter:

  SHow !2 -FDDI PortNeighbor

  A message similar to the following appears:

  ```
  Port A: PortNeighbor = B
  Port B: PortNeighbor = A
  ```

- Use the SMTAddress, UpNeighbor, and DownNeighbor parameters to determine the MAC address (12 hex characters) of your 3Com bridge/router FDDI station and of your neighbors on the ring.

  For example, to display the MAC addresses of the upstream and downstream neighbors for path 2, enter:

  SHow !2 -FDDI UpNeighbor DownNeighbor

- Use the DupAddress parameter to determine if your station has detected a duplicate MAC address on the FDDI ring.

  The DupAddress parameter displays the setting of the duplicate address flag. The duplicate address flag is set to the Detected state when a frame is detected with a MAC address that is a duplicate of the MAC address of the receiving station. For example, to display the duplicate address flag setting for path 2, enter:

  SHow !2 -FDDI DupAddress

- Use the StationCONFig parameter to determine if the station is in a wrap or through state.

  A wrap state occurs when one of the fiber-optic links is operational and one has failed, or when a port is connected to a concentrator. For example, to display the current configuration state of your station, enter:

  ```
  SHow !1 -FDDI StationCONFig
  ```

- Use the RemDisconnect parameter to determine if a disconnect was requested by a remote management station, when a station disconnects automatically (all LEDs red).

  The RemDisconnect parameter displays the current value of the Remote Disconnect Flag. When set (value = yes), this flag indicates that the station has been remotely disconnected. For example, to display the current value of the Remote Disconnect Flag for path 1, enter:

  ```
  SHow !1 -FDDI RemDisconnect
  ```

3 CONFIGURING BRIDGING

This chapter describes how to set up, customize, and troubleshoot a bridge.

If you need to configure source route bridging, refer to Chapter 5.

For conceptual information, refer to "How the Bridge Works" on page 3-19.

Configuring Basic Bridging

This section describes how to set up a basic bridge. After you perform these steps, you can continue using the default values of parameters, or you can customize the bridge according to "Customizing the Bridge" on page 3-8.

Prerequisites

This section assumes that you have logged on with Network Manager privilege and set up ports and paths according to Chapter 1.

To set up your bridge, first perform the procedure in "Transparent Bridging." You must then decide if any protocols that will be used on your network are source-route-only protocols; if they are, you may also need to perform the procedures described in Chapter 5. (To determine this, refer to the documentation that came with your protocol software.) If you want to set up your bridge as a wide area bridge, refer to "Bridging over a Wide Area Network" at the bottom of this page.

Transparent bridging is supported on Ethernet, token ring, Fiber Distributed Data Interface (FDDI), Point-to-Point Protocol (PPP), Frame Relay, Asynchronous Transfer Mode (ATM), X.25, Switched Multimegabit Data Service (SMDS), and Integrated Services Digital Network (ISDN).

Transparent Bridging

To set up a transparent bridge, enable bridging on the entire bridge/router by entering:

```
SETDefault -BRidge CONTrol = Bridge
```

If you do not want transparent bridging enabled on all ports, you can disable it on an individual port basis. For more information, refer to "Per-Port Transparent Bridging" on page 3-9.

Bridging over a Wide Area Network

You can set up your transparent bridge to forward packets over the following types of wide area networks:

- Frame Relay and Asynchronous Transfer Mode Data Exchange Interface (ATM DXI)
- X.25
- ATM

- SMDS
- PPP and Phone Line Gateway (PLG)
- ISDN

Bridging over Frame Relay, ATM DXI, X.25, and ATM is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to bridge Frame Relay, ATM DXI, or X.25 over a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. When bridging over ATM in meshed, partially meshed, and nonmeshed topologies, you must create virtual ports. For configuration information, including a discussion of fully meshed, partially meshed, and nonmeshed topologies and virtual ports, refer to Chapter 42, Chapter 43, Chapter 45, and Chapter 47. For information on the number of virtual ports supported on each bridge/router platform, refer to Table 1-1 on page 1-4.

If you configure bridging over Frame Relay in a meshed topology with multiple data link connection identifier (DLCI) neighbors, the transmission of unknown unicast addresses and multicast (including broadcast) frames is processed separately from packets sent to known addresses. As a result, some frames may arrive at the destination nodes out of order.

For information on configuring transparent bridging over X.25 using X25User or X25DTE type profiles, refer to Chapter 45 in this guide and to Chapter 45 in *Reference for NETBuilder Family Software*.

Bridging over SMDS is supported over a fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach groups of fully meshed devices). For more information, refer to Chapter 44.

To configure bridging over PPP and PLG, refer to Chapter 34. For information about using the Spanning Tree Protocol over PPP, refer to "Using the Algorithm with Wide Area Bridges" on page 3-29 and "Configuring the Spanning Tree Protocol over PPP" on page 3-30.

For more information on wide area networking using ISDN, refer to Chapter 35.

Bridging over Multiple Logical Networks

When you configure multiple logical networks (MLN), you must enable global transparent bridging by entering:

```
SETDefault -BRidge CONTrol = Bridge
```

If you do not enable global transparent bridging, MLN will be unable to bridge the configured network protocol, because stations on member ports will not be learned.

Over logical networks, software version 8.2 supports only transparent bridging, not source route bridging. The only valid media type is Ethernet. For information about logical networks, refer to "Logical Networks" on page 1-8.

When you configure MLN, per-port bridging must remain enabled on all member ports. To display the per-port bridging configuration, use:

```
SHow !<port> -BRidge TransparentBRidge
```

If per-port bridging is disabled on any ports in a port group, re-enable it using:

```
SETDefault !<port> -BRidge TransparentBRidge = TransparentBRidge
```

For complete information on the -BRidge TransparentBRidge parameter, refer to Chapter 14 in *Reference for NETBuilder Family Software*.

A bridge/router configured with MLN is normally connected to a backbone at the outside edge of a network. Consolidating networks into a logical network provides a desirable way to access a backbone: ports within the logical network are bridged, and access to the backbone is routed.

To provide further connectivity, you can also include bridges or switches within a logical network, as shown in Figure 3-1. Do not interconnect logical networks with bridges, as shown in Figure 3-2, or bridge between a logical network and another LAN segment. This topology defeats the MLN configuration and can cause your network to operate incorrectly.

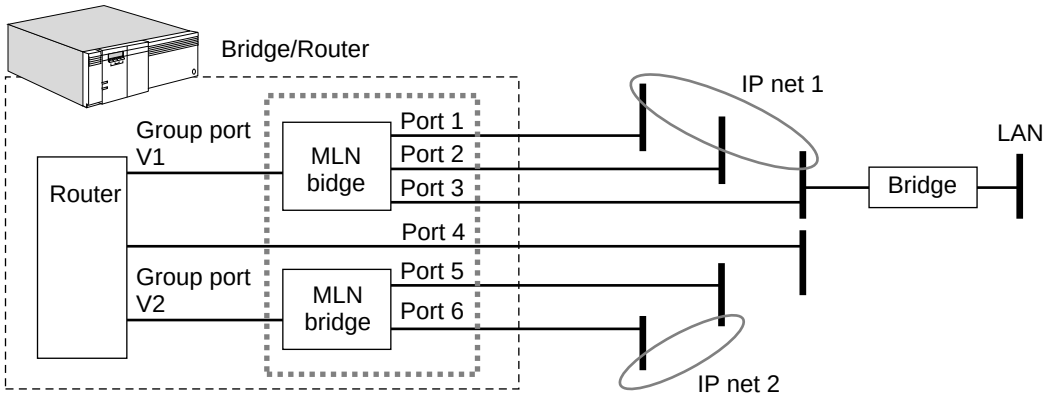


Figure 3-1 Bridge within a Logical Network

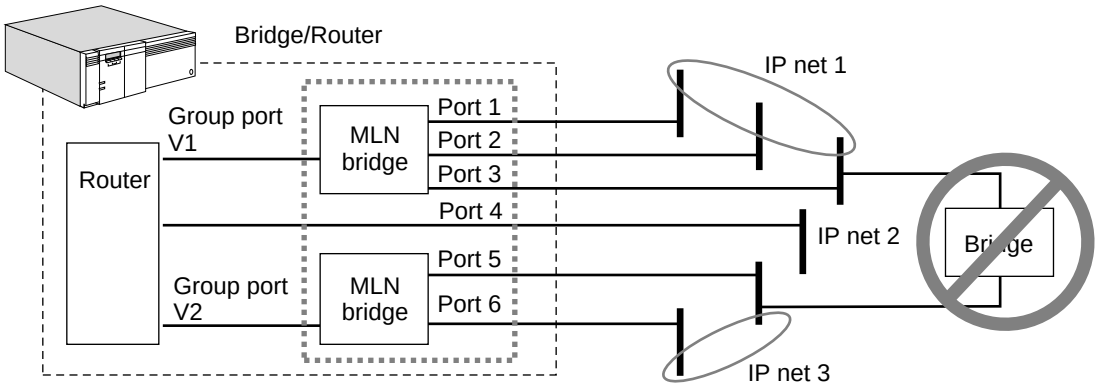


Figure 3-2 Bridge between Logical Networks

Configuring for Bridging and Routing

To configure your bridge/router, follow these steps:

1. Prepare your bridge/router for bridging.
2. Follow the instructions in the appropriate routing chapter to prepare your bridge/router for routing.

See Table 3-1 to find information on configuring specific protocols.

Table 3-1 Protocol Configuration

Protocol	Chapter
AppleTalk	15
APPN	11
DECnet	16
Internet Protocol (IP)	7
Internet Packet Exchange (IPX)	14
Open System Interconnection (OSI)	17
Vines	18
Xerox Network Systems (XNS)	19

You may need to refer to more than one chapter if your bridge/router will be used to route packets of different protocols.

3 Decide whether FireWall should be configured in the CONTrol parameter of the BRidge Service.

FireWall causes the bridge/router to discard all packets of a configured protocol that are addressed to destinations other than the bridge/router. There is a performance cost when FireWall is enabled, because every bridged packet must be checked. NoFireWall bypasses this check, resulting in better bridging performance.

NoFireWall is selected by default. If you need to change the CONTrol parameter, use:

```
SETDefault -BRidge CONTrol = FireWall
```

or

```
SETDefault -BRidge CONTrol = NoFireWall
```

FireWall ensures that protocols configured for routing are never bridged by checking a type field in every packet to determine whether the packet should be routed or bridged. This level of checking slows down the performance of the bridge/router.

NoFireWall causes the router to skip the type field (except for broadcast packets, which are always checked). The bridge/router routes only data that is sent directly to the router at the MAC layer. All other packets are bridged. NoFireWall improves performance, but may forward incorrectly addressed packets. For example, if the IP protocol is configured for routing, and an IP packet is received with an address different from the router address, that IP packet is bridged.

In Figure 3-3, the bridge/router performs IP routing and bridging of other protocols between networks 1 and 2. If station 1 wants to route a packet to station 2 via the bridge/router, the packet should be addressed to the bridge/router. If for some reason an IP packet is unicast to station 2, the FireWall value ensures that the bridge/router discards this packet.

If a bridge/router receives packets of a protocol type that has not been configured on it, it forwards the packets as if it were a bridge. In Figure 3-3, the bridge/router uses bridging to forward OSI packets between networks 1 and 2, because it is not configured for OSI routing.

Broadcast packets are not forwarded for any routed protocol, even if the -BRidge CONTrol parameter is set to NoFireWall. This procedure reduces unnecessary network traffic.

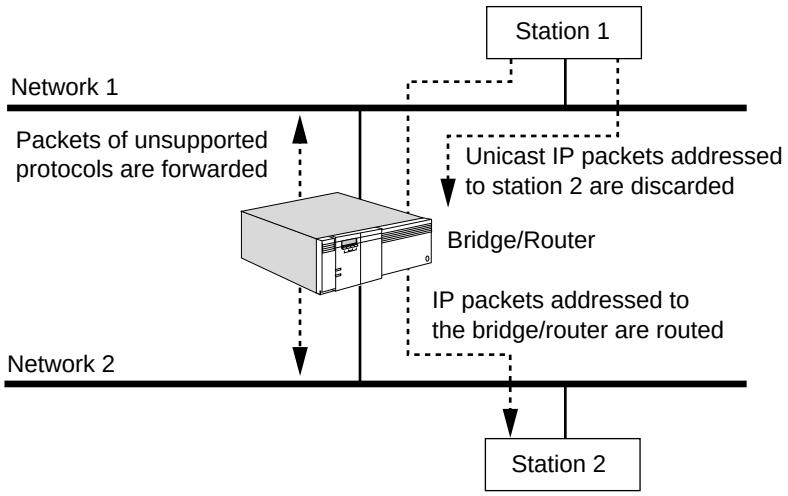


Figure 3-3 Effects of FireWall on Bridging and Routing

Verifying the Configuration

When you finish configuring your bridge, verify its configuration by following these steps:

1 Verify the values assigned to the -BRidge CONTrol parameter by entering:

```
SHow -BRidge CONTrol
```

The current values of the -BRidge CONTrol parameter are displayed. FOrward, LEarn, Aging, and NoFireWall, which are the default values, are usually selected. If the setting of any of these values has been changed, the bridge will perform one or more of the following processes:

- Not forward packets.
- Use only user-defined routes in its routing table to forward packets.
- Not check for addresses of nodes that appear to be "dormant."
- Discard unicast packets of a protocol that is being routed (except for unicast packets to the bridge itself).

2 Verify that transparent bridging is enabled on the appropriate ports by entering:

```
SHow -BRidge TransparentBRidge
```

The current values of the -BRidge TransparentBRidge parameter are displayed. Make sure that TransparentBRidge is selected on the appropriate ports.

3 Verify that the Spanning Tree Protocol is enabled by entering:

```
SHow -STP CONTrol
```

The current values of the -STP CONTrol parameter are displayed. Make sure that Enabled is selected, which is the default value, to ensure that the bridge participates in the spanning tree network configuration.

4 Check the configuration of the transparent bridge and the status of each port and path by entering:

```
SHow -BRidge CONFiguration
```

This is a sample display for a NETBuilder II 4-Slot chassis:

```
.............................Current Configuration Values.............................
CONTrol = (Aging,Bridge,NoFireWall,FOrward,LEarn)
AgeTime = 300
          Name       State         Type         Status                    SRcSec     DStSec
          ----       ----          ----         ----                      ----       ----
Port 1    Port_1     Forwarding    TokenRing    Reachable                 None       None
Path 1    Path_1                                Up (Wed Dec 31 16:00)
Port 2    Port_2     Forwarding    TokenRing    Reachable                 None       None
Path 2    Path_2                                Up (Wed Dec 31 16:00)
Port 3    Port_3     Disabled      Remote       Unreachable               Down       None
Path 3    Path_3                   64 Kbps      Down (Wed Dec 31 16:00)
Port 4    Port_4     Disabled      Remote       Unreachable               Down       None
Path 4    Path_4                   64 Kbps      Down (Wed Dec 31 16:00)
```

The first two lines of this display show the current values of the -BRidge CONTrol and -BRidge AgeTime parameters. The remaining lines show the status of each port and its associated paths, including the name (if any) of the type of line, the time at which the paths status last changed, and the source and destination security.

5 If the display indicates that a port or a path is down, follow these steps:

   a Check the configuration of each port by entering:

   SHow -PORT CONFiguration

   b Check the configuration of each path by entering:

   SHow -PATH CONFiguration

6 Test the bridge by sending packets across it.

For example, make a connection from a device on one attached network to a host on another attached network. If you can successfully make a connection, the bridge is ready for normal operation. If you cannot make a connection, refer to "Troubleshooting the Configuration" on page 3-7.

Getting Statistics

After your bridge is running, you may want to gather statistics.

You can collect statistics for a specified time period by using the -SYS SampleTime and -SYS STATistics parameters. For more information, refer to Chapter 58 in *Reference for NETBuilder Family Software*.

For information on interpreting statistics displays, refer to Appendix H.

To gather statistics, follow these steps:

1 Display statistics for bridged packets by entering:

SHow -SYS STATistics -BRidge

2 Display statistics for all ports by entering:

SHow -SYS STATistics -PORT

3 Display statistics for all paths by entering:

SHow -SYS STATistics -PATH

If the display indicates that there are errors on the attached network (for example, cyclic redundancy check errors), check the following items:

- The transceiver cable is properly attached to the transceiver.
- The transceiver is properly attached to the network cable.
- The network is properly terminated.

If the errors happen on a serial line, check the following items:

- Cable attachments
- Channel service units (CSUs) and digital service units (DSUs)
- Modems on each end of the serial line

If the line is a leased line, request help from the company that leases you the line (for example, the telephone company).

Troubleshooting the Configuration

To troubleshoot the bridge, follow these steps:

1 If one or more devices cannot communicate across the bridge, determine whether the filtering feature has been enabled and if so, what types of filters have been implemented.

a Determine whether filtering has been enabled by entering:

```
SHow -FIlter CONTrol
```

b If filtering has been enabled, review the filters by entering:

```
SHow -FIlter MASK
SHow -FIlter POLicy
```

A filter defined incorrectly can cause packets destined for certain addresses to be discarded.

c If filtering has been enabled, set up a counter to record the number of packets of a particular type that are forwarded by the bridge, using the -FIlter POLicy parameter. For more information on bridge filtering, refer to Chapter 4.

2 Determine if source and destination explicit forwarding have been implemented using:

```
SHow [!<port> | !*] -BRidge SRcSecurity
SHow [!<port> | !*] -BRidge DStSecurity
```

3 Check the routing table by entering:

```
SHow -BRidge AllRoutes
```

The address of the affected station should appear in the routing table, followed by the correct destination network number (port). If the address does not appear, make sure that the -BRidge CONTrol parameter settings include LEarn and FOrward. If necessary, enter the ADD -BRidge ROUte command to add the address of the affected station.

4 Display bridge configuration information and check the status of each path. Verify that each path is assigned to the appropriate network by entering:

```
SHow -BRidge CONFiguration
```

Check the physical attachments of any network not listed as REACHABLE or any path not listed as UP. Verify that the path is enabled by entering:

```
SHowDefault -PORT CONFiguration
SHowDefault -PATH CONFiguration
```

5 If the display indicates that a port or a path is down, follow these steps:

   a Check the configuration of each port by entering:

   ```
   SHow -PORT CONFiguration
   ```

   b Check the configuration of each path by entering:

   ```
   SHow -PATH CONFiguration
   ```

6 Check for other activity on the bridge.

   If there is no other activity, check the physical attachments of the bridge to its networks, including boards, back panel connectors, and transceiver or modem connectors. For lines to wide area bridges, check the CSU/DSU or modem and its configuration.

7 If a large number of errors occur on the serial line of a wide area bridge to a remote network, check the physical lines. For a detailed account of errors on a given path, enter:

   ```
   SHow -SYS STATistics -PATH
   ```

   You can set some statistics to zero using:

   ```
   FLush -SYS STATistics [-<service>]
   ```

8 If a pair of devices cannot communicate across the bridge, check to see whether another pair can communicate across it.

   If the second pair communicates, the problem is the first pair of devices, not the bridge.

   To determine whether a pair of bridges can communicate with each other, initiate the data link test using the DLTest command. This test allows the bridges to exchange test packets and displays the resulting statistics. For more information on the DLTest command, refer to Chapter 1 in *Reference for NETBuilder Family Software.*

9 If possible, replace any bridge you suspect has a problem with another bridge or a repeater.

   If the problem persists, the bridge is not the cause.

Customizing the Bridge

This section briefly describes the following features that enable you to customize your bridge:

- Per-port transparent bridging
- Static routes
- Bridge security
- Filters
- Translation bridging

This section discusses only the DStSecurity, FunctionalAddr, MultiCastAddr, ROUte, SRcSecurity, and TransparentBRidge parameters. For information on

other BRidge Service and STP Service parameters, refer to Chapter 14 and Chapter 57 in *Reference for NETBuilder Family Software*.

If you configure a logical network (port group), you can customize it by configuring global bridge properties, such as AgeTime, or individual properties of member ports, such as the ones described in this section. Configuring properties on the group port has no effect.

Per-Port Transparent Bridging

In addition to enabling transparent bridging on all ports (by setting the -BRidge CONTrol parameter to Bridge), you can enable transparent bridging on specified ports only using:

```
SETDefault !<port> -BRidge TransparentBRidge = TransparentBRidge
```

To disable transparent bridging, use:

```
SETDefault !<port> -BRidge TransparentBRidge = NoTransparentBRidge
```

For complete information on the -BRidge TransparentBRidge parameter, refer to Chapter 14 in *Reference for NETBuilder Family Software*.

Adding or Deleting Static Entries

To add a static (permanent) entry in the routing table used by your bridge, use:

```
ADD !<port> -BRidge ROUte <address>
```

To delete a routing table entry, use:

```
DELete [!<port>] -BRidge ROUte <address>
```

CAUTION: *When you change the owner for any WAN port, you must delete all static routes that were configured for the previous owner and WAN type. Use the DELete -BRidge ROUte command to delete these routes. Failing to delete the routes can cause a crash (fatal error) in the bridge/router software.*

For more information on the -BRidge ROUte command, refer to Chapter 14 in *Reference for NETBuilder Family Software*. For more information on routing tables, refer to "Routing Tables" on page 3-32.

Bridge Security

You can use bridge security features to select certain stations whose packets will be forwarded or blocked depending on their source or destination address. These features are applied only to packets traveling from one port of the bridge to another port. Packets addressed to the bridge are not affected.

The security features include:

- Source explicit forwarding.
- Source explicit blocking.
- Destination explicit forwarding.
- Destination explicit blocking.
- Combined source and destination security.

CAUTION: *Before you use the SRcSecurity and DStSecurity parameters, read the descriptions and examples in this chapter and Chapter 14 in Reference for NETBuilder Family Software. The SRcSecurity and DStSecurity parameters can affect bridge performance. Incorrect use of these parameters can cause the bridge to discard packets that you want to forward or to forward packets that you want to discard.*

For information on restricting packet movement based on packet contents, refer to "Filters" on page 3-15.

Source Explicit Forwarding

The Source Explicit Fowarding (SEF) feature allows you to forward packets from specific source addresses, on a per-port basis, in conjunction with a routing table.

To forward packets using the source explicit forwarding feature, you must enable forwarding on the port where the packet enters the bridge using:

```
SETDefault !<port> -BRidge SRcSecurity = Fwd
```

For a packet to be forwarded, its source address must be a static entry in the routing table on the port where the packet enters the bridge. Static entries are added or deleted using the ADD or DELete -BRidge ROUte <address> syntax. All other packets are discarded.

Some packets that meet forwarding conditions cannot be forwarded, because they are blocked by other constraints such as filtering or destination explicit blocking.

Figure 3-4(a) shows a bridge connecting two Ethernet networks, network A and network B. All stations on network B can communicate with all stations on network A. However, you can restrict packet forwarding so that only stations 1 and 2 on network A can communicate with the stations on network B. If stations 3 to 20 on network A send packets to any of the stations on network B, the packets are discarded.

To configure source explicit forwarding, follow these steps:

1. Set the -BRidge SRcSecurity parameter to Fwd on the port where the packet enters the bridge.

   For example, to set this parameter to Fwd on port 1 of the bridge shown in Figure 3-4(a), enter:

   SETDefault !1 -BRidge SRcSecurity = Fwd

2. Add a static entry in the routing table for each source station from which you want packets to be forwarded.

   The static entry must be generated on the port where the packet enters the bridge.

   For example, to add a static entry for station 1 in Figure 3-4(a), enter:

   ADD !1 -BRidge ROUte %080002001234

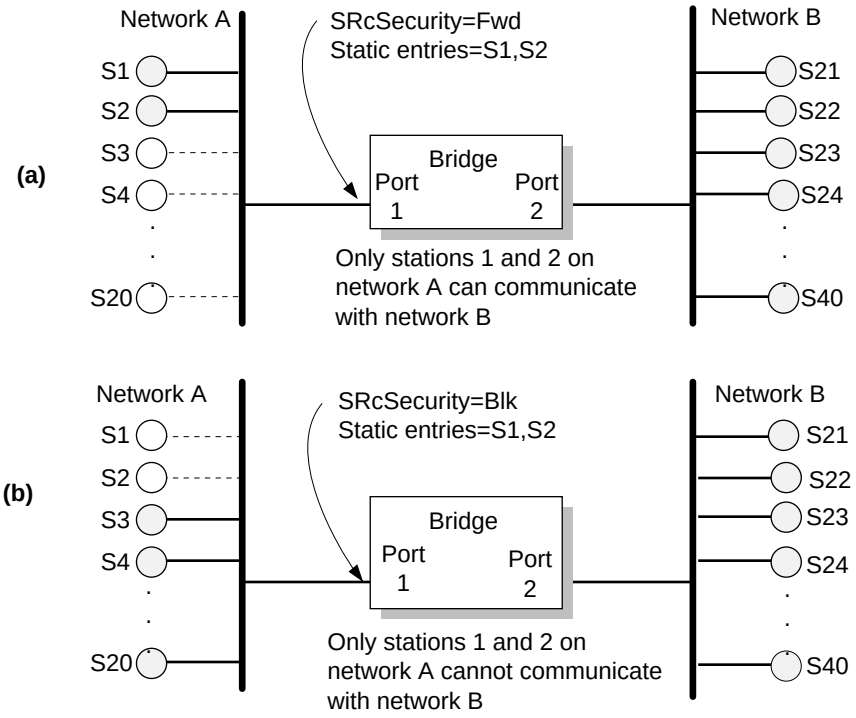


Figure 3-4 Source Explicit Forwarding and Blocking

Source Explicit Blocking

The Source Explicit Blocking (SEB) feature allows you to discard packets from specific source addresses on a per-port basis in conjunction with a routing table. The blocking feature is the reverse of forwarding. Choose whichever feature (forwarding or blocking) allows you to enter fewer source addresses in the routing table.

On any port, only one of the two features, forwarding and blocking, can be turned on at a time.

To block packets using the source explicit blocking feature, you must enable the block feature on the port where the packet enters the bridge using:

```
SETDefault !<port> -BRidge SRcSecurity = Blk
```

For a packet to be blocked, its source address must be a static entry in the routing table on the port where the packet enters the bridge. Static entries are added or deleted using the ADD or DELete -BRidge ROUte <address> syntax. All other packets are forwarded (subject to other constraints).

Figure 3-4(b) shows a bridge connecting two Ethernet networks, network A and network B. All stations on network B must be able to communicate with all stations on network A. However, you can restrict packet forwarding so that if stations 1and 2 on network A send packets to the stations on network B, the packets are discarded. If stations 3 to 20 on network A send packets to the stations on network B, they are forwarded.

To set up source explicit blocking, follow these steps:

1 Set the -BRidge SRcSecurity parameter to Blk on the port where the packet enters the bridge.

For example, to set this parameter to Blk on port 1 of the bridge shown in Figure 3-4(b), enter:

```
SETDefault !1 -BRidge SRcSecurity = Blk
```

2 Add a static entry in the routing table for each source station from which you want packets to be blocked.

The static entry must be generated on the port where the packet enters the bridge.

For example, to add a static entry for station 1 as shown in Figure 3-4(b), enter:

```
ADD !1 -BRidge ROUte %080002001234
```

If Source Explicit Forwarding were used in this example, the addresses of stations 3 to 20 would have to be manually entered in the routing table, requiring more work for the network manager.

Destination Explicit Forwarding

The Destination Explicit Forwarding (DEF) feature allows you to forward packets to specific destination addresses, on a per-port basis, in conjunction with a routing table.

To forward packets using this feature, you must enable the forward feature on the port where the packet enters the bridge using:

```
SETDefault !<port> -BRidge DStSecurity = Fwd
```

For a packet to be forwarded, its destination address must be a static entry in the routing table on the port where the packet exits the bridge. Static entries are added or deleted using the ADD or DELete -BRidge ROUte <address> syntax. All other packets are discarded.

Figure 3-5(a) shows a bridge connecting two Ethernet networks in a company, network A and network B. All stations on network B must be able to communicate with all stations on network A. However, you can set DEF so that network A stations can send packets only to two stations on network B: station 21 and station 22. If stations on network A send packets to stations other than 21 or 22 on network B, the packets are discarded.

To configure destination explicit forwarding, follow these steps:

1 Set the -BRidge DStSecurity parameter to Fwd on the port where the packet enters the bridge.

For example, to set this parameter to Fwd on port 1 of the bridge shown in Figure 3-5(a), enter:

```
SETDefault !1 -BRidge DStSecurity = Fwd
```

2 Add a static entry in the routing table for each destination station that you want packets forwarded to.

The static entry must be generated on the port where the packet exits the bridge.

For example, to add a static entry for station 21 as shown in Figure 3-5(a), enter:

```
ADD !2 -BRidge ROUte %080002001234
```

Some packets that meet forwarding conditions may not be forwarded, because they are blocked by other constraints such as filtering or source explicit blocking.

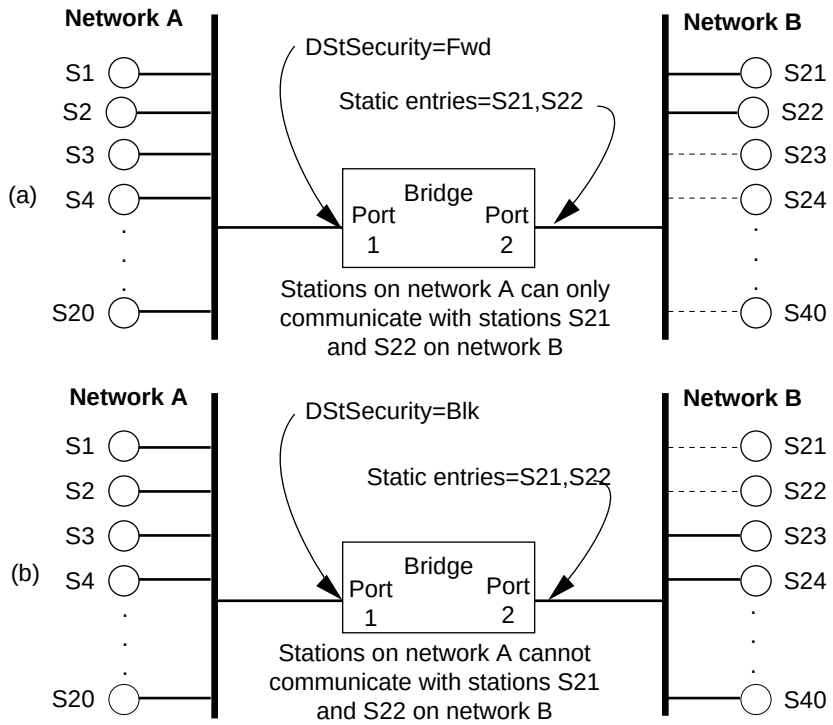


Figure 3-5 Destination Explicit Forwarding and Blocking

Destination Explicit Blocking

The Destination Explicit Blocking (DEB) feature allows you to discard packets sent to specific destination addresses, on a per-port basis, in conjunction with a routing table. The blocking feature is the reverse of forwarding. Choose whichever feature (forwarding or blocking) allows you to enter fewer source addresses in the routing table.

On any port, only one of the two features, forwarding and blocking, can be turned on at a time.

To block packets using this feature, you must enable the blocking feature on the port where the packet enters the bridge using:

```
SETDefault !<port> -BRidge DStSecurity = Blk
```

The destination address must be a static entry in the routing table on the port where the packet exits the bridge. Static entries are added or deleted using the ADD or DELete -BRidge ROUte <address> syntax. All other packets are forwarded (subject to other constraints).

Figure 3-5(b) shows a bridge connecting two Ethernet networks, network A and network B. Any station on network B must be able to communicate with any station on network A. You can set Destination Explicit Blocking so that all stations on network A can communicate with the stations on network B except for stations 21 and 22. If stations on network A send packets to stations 21 or 22 on network B, the packets are discarded.

To configure destination explicit blocking, follow these steps:

1. Set the -BRidge DStSecurity parameter to Blk on the port where the packet enters the bridge.

   For example, to set this parameter to Blk on port 1 of the bridge shown in Figure 3-5(b), enter:

   ```
   SETDefault !1 -BRidge DStSecurity = Blk
   ```

2. Add a static entry in the routing table for each destination station that you do not want packets forwarded to.

   The static entry must be generated on the port where the packet exits the bridge.

   For example, to add a static entry for station 21 as shown in Figure 3-5(a), enter:

   ```
   ADD !2 -BRidge ROUte %080002001234
   ```

   In this example, the DStSecurity parameter can be set to Fwd, but the addresses of stations 23 to 40 must be manually entered in the routing table, requiring more work for the network manager.

   If you want the bridge to discard all the packets destined for a particular address, use:

   ```
   ADD -BRidge ROUte <address> Off
   ```

Combined Source and Destination Security

You can build a complex bridge security system by combining SEF, SEB, DEF, and DEB features.

The -BRidge SRcSecurity and -BRidge DStSecurity commands can be turned on at the same port. However, if both forwarding and blocking conditions are met, the blocking condition takes precedence and the packet is discarded.

For example, suppose that the SRcSecurity parameter is set to Fwd and the DStSecurity parameter is set to Blk. A packet originating from an address that is not a static entry does not meet the forwarding condition. This packet is discarded regardless of its destination address.

A packet originating from an address that is a static entry does meet the forwarding condition. However, if the destination address of this packet is also a static entry, then this packet meets the blocking condition and the packet is discarded.

Packets traveling on a network combining these features reach their destination only if all forwarding conditions are met and no blocking conditions are met.

Figure 3-6 shows a bridge connecting two Ethernet networks, network A and network B.

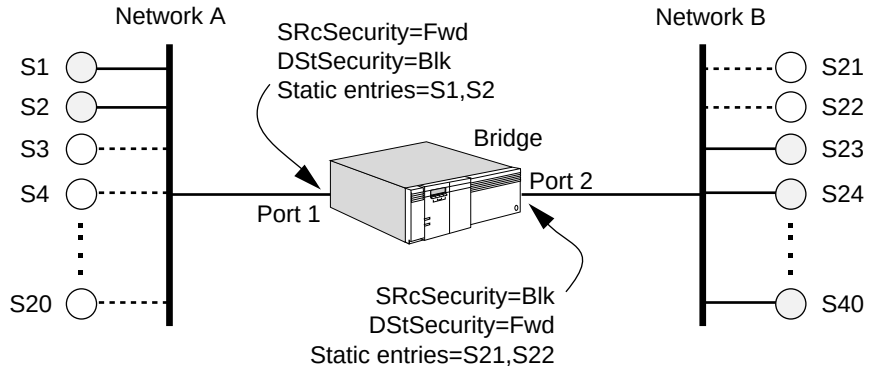


Figure 3-6 Combining Bridge Security Features

The network manager has set up the following complex system of bridge security features that restrict communication between the two networks:

- Stations 1 and 2 on network A *cannot* communicate with stations 21 and 22 on network B, because DStSecurity on port 1 is set to Blk and stations 21 and 22 are static entries on port 2. Although SRcSecurity is set to Fwd, and stations 1 and 2 on network A are static entries on port 1, the setting is ignored because blocking takes precedence over forwarding.
- Stations 1 and 2 *can* communicate with stations 23 to 40 on network B, because:
  - SRcSecurity is set to Fwd on port 1.
  - Stations 1 and 2 are static entries on port 1.
  - DStSecurity is set to Blk on port 1, but stations 23 to 40 are not static entries on port 2.
- Stations 3 to 20 on network A *cannot* communicate with any station on network B because:
  - SRcSecurity is set to Fwd on port 1.
  - None of the stations 3 to 20 are listed as static entries on port 1.
- Stations 21 and 22 on network B *cannot* communicate with any station on network A because:
  - SRcSecurity is set to Blk on port 2.
  - Stations 21 and 22 are static entries on port 2.
- Stations 23 to 40 on network B *can* communicate with stations 1 and 2 on network A because:
  - SRcSecurity is set to Blk on port 2 but stations 23 to 40 are not static entries on port 2.
  - DStSecurity is set to Fwd on port 2 and stations 1 and 2 on network A are static entries on port 1.

Filters You can use the -FIlter MASK and -FIlter POLicy parameters to define a custom filter so that packets meeting the criteria specified in the filter are forwarded or discarded. You can also restrict forwarding of packets by protocol type or other packet contents. For complete information on configuring filters and policies, refer to Chapter 4.

Translation Bridging

Translation bridging is enabled by default, and there is a default table of functional-address-to-multicast-address mappings for well-known protocols. You do not need to take action unless you want to:

- Add an additional functional-address-to-multicast-address mapping to the default table.
- Use translation bridging across a serial line running PPP, Frame Relay, ATM DXI, X.25, or SMDS that connects a NETBuilder II bridge/router with an Ethernet module installed to a NETBuilder II bridge/router with a token ring module installed, as shown in Figure 3-7. If your network is configured in this way, you must be sure that the address resolution translation at each port that terminates the serial line is the same.

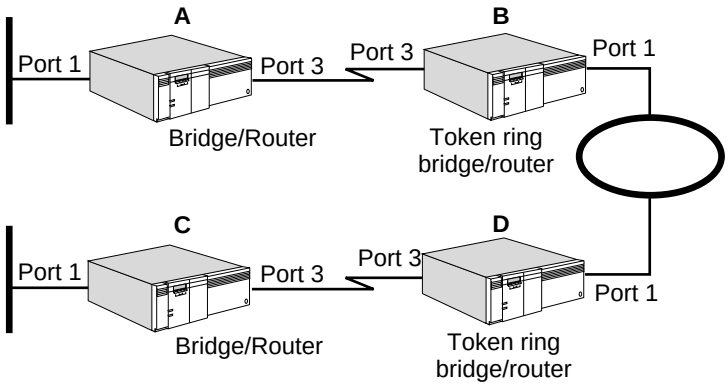


Figure 3-7 ARP Translation in a Bridged Environment

Table 3-2 lists protocol support in a mixed-media bridged environment. In this environment, all protocols are supported when one media type is communicating with a similar media type. 3Com does not support mapping of source-routed frames to transparent frames in a bridged environment except using source route transparent bridging gateway (SRTG) or Logical Link Control type 2 (LLC2) with data link switching. For more information, refer to Chapter 5 and Chapter 24. To be translated by the 3Com implementation, all protocol packets must be transparent frames.

Table 3-2 3Com Protocol Support in a Translation Bridge Environment

Protocol	Ethernet to Token Ring End Station Connectivity	Routable Protocol
ARP	Yes	Yes
TCP/IP	Yes	Yes
XNS	No	Yes
IPX	No	Yes
AppleTalk	No	Yes
NetBEUI	Yes*	No
IBM NETBIOS	Yes*	No
3Com NBP	No†	No
LAT	Yes*	No

(continued)

Table 3-2 3Com Protocol Support in a Translation Bridge Environment (continued)

Protocol	Ethernet to Token Ring End Station Connectivity	Routable Protocol
SNA	Yes	No
DECnet	Yes	Yes
Banyan VINES	No	Yes

* For these protocols, the user must map the following multicast addresses to a unique functional address:
IBM NETBIOS30000000001
MS NetBeui30000000001
LAT09002B00000F or 09002B020104
For LAT, officially assigned multicast-address-to-functional-address mappings should be added to the table with the -BRidge MultiCastAddr command. The address mapping should also be set on all bridges the protocol traverses.

† NBP Connection Request Datagram checksum is computed by NBP protocol on receipt to be different than the checksum value within the datagram; connection request is discarded.

Adding Functional-Address-to-Multicast-Address Mappings to the Default Table

This section describes how to add a functional-address-to-multicast-address mapping to the default table. For conceptual information, refer to "Address Mapping" on page 3-23.

To add a functional address, follow these steps:

1 Map a functional address to a multicast address using:

```
ADD -BRidge FunctionalAddr = %<address> MultiCastAddr = %<address>
```

2 Map a multicast address to a functional address using:

```
ADD -BRidge MultiCastAddr = %<address> FunctionalAddr = %<address>
```

For complete information on the -BRidge FunctionalAddr and -BRidge MultiCastAddr parameters, refer to Chapter 14 in *Reference for NETBuilder Family Software*.

There are 31 token ring functional addresses allowed, of which only 12 are user-configurable. Table 3-3 lists these user-configurable addresses. Some addresses may already be in use for other purposes.

Table 3-3 User-Configurable Addresses

Noncanonical Format	Canonical Format	Possible Other Use
C000 0010 0000	0300 0008 0000	AppleTalk ZIP/NBP
C000 0020 0000	0300 0004 0000	AppleTalk ZIP/NBP
C000 0040 0000	0300 0002 0000	AppleTalk ZIP/NBP
C000 0080 0000	0300 0001 0000	AppleTalk ZIP/NBP
C000 0100 0000	0300 8000 0000	AppleTalk ZIP/NBP, AMP discovery, DEC LAT
C000 0200 0000	0300 4000 0000	AppleTalk ZIP/NBP, DEC LAT
C000 0400 0000	0300 2000 0000	AppleTalk ZIP/NBP, DEC NetBIOS
C000 0800 0000	0300 1000 0000	AppleTalk ZIP/NBP, DECnet Phase IV
C000 1000 0000	0300 0800 0000	AppleTalk ZIP/NBP, DECnet Phase IV
C000 2000 0000	0300 0400 0000	AppleTalk ZIP/NBP
C000 4000 0000	0300 0200 0000	

Setting the Address Format

Protocol implementations on token ring can carry hardware MAC addresses within the protocol packets in either canonical or noncanonical format. End stations on Ethernet and FDDI always use canonical format. End stations on token ring can use either format. When a protocol implementation on token ring uses the noncanonical format within the protocol packet and is connected by a bridge to an Ethernet or FDDI LAN, then the interpretation of the MAC address becomes ambiguous, causing connectivity problems.

3Com has implemented the -PORT ProtMacAddrFmt parameter, which is user-configurable, to address the hardware ambiguity problem for the ARP protocol.

To set the address format, follow these steps:

1. Determine the address format that should be used by each port that terminates a serial line running PPP, Frame Relay, ATM DXI, X.25, or SMDS.

   For example, in the configuration shown in Figure 3-7, you need to determine the MAC address format within the protocol packet that will be used by port 3 of bridge/routers A, B, C, and D.

   For a complete description of the -PORT ProtMacAddrFmt parameter, refer to Chapter 43 in *Reference for NETBuilder Family Software*. This description should help you decide whether the canonical or noncanonical format should be used on a particular port.

2. Set the address format on each port that terminates a serial line running PPP.

   For example, in the configuration shown in Figure 3-7, to set the address format to noncanonical, enter:

   ```
   SETDefault !3 -PORT ProtMacAddrFmt = NonCanonARP
   ```

   in bridges A, B, C, and D.

Optimizing Bridge Performance

To improve the performance of the bridge, follow these steps:

1. Disable the firewall feature in mixed bridging and routing environments by entering:

   ```
   SETDefault -BRidge CONTrol = NoFireWall
   ```

2. If the bridge is performing source route bridging, disable route discovery if the bridge does not need to send source route frames as an end station. Disable route discovery using:

   ```
   SETDefault !<port> -SR RouteDiscovery = None
   ```

   Setting this parameter to None means that the bridge transmits all end system packets as transparent frames, which can reach end systems in a transparent bridged or source route transparent (SRT) bridged environment.

3. Avoid configuring source and destination security features or filters.

4. After the bridge has learned addresses, disable dynamic learning (if you do not need it) by entering:

   ```
   SETDefault -BRidge CONTrol = NoLEarn
   ```

5 If you do not need dynamic learning, increase the aging time for which entries remain in the routing table by using:

```
SETDefault -BRidge AgeTime = <seconds> (10–1000000)
```

The default setting for this parameter is 300 seconds.

How the Bridge Works

This section provides conceptual information on the following topics:

- Transparent bridging
- Translation bridging
- Spanning tree algorithm
- Load sharing
- Routing tables
- Learning and filtering

Transparent Bridging

Transparent bridging is supported on Ethernet, token ring, FDDI, and the following wide area networks: Frame Relay, ATM, X.25, SMDS, PPP, PLG, and ISDN. When transparent bridging is enabled, the bridge forwards packets based on the destination address in the packets it receives. It also learns and records information about the location and addresses of devices on the surrounding networks, based on the source address in the received packets.

You can configure your bridge to forward frames using any of the following methods:

- Transparent bridging only
- Source route bridging only

  The bridge forwards packets based on a route determined by the source or end system from which the packet originated. Because the end system and not the bridge determines the route, a bridge using source route bridging does not record or learn information about addresses on the surrounding networks in the way that a transparent bridge does.
- Transparent and source route bridging simultaneously

  Operating transparent and source route bridging simultaneously is called source route transparent bridging. The bridge automatically determines whether a packet should be forwarded using transparent bridging or source route bridging.

  When configuring parallel bridges, 3Com recommends that you configure both bridges in the same mode, either source route or source route transparent, to prevent unexpected blocking of one type of traffic. For more information on source route bridging, refer to Chapter 5.
- Source route transparent bridging gateway (SRTG)

  You can connect source route domains to transparent bridging domains by configuring SRTG. The SRTG software provides a mapping between the two domains, so that a user on a token ring network using source routing can communicate with another user on an Ethernet network using transparent bridging. For more information, refer to Chapter 5.

IBM-Related Services

IBM-related services such as data link switching (DLSw) and APPN are affected by parameter settings in the BRidge, SR, and LLC2 Services. Table 3-4 lists the required settings in source route (SR), source route transparent (SRT), and transparent (T) bridging environments for each of the IBM-related services.

In this table, the bridging environment (SR, SRT, or T) is shown in the Port Configuration column. Tunneling is the3Com proprietary method of LLC2 tunneling, DLSw is data link switching, and LNM is LAN Net Manager. The settings are shown in abbreviated form. For example, the row labeled DLSw/Tunneling with port configuration SR represents DLSw or 3Com LLC2 tunneling in a source-route-only port configuration. The entries in this row expand to the following software configuration commands:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
SETDefault !<port> -BRidge TransparentBridge = NoTransparentBridge
SETDefault -BRidge CONTrol = Bridge | NoBridge
SETDefault !<port> -SR RouteDiscovery = LLC2
SETDefault !<port> -LLC2 CONTrol = Enable
SETDefault !<port> -SR RingNumber = <number> (1–4095) |
0x<number>(1–FFF)
```

In this configuration, global bridging is enabled or disabled on one or more token ring ports. Transparent bridging is disabled, source routing and route discovery are configured, and LLC2 is enabled.

Table 3-4 IBM-Related Feature Settings

Services	Port Configuration	Source Route Bridging (-SR SRB)	Transparent Bridging (-BR TBR)	Bridging (-BR CONT)	Route Discovery (-SR RD)	LLC2 CONTrol (-LLC2 CONT)	Frame Copy Errors
Bridging only	SR	SRB	NTB	BR	NoLLC2	Disable	None
Bridging only	SRT	SRB	TB	BR	NoLLC2	Disable	*
Bridging only	T	NSRB	TB	BR	NoLLC2	Disable	*
LNM	SR	SRB	NTB	BR	LLC2	Enable	None
DLSw/ Tunneling	SR	SRB	NTB	BR | NBR	LLC2	Enable	None
DLSw/ Tunneling	SRT	SRB	TB	BR	LLC2	Enable	* †
DLSw/ Tunneling	T	NSRB	TB	BR	NoLLC2	Enable	* †
APPN	SR	SRB	NTB	BR | NBR	LLC2	Disable	None
APPN	SRT	SRB	TB	BR | NBR	LLC2	Disable	*
APPN	T	NSRB	TB	BR | NBR	LLC2	Disable	*
Default Setting	SRT	SRB	TB	NBR	NoLLC2	Disable	None

* In this configuration, end systems may generate a small number of MAC Frame Copy error report packets when the bridge/router is initializing or when it ages out a MAC address from its bridge table.

† In this configuration, it is important for global bridging to be enabled, otherwise, the token ring hardware does not filter transparent packets. This can generate many Frame Copy error reports and adversely effect performance.

Token Ring Frame Copy Errors

For transparent bridge (TB) or SRT configurations, token ring end systems may generate a small number of MAC Frame Copy error reports when a NETBuilder II bridge/router is initializing or when the bridge/router ages out a MAC address from its bridge table.

For the bridge/router to learn the MAC addresses of transparent end systems on the token ring, it copies a packet with an unknown source address and sets the address-recognized (A) and frame-copied (C) bits in the Frame Status (FS) field. A problem occurs when the FS (A) and (C) bits have been set and the destination of the frame is an end system on the local ring. The destination end system expects the (A) and (C) bits to be zeros. When it receives a frame with these values already set, it reports an error. The end system counts these errors until the error threshold is reached; then it sends out a MAC Report Error packet.

These Frame Copy errors occur only with transparent token ring packets, because the bridge/router hardware filters source-routed packets based on the route information field, not the MAC address. If the bridge/router is configured for source route only, it never copies frames destined for a station on the local ring. These errors can be avoided by running in source-route-only mode.

Table 3-4 identifies those configurations that can cause Frame Copy errors.

Translation Bridging

With translation bridging, you can communicate between transparent bridging end stations on different LAN media types: Ethernet, token ring, and FDDI. (For source route end stations to communicate with transparent bridging end stations, you must use SRTG as described in Chapter 5.) You also can communicate between end stations on the same media type across backbones of a different media type. The 3Com implementation of translation bridging is based on general principles of media access control (MAC) header translation and encapsulation, as well as protocol-specific translation for well-known protocol problems.

When a packet needs to be forwarded from a token ring or FDDI network to an Ethernet network, translation bridging transforms the packet from a token ring or FDDI format to an Ethernet format, or vice versa. When a packet is forwarded to a serial port, translation bridging takes place automatically at the remote bridge port when it receives the packets. For translation bridging to occur on wide area bridges, translation software is necessary in both units.

Translation bridging between Ethernet and token ring networks connected by a NETBuilder II bridge/router can take place either across serial lines or through a local port. Translation bridging between Ethernet and FDDI networks takes place through local ports. Figure 3-8 illustrates each of these concepts.

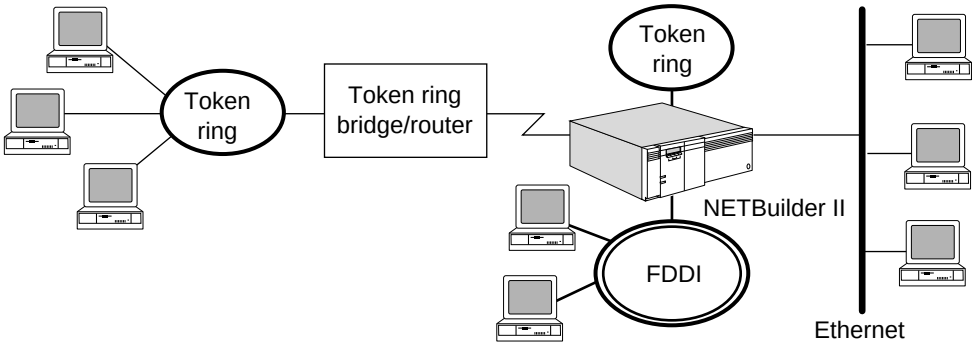


Figure 3-8 Using Translation Bridging to Interconnect Token Ring and Ethernet Networks

Figure 3-9 illustrates the general principles of MAC header translation, as applied to bridging packets of different formats on Ethernet over a token ring or FDDI backbone.

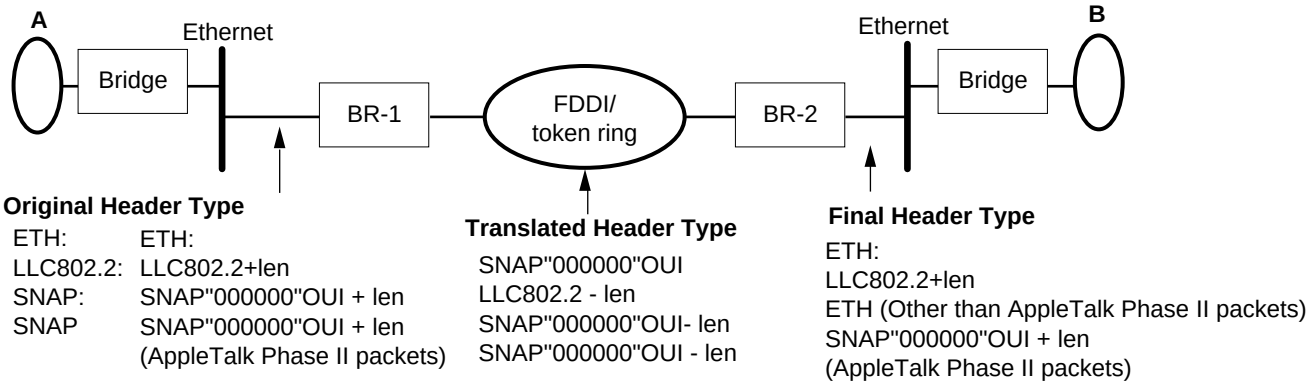


Figure 3-9 MAC Header Translation

OUI Packets

AppleTalk Phase 2 packets originating on an Ethernet network and destined for another Ethernet network across an FDDI backbone remain in AppleTalk Phase 2 Subnetwork Access Protocol (SNAP) format. SNAP packets use an Organizationally Unique Identifier (OUI) of 000000. AppleTalk Phase 1 packets originating on an Ethernet network are tunneled through the FDDI backbone using an OUI value of 0000F8.

For networks other than AppleTalk Phase 2, a SNAP header on Ethernet is translated to a SNAP header on FDDI and then back to Ethernet, instead of SNAP. If you are using translation bridging from Ethernet to Ethernet across FDDI, use the Ethernet header format on both sides.

Some protocols use a format similar to SNAP for encapsulating other types, but use their own proprietary OUI instead of the 000000 OUI used by SNAP. These packets are not converted back to Ethernet when bridging from FDDI or token ring onto an Ethernet LAN.

Maximum Transmission Unit

The maximum transmission unit (MTU) is the maximum packet size allowed, which varies according to the LAN media. Applications that run in a multimedia bridged environment must be configured to use packet sizes that are equal to or smaller than the smallest of the MTU sizes in the extended LAN; otherwise, some media may drop oversize packets. If a particular application cannot accept smaller packets, using network layer routing instead of MAC layer bridging may provide a solution.

For IP packets being bridged between interfaces that have mismatched MTU sizes, you can enable the IP fragmentation feature by setting the -BRidge CONTrol parameter to IPFragment. The bridge then fragments IP packets that are being forwarded to ports with a smaller MTU size.

LLC Length and Packet Size

LLC packets on Ethernet networks contain a length field that is removed before the packet is transmitted to FDDI and token ring media. In some systems, the actual length of the packet and the LLC length field may not match. When these packets are bridged to another Ethernet across an FDDI or token ring backbone, the resulting packet length cannot be determined. The 3Com implementation ignores the actual packet length and transmits according to the LLC length field. If the actual length of the packet is greater than the LLC length, it is cut short to correspond to the LLC length. If it is less than the LLC length, the packet is padded at the end to match the LLC length.

Address Mapping

On Ethernet and FDDI media, multicast addresses are used in the destination address field to reach a group of stations running a certain type of protocol. Because the multicast address is identified by one bit in the address space, it is possible to have millions of such addresses in the available 48-bit address space.

For similar applications on token ring media, functional addresses are used. Only 32 functional addresses are possible. When bridging packets from Ethernet or FDDI to token ring, multicast packets should be mapped to the corresponding functional address and vice versa. Multicast packets that do not have a one-to-one mapping are dropped.

The 3Com implementation maintains a table of multicast-to-functional address mappings for well known protocols, shown in Table 3-5. User-defined mappings can be added using the -BRidge MultiCastAddr or FunctionalAddr parameters. For further information, refer to "Adding Functional-Address-to-Multicast-Address Mappings to the Default Table" on page 3-17.

Table 3-5 Multicast-to-Functional Address Mappings

Type of Packet	Token Ring Functional Address	FDDI or Ethernet Multicast Address
Broadcast	0300FFFFFFFF	FFFFFFFFFFFF
Broadcast	FFFFFFFFFFFF	FFFFFFFFFFFF

By default, the bridge displays addresses in canonical format.

Priority Mapping

Token ring and FDDI media provide a means of prioritizing access over the ring. Applications can request a priority between 0 and 7, and the MAC sublayer maps these user priorities to access priorities supported by the individual media access methods. A token with an access priority equal to or less than the requested user priority transmits this packet over the media.

To prevent a bridge from reordering frames of a given user priority received on one port when forwarding to another port, user priority information is conveyed to the driver along with the frames submitted for transmission. The mapping of user and access priorities is done in accordance with the 802.1d IEEE standard for MAC bridging. For packets that are bridged from Ethernet to token ring, the default user priority of 4 is used.

Configuring Address Format

If you are connecting a 3Com bridge to a bridge from another vendor and are bridging token ring or FDDI packets over a WAN link, you can configure the DatalinkAddrFmt parameter to ensure that the 3Com bridge conforms to standards used by the other bridge.

Protocol-Specific Issues

The following section describes protocol-specific translation bridge issues.

AppleTalk. 3Com has not implemented translation bridging of AppleTalk packets between token ring and other media. Communication between AppleTalk nodes on token ring and nodes on other media types should be accomplished using routing.

Bridging of AppleTalk Phase 1 and Phase 2 packets between Ethernets across an FDDI backbone is implemented according to the recommended practice published by IEEE. This bridging is controlled by the -BRidge APPleTalk parameter. 3Com recommends that you retain the default value of Enable.

IP. MAC-layer bridging typically does not bridge large frames between physical media that have dissimilar maximum frame sizes. To solve this problem, 3Com implements fragmentation of bridged IP packets. IP fragmentation is supported between LAN media and also on WAN media. Fragmentation can be enabled by setting the -BRidge CONTrol parameter to IPFragment.

Fragmentation may cause some deterioration in performance.

IPX. 3Com supports IPX translation bridging between end stations on the same LAN media type across a backbone of another media type. Bridging IPX between end stations on different media types is not supported.

NetWare stations running IPX can be configured to operate in pure Ethernet format, SNAP format, or 802.2 LLC format. Bridging with either Ethernet or 802.3 is uncomplicated. In bridging IPX SNAP format from Ethernet to FDDI or token ring and then back onto Ethernet, SNAP packets are translated back into Ethernet format.

When translation bridging of the IPX Protocol involves an Ethernet backbone, the Novell file server MTU should be configured to be less than or equal to the MTU size of the Ethernet backbone (1,514 bytes).

Spanning Tree Algorithm

The spanning tree algorithm detects loops and puts some bridge ports into blocking state, if necessary, so that only one route exists between any two stations. (A port in blocking state does not forward or receive packets.) Eliminating the extra paths creates a stable network configuration. When one or more bridges or ports in the stable topology fail, the algorithm automatically returns some ports from blocking state to forwarding state to ensure that all stations are connected.

For the spanning tree algorithm to be effective, all bridges in your extended network must run it.

An extended network without loops can be viewed as a spanning tree. A spanning tree is a topology in which one node is designated as the root, and any two nodes are connected to each other through one and only one route. Figure 3-10 is an example of the spanning tree structure in which one bridge represents the root, other bridges represent the nodes, and the communications lines represent the branches. The arrows illustrate the unique path that a packet from the San Francisco bridge takes when destined for the Los Angeles bridge. The topology would not be a spanning tree if there were also a line directly linking the San Francisco bridge and the San Jose bridge, or if the line between the San Jose bridge and the Santa Clara bridge were broken.

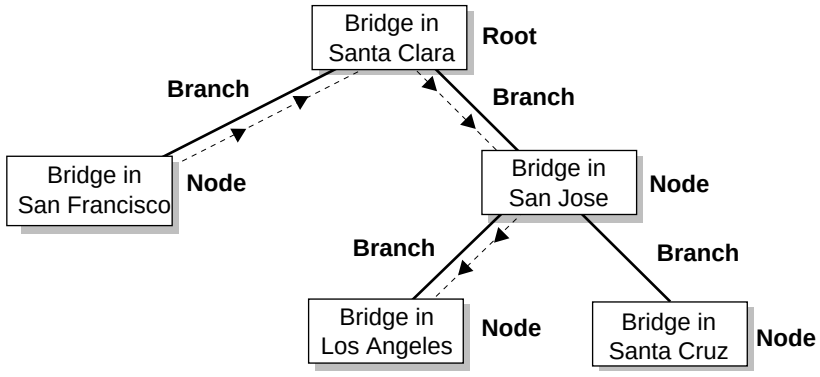


Figure 3-10 Spanning Tree Structure

When more than one bridge connects LANs, the network manager may inadvertently configure the network with loops, causing packets to circulate indefinitely. A loop exists if more than one path can be used to forward a packet from one end station to another. For example, the dotted line in Figure 3-11 highlights a loop; packets from a station on LAN 1 can be forwarded to one or more stations on LAN 2 via either bridge A or bridge B. The destination stations receive duplicate packets because both bridge A and bridge B replicate the packet and then forward the packet to LAN 2. If the station sends out a broadcast packet, both bridges forward it to their attached networks, creating packets that circulate indefinitely.

The spanning tree algorithm detects and breaks loops that can form within a bridging topology.

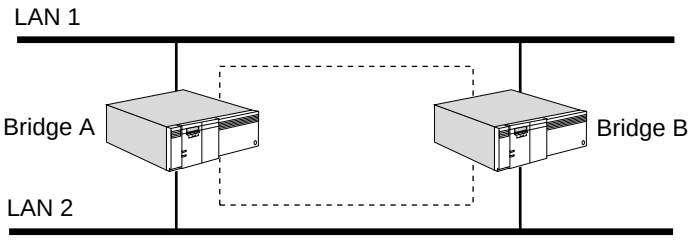


Figure 3-11 Network with a Loop

How the Algorithm Works

The spanning tree algorithm configures the network so that no loops exist in the extended network, and every two LANs can communicate with each other.

This section lists the prerequisites required for the algorithm to work and gives an example to explain how the algorithm arrives at a loop-free configuration.

Algorithm Requirements for Configuring the Network

For the algorithm to configure the network:

- Each bridge must be able to recognize a unique destination address.
- Each bridge must have a unique identifier (bridge ID) that contains a priority field and a data link address.
- Each port of a bridge must have a unique identifier (port ID) that contains a priority field and a port number.
- Each port must be associated with a path cost, which is determined by the speed of its network interface (the faster the speed, the smaller the cost).

How the Algorithm Creates a Loop-free Configuration

To arrive at a loop-free configuration based on the bridge ID, port ID, and path costs, the algorithm performs the following tasks:

- Selects a bridge that acts as the root of the spanning tree network. This is usually the bridge with the lowest bridge ID of all the bridges on the extended network.
- Selects a root port on each bridge (except the root bridge) that incurs the lowest root path cost when the bridge forwards a packet to the root bridge.
- Selects the designated bridge on each LAN that incurs the lowest path cost when forwarding a packet from that LAN to the root bridge. The port through which the designated bridge is attached to the LAN is called the designated port.
- Enables all root ports and designated ports so they can forward packets, and blocks all other ports.

The following example shows how the algorithm makes the selections, then eventually eliminates loops. Figure 3-12, Figure 3-13, Figure 3-14, and Figure 3-15 illustrate an extended network. In these figures, the bridges are numbered from 1 to 5, where bridge 1 has the lowest data link address, and bridge 5 has the highest.

When the bridges are turned on, each assumes that it is the root bridge. Each bridge then transmits a packet called the Configuration Bridge Protocol Data Unit (CBPDU) through all its ports. A CBPDU contains information such as the ID of the bridge that the transmitting bridge considers the root bridge, the root path cost of the transmitting bridge, and the number of the source port.

When a bridge receives a CBPDU that contains superior information on one of its ports, it stores the information at that port. If this CBPDU is received at the root port of the bridge, the bridge also forwards it with an updated message to all attached LANs for which it is the designated bridge.

If a bridge receives a CBPDU on one of its ports that contains information inferior to that currently stored at that port, it discards it. If the bridge is a designated bridge for the LAN from which the CBPDU is received, it sends that LAN a CBPDU containing the up-to-date information stored at that port. In this way, inferior information is discarded and superior information is propagated on the extended network.

Assume that each port in Figure 3-12 is equipped with an Ethernet interface that has a path cost of 100, and that the priority fields in the IDs of bridge 1 and bridge 3are the same. Having the lowest bridge ID (because its data link address is the lowest), bridge 1 becomes the root bridge, and its CBPDU is superior to the ones from other bridges. After exchanging a few CBPDUs and discarding the inferior ones, all bridges contain the same information that indicates that bridge 1 is the root bridge. Because a root bridge is automatically the designated bridge for all LANs to which it is attached, bridge 1 is also the designated bridge for LANs 1 and 2.

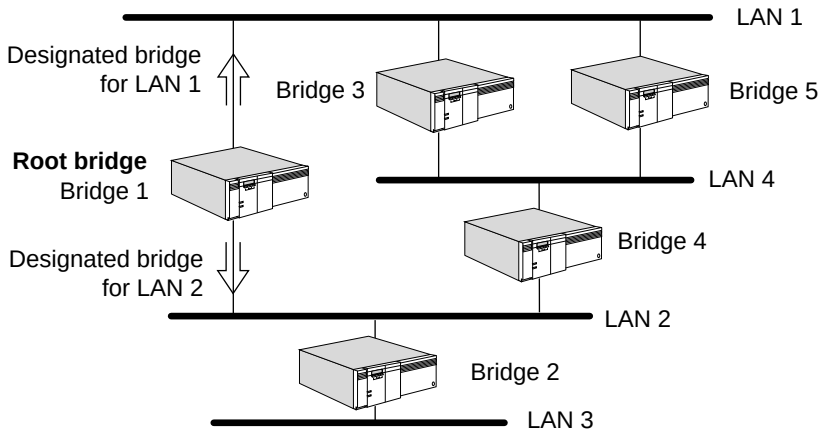


Figure 3-12 Root Bridge

Each bridge (except the root bridge) has to select a root port that will incur the least cost when the bridge forwards a packet to the root. The cost depends partly on the path cost of the port (determined by the speed of its network interface) and partly on the root path cost of the designated bridge for the LAN to which this port is attached.

For example, in Figure 3-13, while ports 1 and 2 of bridge 3 both have the same network interface type and the same path cost, bridge 3 incurs less cost if it forwards a packet from port 1 than from port 2. The algorithm then decides that port 1 should be the root port for bridge 3.

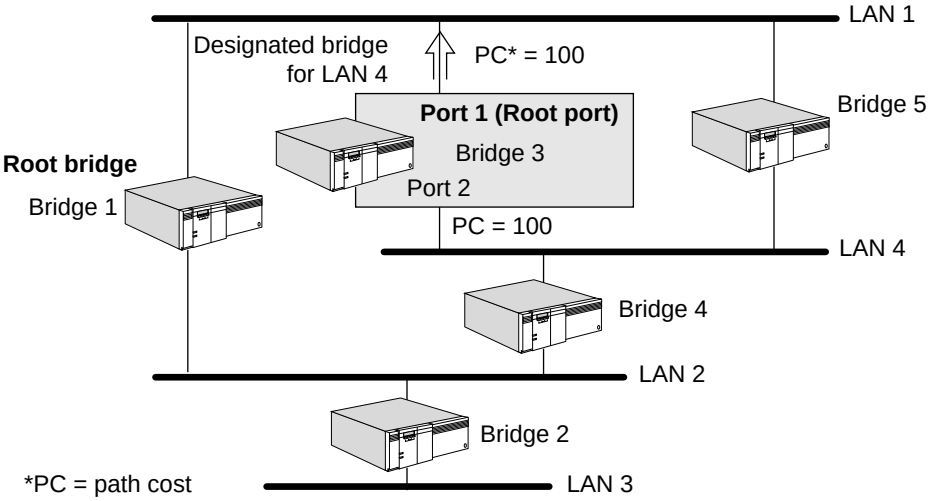


Figure 3-13 Root Port

If a LAN is attached to a single bridge, that bridge is the designated bridge of the LAN. For example, in Figure 3-14, bridge 2 is the designated bridge for LAN 3, because bridge 2 is the only bridge attached to LAN 3.

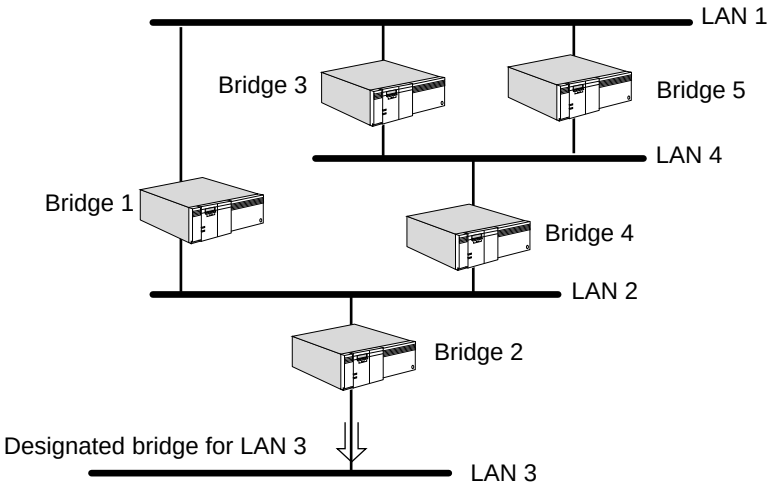


Figure 3-14 Selecting a Designated Bridge when One Bridge Is Attached to a Network

For a LAN that is attached to more than one bridge, a designated bridge must be selected. For example, in Figure 3-15, because LAN 4 is attached to bridge 3, bridge 4, and bridge 5, the algorithm must compare the root path costs of these bridges. In this case, their root path costs are the same. Having the lowest bridge ID, bridge 3 becomes the designated bridge for LAN 4. Because bridge 3 is attached to LAN 4 through port 2, port 2 is the designated port for LAN 4.

Bridge 1, which is the root bridge, is automatically the designated bridge for all attached LANs (that is, LANs 1 and 2). Because bridge 2 is the only bridge attached to LAN 3, it becomes the designated bridge for LAN 3.

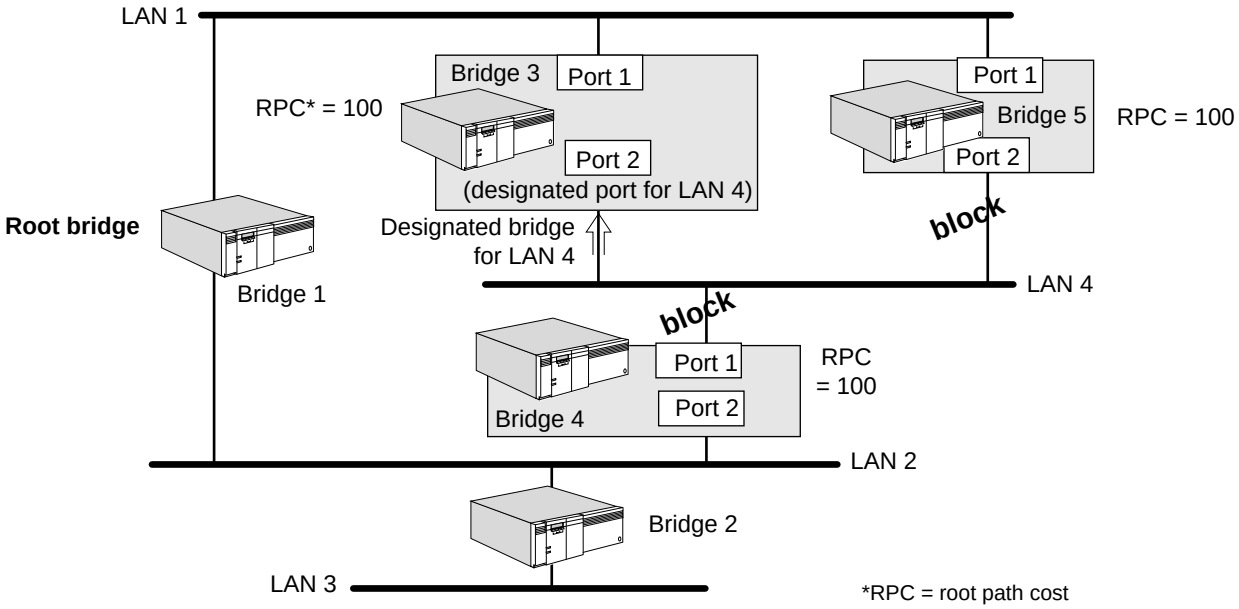


Figure 3-15 Selecting a Designated Bridge when Multiple Bridges Are Attached to a Network

Only root ports and designated ports are put into forwarding state. Other ports, such as port 1 of bridge 4 and port 2 of bridge 5, are put into blocking state, as shown in Figure 3-15.

When a port is in forwarding state, it performs learning, filtering, and forwarding functions. When it is in blocking state, it performs none of these functions.

Because some ports are put into blocking state, none of the packets circulate on the extended network indefinitely.

Using the Algorithm with Wide Area Bridges

Although the examples in the previous section involve only local bridges, local and wide area bridges participate in configuring loop-free networks using the spanning tree algorithm.

In Figure 3-16, two bridges connect two remote networks. On each bridge, one of the interfaces is a network interface, and the others are serial links connected to the other wide area bridge. To apply the spanning tree algorithm in such a network configuration, it is assumed that the serial links are attached to an imaginary backbone network on which no end stations exist. The only traffic on the backbone is the traffic between the bridges. With this assumption, all bridge interfaces operate as if they were network interfaces, and the same spanning tree principle described above applies.

When wide area bridges with parallel lines, as shown in Figure 3-16, participate in the spanning tree algorithm, all remote links connected to the same wide area bridge are considered one network interface. The algorithm puts all links into either forwarding or blocking state. This ensures that the network topology can maximize the use of the bandwidth provided by parallel network links.

If you configure your wide area bridge with parallel lines, make sure that both paths are assigned to the same port. If you use separate ports, the spanning tree algorithm considers each port to be a separate network. As a result, one port will be put into blocking state. You can use parallel lines on different ports as a backup. If the line in the forwarding state fails, the second line moves from the blocking state to the forwarding state.

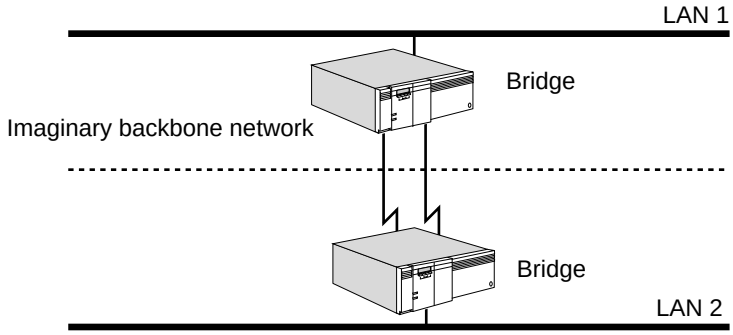


Figure 3-16 Two Wide Area Bridges Connected to Imaginary Backbone Network

Configuring the Spanning Tree Protocol over PPP

When you connect two bridges over a PPP serial link, both bridges must operate in the same spanning tree domain. 3Com supports the following configurations of the STP over PPP:

- Source route to source route
- Source route transparent to source route transparent
- Transparent bridge to transparent bridge
- Transparent bridge to source route transparent

The following configurations are not supported:

- Source route to transparent bridge
- Source route to source route transparent

If you connect bridges in the unsupported configurations, the separate spanning tree domains are combined into a single domain.

When two bridges are connected over a PPP serial link, both bridges must be operating in the same spanning tree domain (SR or TB/SRT). The following configurations are supported:

- SR–SR
- SRT–SRT
- TB–TB
- TB–SRT

The following configurations are not supported:

- SR–TB
- SR–SRT

If you connect bridges in the unsupported configurations, the separate SR and SRT/TB spanning tree domains will combine into a single spanning tree domain.

A bridge is configured for SRT, SR, or TB modes as follows:

- SRT

  One or more ports are configured for transparent bridging and one or more ports are configured for source route bridging.

- SR

  One or more ports are configured for source route bridging and no ports are configured for transparent bridging.

- TB

  One or more ports are configured for transparent bridging and no ports are configured for source route bridging.

Configure ports for transparent bridging by setting the TransparentBRidge parameter in the BRidge Service. Configure ports for source route bridging by setting the SrcRouBridge parameter in the SR Service.

Spanning Tree Addressing

Transparent and source route transparent bridges participate in a spanning tree domain, which is identified when the destination address field of the spanning tree packet is the hexadecimal group address 0180C2000000. Source route bridges participate in a different spanning tree domain, which is identified when the destination address field of the spanning tree packet is the hexadecimal bridge functional address 030000008000. Both addresses are shown in canonical addressing format.

If a bridge has different types of bridging enabled on different ports, the spanning tree algorithm determines what type of bridge it is overall (transparent, source route, or source route transparent) according to the following criteria:

- If a bridge does not have transparent bridging enabled on any ports and has source route bridging enabled on at least one port, it is considered a source route bridge.
- If a bridge has transparent bridging enabled on at least one port and source route bridging enabled on at least one port, it is considered a source route transparent bridge.
- If a bridge does not have source route bridging enabled on any ports, it is considered a transparent bridge.

The spanning tree algorithm detects loops independent of the operating mode of the bridge.

Modifying Spanning Tree Parameters

The Spanning Tree Protocol (STP) Service controls parameters used by the spanning tree algorithm (for example, the priority field in the bridge identifier) to influence the final network configuration. For more information on setting STP parameters, refer to Chapter 57 in *Reference for NETBuilder Family Software*.

Reconfiguring the Topology

The spanning tree algorithm reconfigures the network topology when bridges are added or removed, or when the network manager changes the parameters.

Whenever a bridge detects a topology change, if it is a designated bridge for a LAN, it sends out a topology change notification Bridge Protocol Data Unit (BPDU) through its root port. This information is eventually relayed to the root bridge. The root bridge then sets the topology change flag in its CBPDU so that the information is broadcast to all bridges. It transmits this CBPDU for a fixed amount of time to ensure that all bridges are informed of the topology change.

If a port is changed from blocking state to forwarding state as a result of the topology change, the algorithm ensures that it propagates the topology information to all ports before that port starts forwarding data. This prevents temporary data loops.

If a bridge does not receive packets from an address within a fixed period of time, it removes that address from its routing table. After reconfiguration, the bridge removes these addresses faster to ensure that each active port still forwards packets to the right network after a topology change.

Load Sharing

When multiple paths are assigned to a port on a NETBuilder II bridge/router, a load-sharing algorithm is used. The load-sharing algorithm selects the highest bandwidth line as the primary line. Any outgoing data is transmitted through this line until a certain threshold (defined within software limits for that bandwidth) is reached. When the threshold is reached, packets are forwarded on the next highest bandwidth line. If the number of bytes queued on the primary line falls below the threshold, outgoing packets revert to the primary line.

Routing Tables

A bridge forwards packets according to information in the routing table. Each entry in this table lists an address, the network on which the station with that address can be found, and an indication of elapsed time since a packet was received from that node. For an interpretation of the routing table, refer to Chapter 14 in *Reference for NETBuilder Family Software*.

The two types of routing table entries are: learned (dynamic) entries and user-assigned (static or permanent) entries.

- Learned entries are entries that the bridge learns from the network. The learned entries are subject to dynamic changes or deletion whenever the -BRidge CONTrol parameter is set to Aging and LEarn.
- User-assigned entries are entries assigned by entering ADD -BRidge ROUte. The user-assigned entries can be changed or deleted manually only through the ADD or DELete commands.

You can access the routing table of transparent bridges by entering:

```
SHow -BRidge AllRoutes
```

For complete information on this parameter, refer to Chapter 14 in *Reference for NETBuilder Family Software*.

You can configure the size of the routing table on the transparent bridge using the -BRidge RouteTableSize parameter. For complete information on this parameter, refer to Chapter 14 in *Reference for NETBuilder Family Software*.

Learning and Filtering

This section describes how a bridge learns the network configuration and adapts to the addition or removal of stations on the attached network segment in order to perform standard filtering. For information on 3Com mnemonic filtering and related filtering processes such as logging, sequencing, and packet prioritization, refer to Chapter 4. For complete explanations of packet filtering parameters, refer to Chapter 23 in *Reference for NETBuilder Family Software.*

Figure 3-17 shows two networks interconnected by a bridge. After the bridge receives a packet, it decides whether to forward it to the other network or discard it. To help make this decision, the bridge determines to which network the destination of the packet belongs.

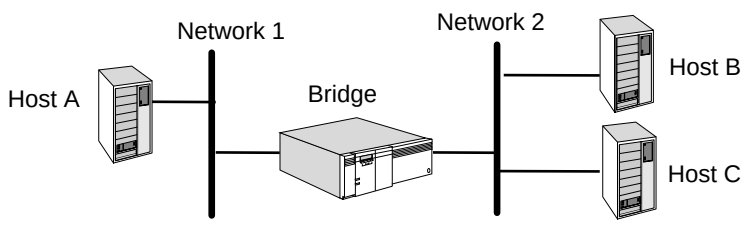


Figure 3-17 Bridge Learning

When a bridge is operating, it receives packets from all attached networks. By looking at the source address of packets, the bridge learns the addresses of stations on each network and stores them in its routing table. For example, when the bridge in Figure 3-17 receives a packet from network 1 with the address for host A as the source address, it learns that host A is on network 1. In the same way, it also learns that hosts B and C are on network 2.

If a packet is destined for the network where it originated, the bridge discards it. This is called standard filtering. For example, if the bridge in Figure 3-17 receives a packet on network 2 from host C that is addressed to host B, and it determines from the learned entries in its routing table that host B is on the same network as host C, then it discards the packet.

In addition, the bridge uses the learned network configurations to forward packets destined for another network. For example, if the bridge receives a packet on network 2 from host C addressed to host A, it determines that host A is on another attached network, and forwards the packet to that network.

If the bridge receives a packet from a host on a network that has not yet been learned, the bridge forwards the packet to all ports except the port on which the packet was received.

The bridge also can learn that a station has been removed from one of its attached networks. For example, in Figure 3-18, host C was moved from network 2 to network 1. The bridge no longer receives packets on network 2 with host C as the source address. The bridge record of the location of host C is no longer updated and is removed (aged) from the routing table. With host C attached to network 1, the bridge receives packets from network 1 with the address of host C as the source address, and learns that network 1 now includes host C.

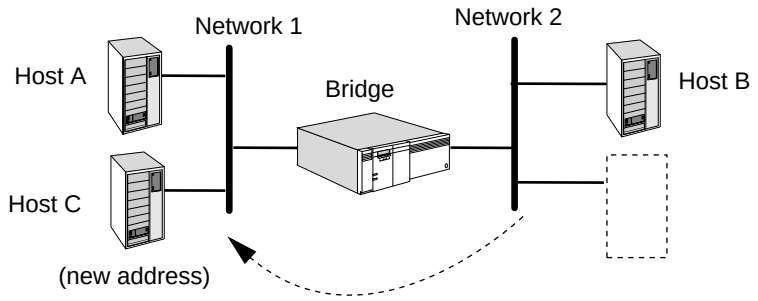


Figure 3-18 Network Configuration after Host C Is Moved

4 CONFIGURING MNEMONIC FILTERING

This chapter describes the procedures for configuring filters and also lists all the built-in masks for the bridge and Internetwork Packet Exchange (IPX) router. Filtering is an operation that determines whether specified packets are forwarded or discarded by your 3Com bridge or IPX router. The FIlter Service also controls these and other capabilities through the FIlter POLicy parameter action options: Count, Discard, DodDiscard, Forward, PROTocolRsrv <tag>, Sequence, Prioritization, and Trace. These action options are described in "Action" on page 4-6.

You need to configure prioritization separately. For complete information on the prioritization allocation, refer to Chapter 41.

By using filtering in a bridged or IPX routed environment, you can:

- Achieve security and bandwidth protection by isolating specific segments of the network.
- Monitor network traffic by gathering statistics.
- Adjust the performance of your network to fit the traffic flow.
- Sequence packets so that they are received in the order they were sent.
- Reserve bandwidth for particular protocols, so that large-bandwidth user applications, such as file transfer and mail, share link capacity with lower bandwidth users such as interactive sessions and transaction-oriented applications.

The NETBuilder software includes the use of mnemonics and built-in masks for specific protocols in the configuration of filters. Through the use of built-in mnemonics, you can also create user-defined masks to meet more specialized needs.

For more information on the parameters used in creating filters and masks, refer to Chapter 23 in *Reference for NETBuilder Family Software*. For conceptual information, refer to "How Filtering Works" on page 4-5.

Configuring Filters

When you configure filters, you perform the selection, qualification, and action steps using the ADD and DELete commands. The MASK parameter specifies the selection criteria and the POLicy parameter specifies the context by qualifying the selection and associating the action.

You can use the same selection criteria (masks) in different contexts (policies). You can also combine different selection criteria while qualifying them and specifying the action. The procedures in this section use the minimum number of steps required to configure basic built-in and user-defined filters for the bridge.

Using Built-in Masks

To configure filters for the bridge or IPX router using built-in masks, follow these steps:

1. Determine whether or not a built-in mask can be used as follows:
   a. Identify the type of packet to be filtered.
   b. After identifying the packet type, refer to Table 4-2 on page 4-8 for the BRidge Service, or Table 4-3 on page 4-9 for the IPX Service, or Table 4-4 on page 4-9 for IBM Trace built-in masks.

      These tables identify all types of packets for which built-in masks can be used.

      If a built-in mask can be used, proceed to step 2. If a built-in mask cannot be used, follow the steps in "Using User-defined Masks"next.
2. Define the policy by using the ADD POLicy command.

   Add a policy whether or not the mask is built-in.

   For example, suppose you want to discard all Internet Protocol (IP) multicast packets at port 2. To define the policy, enter:

   ```
   ADD -FIlter POLicy NoIPMC Discard IP MC AT !2
   ```

   The following message appears on the screen:

   ```
   Policy NoIPMC is added
   ```

   Continue using the ADD MASK and ADD POLicy commands for all types of packets to be filtered.
3. Specify the action for packets that do not match any policy by setting the DefaultAction parameter:

   ```
   SETDefault -FIlter DefaultAction = [Forward | Discard]
   ```

   When DefaultAction is set to Discard, all packets not matching a policy are discarded. All packets matching the policy are handled according to the policy.
4. Enable filtering by entering:

   ```
   SETDefault -FIlter CONTrol = Enabled
   ```

Using User-defined Masks

To configure a filter using user-defined masks, follow these steps:

1. Determine whether or not a built-in mask can be used as follows:
   a. Identify the type of packet to be filtered.
   b. After identifying the packet type, refer to Table 4-2 on page 4-8 for the BRidge Service or Table 4-3 on page 4-9 for the IPX Service.

      These tables identify all types of packets for which built-in masks can be used. If a built-in mask can be applied, follow the steps in "Using Built-in Masks"on page 4-2. If a built-in mask cannot be applied, proceed to step 2.
2. If a built-in mask cannot be used, and built-in mnemonics is supported, define your own mask by using the ADD MASK command.

   Table 4-5 on page 4-10 and Table 4-6 on page 4-11 list the built-in mnemonics that can be used to construct user-defined masks for the BRidge and IPX Services.

Suppose you want to define a pattern for a mask that is not built-in (that is, not represented in Table 4-2 or Table 4-3). For example, you may want to discard all packets that are longer than 512 bytes. Because you cannot represent this pattern as a built-in mask, you must enter the following command and the built-in mnemonics (dl.length) to define the mask:

```
ADD -FIlter MASK longpkts dl.length>%0200
```

The following message appears on the screen:

```
Mask LONGPKTS is added
```

The expected value must be an even number of digits.

3 Define the policy by using the ADD POLicy command.

Add a policy whether or not the mask is built-in. For example, suppose you still want to discard all packets that are longer than 512 bytes at port 2, as in step 2. You have defined the mask. To define the policy, enter:

```
ADD -FIlter POLicy toolong Discard longpkts AT !2
```

The following message appears on the screen:

```
Policy TOOLONG is added
```

4 Specify the action of the packet that does not match any policy by setting the DefaultAction parameter using:

```
SETDefault -FIlter DefaultAction = [Forward | Discard]
```

When DefaultAction is set to Discard, all packets that do not match a policy are discarded. All packets that match the policy are handled as designated in the policy.

5 Enable filtering by entering:

```
SETDefault -FIlter CONTrol = Enabled
```

Grouping Related Stations

To configure a filter for a group of logically related stations, use the StationGroup parameter. When using the StationGroup parameter, you need to complete the following tasks:

- Assign a set of station addresses for easy reference.
- Give the group a name.
- Create a mask by referencing the station group name.

Example Figure 4-1 is an example of specifying a policy based on station groups.

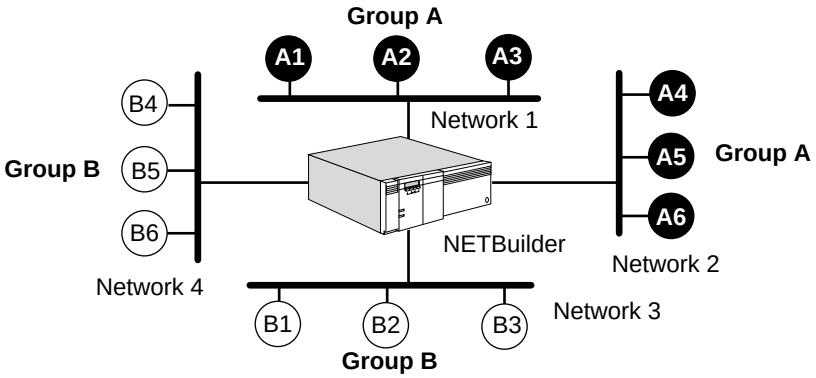


Figure 4-1 Network Showing Station Groups

In this figure, stations belong to group A or to group B. Group A has stations on network 1 and network 2. Group B has stations on network 3 and network 4. After grouping the stations, you can create a policy that would, for example, prohibit a certain type of traffic between group A and group B. Assuming that the media access control (MAC) address for station A1 is %0800020000a1 and the MAC address for station A2 is %0800020000a2, follow these steps to configure a filter between group A and group B:

1. Define a station group and add the MAC addresses of the stations belonging to the defined group.

   For example, create group A and group B, and add appropriate addresses to them by entering:

```
ADD -FIlter StationGroup group_a %0800020000a1
ADD -FIlter StationGroup group_a %0800020000a2
ADD -FIlter StationGroup group_a %0800020000a3
ADD -FIlter StationGroup group_a %0800020000a4
ADD -FIlter StationGroup group_a %0800020000a5
ADD -FIlter StationGroup group_a %0800020000a6
ADD -FIlter StationGroup group_b %0800020000b1
ADD -FIlter StationGroup group_b %0800020000b2
ADD -FIlter StationGroup group_b %0800020000b3
ADD -FIlter StationGroup group_b %0800020000b4
ADD -FIlter StationGroup group_b %0800020000b5
ADD -FIlter StationGroup group_b %0800020000b6
```

2. Define masks using the station groups.

   For example, to create masks, enter:

```
ADD -FIlter MASK from_group_a DataLink.SrcAddr = group_a
ADD -FIlter MASK from_group_b DataLink.SrcAddr = group_b
ADD -FIlter MASK to_group_a DataLink.DstAddr = group_a
ADD -FIlter MASK to_group_b DataLink.DstAddr = group_b
```

3. Define policies using the previously defined masks.

   For example, to create policies, enter:

```
ADD -FI POLicy block_from_a Discard from_group_a, to_group_b, IP
ADD -FI POLicy block_from_b Discard from_group_b, to_group_a, IP
```

   For more information on the StationGroup parameter, refer to Chapter 23 in *Reference for NETBuilder Family Software*.

Parameter Overview

Table 4-1 lists and briefly describes the FIlter Service parameters. For detailed descriptions of these parameters, refer to Chapter 23 in *Reference for NETBuilder Family Software*.

Table 4-1 FIlter Service Parameters

Parameter	Description
CONFiguration	Displays the overall configuration of the FIlter Service.
CONTrol	Disables and enables the FIlter Service. Must be enabled for any filter-related actions to be performed.
DefaultAction	Specifies the action applied to a packet if it does not match any of the policies configured. (If default is altered to Discard, and there are no forwarding policies defined, no packets are forwarded by the system.)

(continued)

Table 4-1 FIlter Service Parameters (continued)

Parameter	Description
DIAGnostics	Shows the current decision tree that the system is using. Shows which MASKs are associated with which POLicies.
MASK	Defines the criteria used to select a packet for special handling.
MNEmonics	Displays all possible options for a location that can be used to construct a user-defined mask.
POLicy	Defines the system context within which the specified masks are applied and the action to be taken. Uses the MASKs that are defined, and applies specific operations to packets that match the MASK conditions of the POLicy.
SELection	Lists all services for which the filter function can be invoked (BRidge, IPX, DLSW, LLC2 or SDLC).
StationGroup	Groups a set of station addresses for easy reference.

How Filtering Works

This section explains the filtering process.

A filter contains the following two components:

- A mask, which defines the qualifications a packet must meet
- A policy, which defines which masks are to be applied and what action is to be taken for the packets that meet the criteria of the mask

For packets using filters based on either user-defined masks or built-in mnemonic masks, the following FIlter Service POLicy parameter action options are available: Count, Discard, DodDiscard, Forward, PROTocolRsrv <tag>, Sequence, Prioritization, and Trace.

When you use filters with user-defined masks, you need to determine location offsets and values to create the mask. Using built-in masks allows you to specify packet selection criteria without determining specific offsets, encapsulation, and frame formats. These built-in masks simplify filtering operations for the bridge and make filtering configurations transferable across interfaces of different types. Most built-in masks are defined for specific protocols. These masks are listed later in this chapter.

To support user-defined masks, NETBuilder software has several built-in mnemonics that can be used to specify location and pattern. The locations and patterns are listed later in this chapter.

The filtering operation involves the steps of selection, qualification, and action.

Selection

Selection identifies the packets on which filtering is performed. You can select packets for special action by specifying a particular pattern of data at a particular location. You can also specify other, more complicated, selection criteria. Use the MASK parameter to select the packet.

Figure 4-2 is an example of the use of MASK parameter in the selection process. The location is typically specified as a string of hexadecimal numbers. In Figure 4-2, the use of built-in BRidge mnemonics lets you specify the location at the DataLink layer. The offset for the same field within a packet can vary, depending on the encapsulation or frame format. For more information on the MASK parameter, refer to Chapter 23 in *Reference for NETBuilder Family Software.*

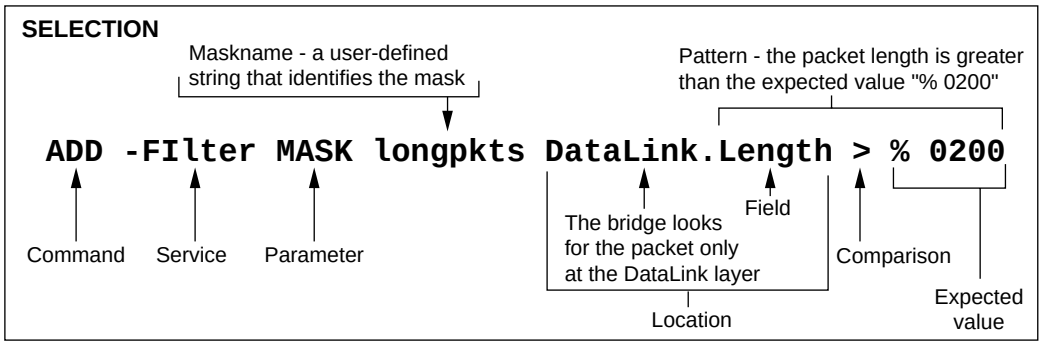


Figure 4-2 Filter Selection Process

Qualification Qualification specifies the context of the filtering operation, that is, the direction of travel and the ports affected.

After selecting a packet for special action using the MASK parameter, you may specify additional qualifications before the action is taken. For example, using NETBuilder software, it is possible to select only those broadcast packets that arrive on a specified port, instead of all broadcast packets. Use the POLicy parameter to specify qualifications for the packet.

Action After the packet is selected and qualified, a specified action occurs. Use the POLicy parameter in the FIlter Service to specify the desired action. The action options supported in the NETBuilder software are Count, Discard, DodDiscard, Forward, PROTocolRsrv <tag>, Sequence, Prioritization, and Trace. Figure 4-3 illustrates the qualification and action processes using an example of the POLicy parameter.

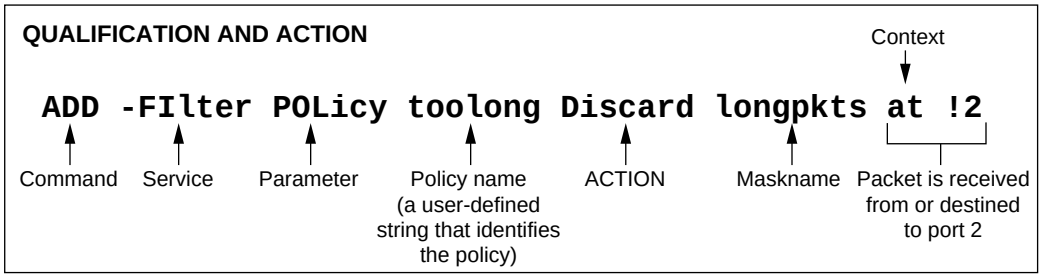


Figure 4-3 Filter Qualification and Action Processes

Count

When you use the Count option, you count packets that meet specified criteria. For example, you may want to count all IP packets forwarded by the bridge before deciding how the bridge should handle them. To perform this operation, enter:

```
ADD -FIlter POLicy IP_count Count ip
```

Discard

When you use the Discard option, you can discard packets that match specific criteria.

DodDiscard

When you use the DodDiscard option for a dial-on-demand (DOD) port, if the dial-up path is down, you can ensure that the packet is discarded and does not cause the dial-up path to be raised. If the path is up, the packet is forwarded, but is not considered as user traffic that keeps a dial-up path up.

Forward

Filters can prevent packets meeting certain criteria from being forwarded across the system or forward only those packets meeting specified criteria while blocking all others. When you use the Forward option, you forward packets that match specific criteria. For more information on forwarding, refer to the POLicy parameter in Chapter 23 in *Reference for NETBuilder Family Software*.

PROTocolRsrv <tag>

Protocol reservation assigns a specified percentage of bandwidth to designated packets passing through a specified port and meeting specified conditions. The specified conditions can be protocol type, packet length, packets destined for specified address, and so on.

Protocol reservation is set up with different procedures depending on the packet types being configured for protocol reservation. The mnemonic filtering procedure applies to all bridged packets and all IPX-routed packets. The IP fIltering procedure applies only to IP-routed packets. IP-routed packets are also filtered using the IP firewall feature. Refer to Chapter 7 for detailed information about the IP firewall feature.

For a detailed description of the protocol reservation procedures for all the packet types, refer to Chapter 38.

As part of the mnemonic filtering procedure, you enter the PROTocolRsrv <tag> action option to apply protocol reservation to designated packets. The tag name identifies those packets that receive a specified percentage of bandwidth when passing through the specified WAN port and when meeting the mask conditions set up with the FIlter Service POLicy parameter. Tag the designated packets with the identifying name by entering a name as the <tag> value when you enter the PROTocolRsrv <tag> action option. The tag name can be any alphanumeric string no longer than 15 characters.

For bridge filtering examples using the PROTocolRsrv <tag> action option and the -PORT PROTocolRsrv parameter, refer to example 26 on page 4-17, example 27 on page 4-18, and example 28 on page 4-18 in "Bridge Filtering Examples."

For an IPX filtering example using the PROTocolRsrv <tag> action option and the -PORT PROTocolRsrv parameter, refer to example 9 on page 4-24 in "IPX Filtering Examples."

Sequence

You can sequence packets to ensure that they arrive at their destination in the order they were sent. To ensure that packets arrive in sequence, use the Sequence option. When the load-balancing algorithm is operating, packets can arrive out of sequence.

When operating with two or more parallel lines (including bandwidth-on-demand dial-up lines), local area transport (LAT), NETBEUI, and Logical Link Control type 2 (LLC2) should be packet-sequenced using the sequence policies. If all of the traffic on the port is sequenced, bandwidth-on-demand is not used for that data. Sequenced traffic is only sent on the primary path.

For example, if you want to sequence and send LAT packets to port 4 in the order they are received, enter:

```
ADD -FIlter POLicy LATorder Sequence LAT TO !4
```

For more information on sequencing and the POLicy parameter, refer to Chapter 23 in *Reference for NETBuilder Family Software.*

Prioritization (Priority Queuing)

The Prioritization option allows you to prioritize different packet types transmitted over wide area networks. You can assign priorities to packets according to their protocol type. Prioritization is a filtering component and needs to be configured separately. For complete information on data prioritization, refer to Chapter 41.

Trace

You can trace packets from IBM-related protocols such as APPN, DLSw, LLC2, and SDLC. You can use these traces to determine the status of connections and to isolate problems. The Trace option cannot be used for any other type of packet.

For a more detailed explanation of the -FIlter MASK and -FIlter POLicy parameters, refer to Chapter 23 in *Reference for NETBuilder Family Software.*

Built-in Bridge Masks

NETBuilder software supports several built-in predefined selection criteria, or masks. All bridge masks are associated with DataLink level as the protocol, and all IPX built-in masks are associated with IPX as the protocol. Table 4-2 lists the built-in DataLink masks. To display this table, enter:

```
SHow -FIlter MASK BuiltIn
```

Table 4-2 Built-in Bridge Masks

Built-in Mask	Equivalent	Packet Type
BC	DataLink.DestinationAddr=BroadCast	Bcast
MC	DataLink.DestinationAddr=MultiCast	Mcast
ATALK	DataLink.Protocol=AppleTalk	AT
AARP	DataLink.Protocol=AARP	AppleTalkARP
ARP	DataLink.Protocol=ARP	ARP
CLNP	DataLink.Protocol=CLNP	OSI-related
DECNET	DataLink.Protocol=DECnet	DECnet
DLTEST	DataLink.Protocol=DLTest	DLTest
IP	DataLink.Protocol=IP	IP
IPX	DataLink.Protocol=IPX	Novell IPX

(continued)

Table 4-2 Built-in Bridge Masks (continued)

LAT	DataLink.Protocol=LAT	LAT
NMIP	DataLink.Protocol=NetMapIP	NetMapIP
NMXNS	DataLink.Protocol=NetMapXNS	NetMapXNS
STP	DataLink.Protocol=STP	Spanning Tree
VIP	DataLink.Protocol=VIP	VINES
XNS	DataLink.Protocol=XNS	XNS
SR	DataLink.RoutingType=SpecificRoute	Specifically Routed Frame
SRE	DataLink.RoutingType=SingleRouteExplorer	Spanning Tree Explore
ARE	DataLink.RoutingType=AllRouteExplorer	All Route Explore
ALLRT	DataLink.RoutingType=ALL	Any source-routed frame

Built-in IPX Masks

Table 4-3 lists the built-in IPX masks. These predefined masks identify different types of IPX packets. To display this table, enter:

```
SHow -FIlter MASK Builtin
```

Table 4-3 Built-in IPX Masks

Built-in Mask	Use
IPXRIP	Matches a RIP packet.
SAP	Matches a SAP packet.
FSP	Matches a Netware File Service NCP packet.
WANBC	Matches a broadcast packet of IPX packet type 20.
TRACERT	Matches a 3Com-proprietary Trace packet (soc = 0x874e).
IPXPING	Matches an IPX Ping packet (soc = 0x9086).
IPXDIAG	Matches an IPX Diagnostic packet (soc = 0x456).
NWSEC	Matches a Netware Security packet (soc = 0x457).

Built-in IBM Trace Masks

Table 4-4 lists the built-in IBM Trace masks. For more information about using the IBM Trace facility, refer to Appendix O.

Table 4-4 Built-in IBM Trace Masks

Built-in Mask	Equivalent	Packet Type
LLC2	Datalink.Protocol=LLC2	LLC2
SDLC	Datalink.Protocol=SDLC	SDLC
DLSW	Datalink.Protocl=DLSW	DLSW
DLSCTL	DLSW.1 = 72	DLSW Control Message
DLSWI	DLSW.1 = 16	DLSW Information Message

User-defined Bridge Masks

When you use the ADD MASK command, you must specify a location. The location is usually expressed as a hexadecimal value representing the offset from the beginning of a packet at which a specified pattern of data is compared to the contents of a packet. The packet is selected if it matches the pattern of data at the specified location.

You also can specify a location in the mnemonic form: <protocol>.<field>. This format allows encapsulation-independent relative offsets to be used. You do not need to determine frame formats or specific offsets. All bridge mnemonics are associated with DataLink as <protocol>. Different mnemonic values are allowed for the <field> and <match> locations. To support IPX filtering, a set of IPX-specific mnemonics is provided. All IPX mnemonics are associated with IPX as <protocol>.

Table 4-5 shows valid locations that match the DataLink protocol. Use these fields to specify an address, instead of specifying the offset of a particular field.

To display a list of valid locations supported by the bridge, enter the SHow -FIlter MNEmonics command. ALL is a valid match mnemonic for certain field categories. When ALL is specified, any value in the location is considered to match the criteria. Field mnemonics indicate encapsulation-independent relative offset. The software recognizes the encapsulation and locates the <field> at the correct offset.

Table 4-5 User-defined Bridge Masks and DataLink Locations

Field	Description	Matching Value
DstAddr	Destination Address at DataLink layer	<MAC address> ALL <StationGroup>
SrcAddr	Source Address at DataLink layer	<MAC address> <StationGroup>
Address	Either Destination or Source Address at DataLink layer	<MAC address> <StationGroup>
Protocol	Packet protocol type	<numerical value>
LENgth	Frame size, including padding	<numerical value>
DSAP	Destination service access point	<numerical value>
SSAP	Source service access point	<numerical value>
LSAP	Link service access point, destination or source SAP	<numerical value>
OUI	Organizationally unique ID	%<hexadecimal number>
LanID	LAN identifier in a source-routed frame	<numerical value>
DATA+[%]<offset>[:[%]<length>]	Offset from start of DataLink data	%<hexadecimal number> <numerical value>
[%]<offset>[:[%]<length>]	Offset from start of DataLink header	%<hexadecimal number> <numerical value>

The SR bit in the SourceAddress field of a source-routed frame is ignored during comparison.

User-defined IPX Masks

Table 4-6 lists user-defined IPX masks and valid locations. You can use these fields to specify an address, instead of specifying the offset of a particular field. ALL is a valid match mnemonic for certain field categories. When ALL is specified, any value in the location is considered to match the criteria. The % sign is used to enter hexadecimal values.

To display a list of valid locations supported by the Internetwork Packet Exchange (IPX) router, enter:

```
SHow -FIlter MNEmonics
```

Table 4-6 IPX Built-in Mnemonics for User-defined Masks

Field	Description	Matching Value
DsrNETwork	IPX destination network	<network number>
SrcNETwork	IPX source network	<network number>
NETwork	Either IPX destination or source network	<network number>
DstNodeAddr	IPX destination node address	%<host address>
SrcNodeAddr	IPX source node address	%<host address>
NodeAddr	Either IPX destination or source node address	%<host address>
DstSocKeT	IPX destination socket	FileServicePacket ServiceAdvertisingPacket RoutingInformationPacket IpxPingPacket IpxDiagPacket IpxTraceRoute NWSecurityPacket %<hexadecimal value> <numerical>
SrctSocKeT	IPX source socket	FileServicePacket ServiceAdvertisingPacket RoutingInformationPacket IpxPingPacket IpxDiagPacket IpxTraceRoute NWSecurityPacket %<hexadecimal value> <numerical>
SocKeT	Either IPX destination or source socket	FileServicePacket ServiceAdvertisingPacket RoutingInformationPacket IpxPingPacket IpxDiagPacket IpxTraceRoute NWSecurityPacket %<hexadecimal value> <numerical value>
PacketLength	IPX packet length	%<hexadecimal value> <numerical>
PacketType	IPX packet type	%<hexadecimal value> <numerical>
TransportCtl	IPX transport control	%<hexadecimal value> <numerical>
DATA+[%]<offset> [:[%]<length>]	Starting <offset> bytes after the end of the IPX header and <length> bytes long	%<hex num string> <“ascii string”>

Bridge Filtering Examples

This section contains examples of bridge filtering features. Examples of configuring the prioritization component of filtering are provided in Chapter 41.

Example 1 To enable filtering and to stop checking policies after a policy that matches the packet is found, use:

```
SETDefault -FIlter CONTrol = (Enabled, MatchOne)
```

Example 2 **Displaying all masks.** To display all masks, enter:

```
SHow -FIlter MASK
```

Displaying built-in masks. To display all built-in masks, enter:

```
SHow -FIlter MASK BuiltIn
```

Displaying a specific mask. To display a specific mask, use:

```
SHow -FIlter <maskname>
```

Example 3 **Displaying all policies.** To display all policies, enter:

```
SHow -FIlter POLicy
```

Displaying a specific policy. To display a specific policy, use:

```
SHow -FIlter <policyname>
```

Example 4 To discard all source-routed IP packets, enter:

```
ADD -FIlter POLicy dissr_ip Discard ip, allrt
```

Example 5 This example describes how to discard all IP packets from port 1 to port 2 using two options: the command syntax and the menu (see Figure 4-4). IP packets are selected for special action. The selection is further qualified by specifying from port 1 to port 2. The action is designated as discard. Because built-in masks are defined for IP packets, it is not necessary to use the ADD MASK command.

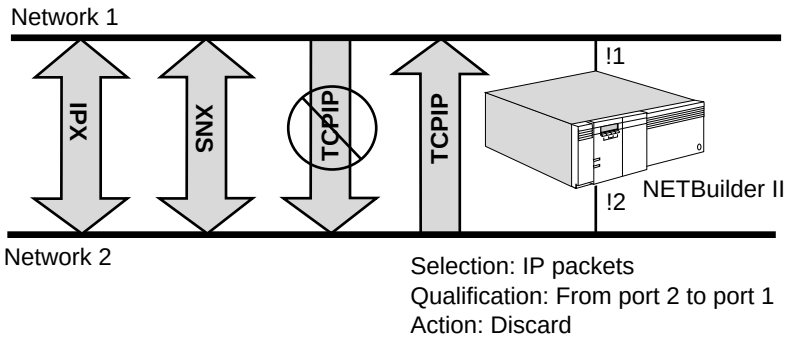


Figure 4-4 Discarding IP Packets

Command Syntax Option. Define the policy by entering:

```
ADD -FIlter POLicy noip Discard ip FROM !1 TO !2
```

Menu Option. You can use the FIlter Service menu to discard all IP packets from port 1 to port 2. After entering the FIlter Service, select the POLicy option of the Level 2 menu. The following screen appears:

```
=======================Show -FIlter POLicy=======================
No policy defined
=============-FIlter POLicy parameter menu (Level 3)==============
       1 - Add
       2 - Delete
       3 - Flush
Select (1-3) ... <CR> to Exit ====> 1
```

Select 1. When the following screen appears, enter the policy "noip Discard IP FROM!1 to !2."

```
=======================Show -FIlter POLicy=======================
No policy defined
=============-FIlter POLicy parameter menu (Level 3)==============
       1 - Add
       2 - Delete
       3 - Flush
Select (1-3) ... <CR> to Exit ====> 1
Add    POLicy       <policyname> <action> <masks> [<context>]
Add    POLicy       noip discard ip from !1 to !2
```

After the policy is added, the message "Policy noip is added" appears on the screen. The following screen now appears:

```
=======================Show -FIlter POLicy=======================
1 policy defined
       id          name     action       masks
=========================================================================
       p0          NOIP     Discard IP   FROM !1 TO !2 (0, 0)
=============-FIlter POLicy parameter menu (Level 3)==============
       1 - Add
       2 - Delete
       3 - Flush
Select (1-3) ... <CR> to Exit ====> 1
[4]NETBuilder #
```

Example 6 Figure 4-5 shows how to count all XNS packets from port 2 to port 1.

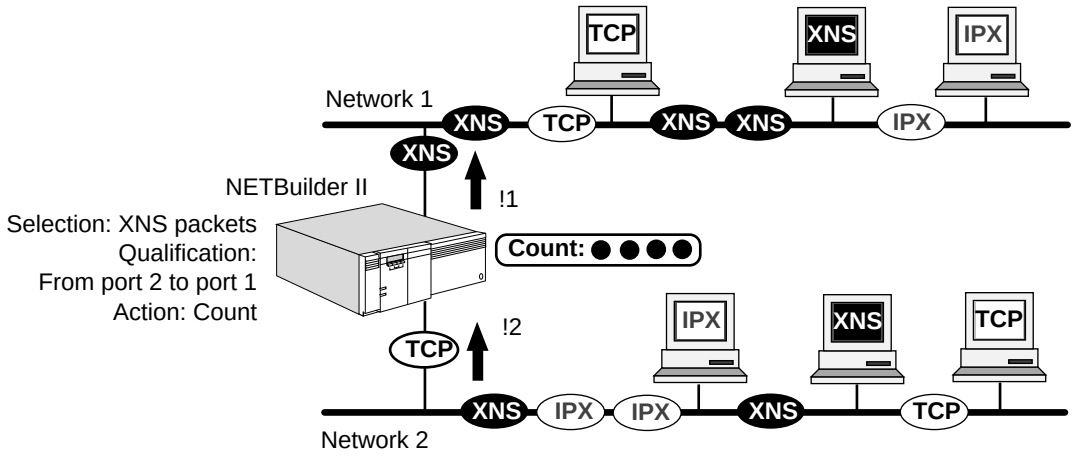


Figure 4-5 Counting XNS Packets

Xerox Network Systems (XNS) packets are selected for special action. The selection is further qualified by specifying from port 2 to port 1. The action is designated as Count. Because built-in masks are defined for XNS packets (refer to Table 4-2), you only need to use the ADD POLicy command to define the policy.

The policy is added after you enter:

```
ADD -FIlter POLicy xnspac Count xns FROM !2 TO !1
```

Example 7 **Discarding Packets on All Ports.** To define a filter to discard DECnet packets on all ports, you need not define a mask, because a predefined mask for DECnet exists. This example could be used for any built-in mask by replacing the mask DECnet with the built-in mask that fits your need.

To define a DECnet filter for all ports, enter:

```
ADD -FIlter POLicy discard_dec Discard decnet
```

Discarding Packets on a Specific Port. To filter out DECnet packets at ports 2 and 3, enter:

```
ADD -FIlter POLicy discdec Discard decnet AT !2, !3
```

Example 8 To check all policies, enter:

```
SHow -FIlter POLicy
```

Example 9 To add a mask that selects packets destined to %080002123456, enter:

```
ADD -FIlter MASK to_atlas dl.dstaddr = %080002123456
```

Example 10 To add a mask that selects packets with LLC encapsulation, enter (the value of either DSAP or SSAP is %aa.):

```
ADD -FIlter MASK snap dl.dsap = %aa
```

Example 11 To add a mask that selects packets with a value greater than %45 at the first byte of data, enter:

```
ADD -FIlter MASK some_data dl.data+%0>%45
```

After the mask is added, the message "Policy some_data is added" appears on the screen.

Example 12 To bridge IP traffic among ports 1, 3, 5, and 6, you can use either command A or B. Command A is preferred, because the built-in mask is encapsulation-independent. Command B forwards IP packets with Ethernet II encapsulation. However, IP packets from token ring or FDDI are handled incorrectly.

Command A:

```
ADD -FIlter POLicy ipgroup Forward ip AMONG !1, !3, !5, !6
```

Command B:

```
ADD -FIlter MASK ethernet_ip %c = %0800
ADD -FIlter POLicy ipgroup Forward ethernet_ip AMONG !1, !3, !5,
!6
```

Example 13 To isolate traffic between two groups of networks, enter:

```
ADD -FIlter MASK any %0 | %ff = %ff
ADD -FIlter POLicy wall Discard any BETWEEN !1, !2 AND !3, !4
```

Packets with any value at offset %0 meet the condition of mask any. Any packet received on port 1 or port 2 and sent to port 3 or port 4 is discarded, but packets received on port 1 and sent to port 2 are not discarded. Similarly, packets received on port 3 and sent to port 4, or packets that are received on port 4 and sent to port 3, are not discarded.

Example 14 If you want to discard all XNS broadcast packets, enter command A or command B. Command A is preferred because the built-in mask is encapsulation-independent.

Command A:

```
ADD -FIlter POLicy noxnsbc Discard xns bc
```

Command B:

```
ADD -FIlter MASK m1 %0 = %ffffffff
ADD -FIlter MASK m2 %4 = %ffff
ADD -FIlter MASK m3 %C = %0600
ADD -FIlter POLicy p1 Discard m1, m2, m3
```

Table 4-7 explains the filter conditions in command B. All broadcast packets that have destination addresses of %ffffffffffff meet the conditions of the first and second masks. Only XNS packets meet the third condition.

Table 4-7 Filter Conditions

	Mask m1	Mask m2	Mask m3
Offset	0	4	C
Meaning	First 4 bytes of destination address	Last 2 bytes of destination address	Packet type
Mask	ffffffff	ffff	0600
Operator	None	None	None
Effect	If first 4 bytes of destination address are ffffffff, the condition is met.	If last 2 bytes of destination address are ffff, the condition is met.	If packet is an XNS packet, the condition is met.

Example 15 The following example shows the use of the logical OR operator. The following commands filter all packets that contain 500 (hexadecimal) or more bytes by applying the mask 11111111 to the byte at offset 500. If any value is present at that location, the filtering condition is met.

```
ADD -FIlter MASK tail %500 | %ff = %ff
ADD -FIlter POLicy drop Discard tail
```

Suppose the value 10110010 is present at offset 500 hexadecimal. When the logical OR operates on this value against the mask 11111111, the result is as follows:

10110010 OR 11111111 = 11111111

Because the result is the same as the mask, the condition is met.

If no value is present at that location, the result is always false. Packets that contain more than 500 hexadecimal bytes should be blocked.

Applying a logical OR to any value and a mask of 11111111 always has a result of 11111111; if any value is present at byte 500, the condition is met. This means that any packet that contains 500 (hexadecimal) or more bytes is filtered.

Example 16 The following example shows the use of one logical operator:

```
ADD -FIlter MASK andmask %a&%80 = %80
ADD -FIlter MASK ormask %a | %fe = %fe
ADD -FIlter MASK notmask %a! = %8c
ADD -FIlter POLicy together Discard andmask ormask notmask
```

In this example, all packets that meet the following three conditions are filtered:

Condition 1. This condition, %A:&%80, is met if the most significant bit of byte A is 1. It applies the logical AND operator to the value found at byte A and the mask 10000000. Suppose the value at byte A is 10111000:

```
        10111000
AND     10000000
        10000000
```

Because the result, 10000000, equals the mask, 10000000, the condition is met.

Condition 2. This condition, %A:\%FE, is met if the least significant bit of byte A is 0. It applies the logical OR operator to the value found at byte A and the mask 11111110. Suppose the value at byte A is 10111000:

```
        10111000
OR      11111110
        11111110
```

Because the result, 1111110, equals the mask, 11111110, the condition is met.

Condition 3. This condition, %A:!%8C, is met if byte A of the packet does not equal 8C. It compares the value found at byte A to the mask 10001100. Suppose the value at byte A is 10111000; because 10111000 is not equal to 10001100, this condition is met.

If a packet meets all three of these conditions, it is filtered. The packet used in this example meets all three conditions, because the value at byte A is assumed to be %B8; therefore, it is filtered.

A packet with the value 8F at byte A satisfies conditions 1 and 3, but does not meet condition 2; it is not filtered, but is forwarded to the appropriate destination.

Example 17 To add one specific address to the station group "accounting," enter:

```
add -FIlter StationGroup accounting %080002123456
```

Example 18 To discard any traffic destined to the station group "accounting," enter:

```
add -FIlter MASK to_accounting datalink.dstaddr = accounting
add -FIlter POLicy block_account Discard to_accounting
```

Before entering these commands, enter the addresses of the stations belonging to the station group "accounting" using the ADD -FIlter StationGroup command.

Example 19 To delete one specific address from the station group "accounting," enter:

```
DELete -FIlter StationGroup accounting %080002123456
```

Example 20 To delete the station group "accounting," enter:

```
DELete -FIlter StationGroup accounting
```

Before executing this command, you must delete all members of the station group "accounting" and delete any masks using the station group "accounting."

Example 21 To delete all members from the station group "accounting," enter:

```
DELete -FIlter StationGroup accounting ALL
```

Example 22 To show the names of all station groups and the number of addresses in them, enter:

```
SHow -FIlter StationGroup
```

Example 23 To change the name of station group "bldg_100" to the station group "bldg_200," enter:

```
CHange -FIlter StationGroup bldg_100 bldg_200
```

Example 24 This example illustrates how to allow NetWare Security Packets to go across a WAN dial-up link on port 4 only if the link is up, and be discarded if the link is down. You could set the WAN port to DOD and add a user-defined mask, NWSEC for the NetWare Security Packets. To add a filter policy for this, enter:

```
ADD -FIlter POLicy DROPNWSEC DODDISCARD NWSEC AT!4
```

Example 25 This example illustrates how to allow all broadcasts from port 1 to go across a WAN dial-up link on port 4 only if the link is up, and be discarded if the link is down. You could set the WAN port to DOD. You can then add a filter policy with a built-in mask, BC, by entering:

```
ADD -FIlter POLicy DROPBC DODDISCARD BC FROM !1 TO !4
```

Example 26 To create a mnemonic filter using the PROTocolRsrv <tag> action option to allot 10 percent of the bandwidth to packets destined for a certain address that are passing through WAN port 3, follow these steps:

1. Add a filter mask with the name "DSTA_Mask" for a destination address of %0800AABB1111 by entering:

   ```
   ADD -FIlter MASK DSTA_MASK DL.DA = %0800AABB1111
   ```

2. Add a filter policy that will assign the name "dstpol" to the policy, select the name tag "dsta_tag" for the PROTocolRsrv <tag> action option, and add the mask "dsta_mask" by entering:

   ```
   ADD -FIlter POLicy dstpol PROTocolRsrv DSTA_TAG DSTA_MASK
   ```

3. Enable the FIlter Service by entering:

   ```
   SETDefault -FIlter CONTrol = Enable
   ```

4. Assign 10 percent of bandwidth to the PROTocolRsrv name tag "dsta_tag" for port 3 by entering:

   ```
   ADD !3 -PORT PROTocolRsrv DSTA_TAG 10
   ```

5. Set PROTocolRsrv as the -PORT QueueCONTrol parameter option for port 3 by entering:

   ```
   SETDefault !3 -PORT QueueCONTrol = PROTocolRsrv
   ```

After you have made these entries, any packet forwarded by the system matching the mask criteria is allotted 10 percent of the bandwidth in accordance with its name tag ("DSTA_TAG") and bandwidth allocation.

Example 27 This example shows how to use the PROTocolRsrv <tag> action option to reserve a specified percentage of bandwidth for different protocols running on the same bridge/router.

In this example, in a bridge/router bridging IPX, XNS, and IP traffic, the user wants to reserve 40 percent of the bandwidth for IPX traffic, 35 percent for IP traffic, and 20 percent for XNS traffic, and 5 percent is set aside as a default for untagged traffic:

To allocate the required bandwidth for all the protocols, follow these steps:

1. Add a filter policy for each protocol with built-in IPX, IP, and XNS filter masks by entering:

```
ADD -FIlter POLicy POLICY1 PROTocolRsrv ANY_IPX IPX
ADD -FIlter POLicy POLICY2 PROTocolRsrv ANY_IP IP
ADD -FIlter POLicy POLICY3 PROTocolRsrv ANY_XNS XNS
```

2. Select BRidging as the type of packet filtering to occur by entering:

```
SETDefault -FIlter SELection = BRidging
```

3. Enable the FIlter Service by entering:

```
SETDefault -FIlter CONTrol = Enable
```

4. To define the bandwidth percentage to be reserved for each protocol, and to enter name tags that match those entered in the -FIlter POLicy commands, enter:

```
ADD !4 -PORT PROTocolRsrv ANY_IPX 40
ADD !4 -PORT PROTocolRsrv ANY_IP 35
ADD !4 -PORT PROTocolRsrv ANY_XNS 20
```

5. Specify the PROTocolRsrv option for the -PORT Service QueueCONTrol parameter by entering:

```
SETDefault !4 -PORT QueueCONTrol = PROTocolRsrv
```

Example 28 This example shows how to use the PROTocolRsrv <tag> action option to reserve a specified percentage of bandwidth for bridged packets of specified lengths being bridged outbound through a bridge/router WAN port.

In this example, in a bridge configured for IPX traffic, a user wants to reserve the following percentages of bandwidth for packets of the following lengths:

- 50 percent of the bandwidth for packets of a length less than 100 bytes
- 25 percent of the bandwidth for packets of a length between 100 and 400 bytes
- 20 percent of the bandwidth for packets of a length greater than 400 bytes
- 5 percent of the bandwidth is reserved as a default for untagged traffic.

To reserve the specified bandwidth for these packets, follow these steps:

1. Add a user-defined mask for each packet length condition that must be met by entering:

```
ADD -FIlter MASK MYMASK1 IPX.PACKETLEN <100
ADD -FIlter MASK MYMASK2 IPX.PACKETLEN 100–400
ADD -FIlter MASK MYMASK3 IPX.PACKETLEN >400
```

2. Add filter policies to use the filter masks by entering:

```
ADD -FIlter POLicy POLICY_x PROTocolRsrv MYTAG_A MYMASK1
ADD -FIlter POLicy POLICY_y PROTocolRsrv MYTAG_B MYMASK2
ADD -FIlter POLicy POLICY_z PROTocolRsrv MYTAG_C MYMASK3
```

3. Select BRidging as the type of packet filtering to occur by entering:

```
SETDefault -FIlter SELection = BRidging
```

4. Enable the FIlter Service by entering:

```
SETDefault -FIlter CONTrol = Enable
```

5. Define the percentage of bandwidth to be reserved for each of the policies entered in step 4, and enter name tags that match those entered in step 4, by entering:

```
ADD !3 -PORT PROTocolRsrv MYTAG_A 50
ADD !3 -PORT PROTocolRsrv MYTAG_B 25
ADD !3 -PORT PROTocolRsrv MYTAG_C 20
```

6. Specify the PROTocolRsrv option for the -PORT Service QueueCONTrol parameter, by entering:

```
SETDefault !3 -PORT QueueCONTrol = PROTocolRsrv
```

After you have made these entries, all IPX packets of lengths less than 100 bytes going outbound WAN port 3 get 50 percent of the bandwidth. Any IPX packets of a length between 100 and 400 bytes get 25 percent of the bandwidth, and IPX packets of a length greater than 400 bytes get 20 percent of the bandwidth.

Five percent of the bandwidth is reserved by default for untagged traffic. If the full 100 percent of bandwidth is allocated by the commands for various filtering conditions, the system normalizes the amount of bandwidth allotted for each condition so that there is always a reserve of 5 percent for untagged traffic.

IPX Filtering Examples

This section contains examples of filtering features in an IPX environment.

Setting Up IPX Filter Masks

The following examples illustrate how the mnemonic filter can be configured to set up filter masks in an IPX environment.

Example 1

To create a mask named ipxmask1 that filters all IPX packets with the destination socket number equal to that of a NetWare Security Packet (0x457), enter:

```
ADD -FIlter MASK ipxmask1 IPX.DstSocKeT = %0457
```

or

```
ADD -FIlter MASK ipxmask1 IPX.DstSocKeT = NWSecPkt
```

Example 2 To create a mask named ipxmask2 that filters all IPX packets with the destination network number 10 to 20, enter:

```
ADD -FIlter MASK ipxmask2 IPX.DstNETwork 10-20
```

Example 3 To create a mask named ipxmask3 that filters all IPX packets of length greater than 96, enter:

```
ADD -FIlter MASK ipxmask3 IPX.PacketLength > 96
```

Example 4 To create a mask named ipxmask4 that filters all IPX packets where the next 9 bytes match the string "MYSERVER1" (bytes starting from offset 4 bytes after the IPX header), enter:

```
ADD -FIlter MASK ipxmask4 IPX.Data+%4:9 = "MYSERVER1"
```

Setting Up IPX Filter Policies

The following examples illustrate how the mnemonic filter can be configured to set up filter policies that manage IPX traffic in either a bridged or IPX routed environment. The examples assume that no other bridge or IPX policies are active except those that are explicitly configured in the examples. Bridge policies, if configured and selected, are always applied after the IPX policies have been applied and a no-match was the result.

Example 1 For a bridge, to discard all IPX packets with any socket number and forward all Service Advertising Protocol (SAP) and Routing Information Protocol (RIP) packets, follow these steps:

1 Create the user-defined IPX mask by entering:

```
ADD -FIlter MASK IPXM1 ipx.SocKeT = ALL
```

2 Create the filter policies by entering:

```
ADD -FI POLicy IPXP1 discard IPXM1
ADD -FI POLicy IPXP2 forward IPXRIP
ADD -FI POLicy IPXP3 forward SAP
```

The policies are applied as follows:

- An IPX packet is evaluated against policy IPXP2. If it matches, this packet (RIP) is forwarded. If it does not match (not a RIP packet), then it is evaluated against policy IPXP3.
- If it matches IPXP3, this packet (SAP) is forwarded. If it does not match, (not a SAP packet), then it is evaluated against policy IPXP1.

Since IPXP1 has a mask value of "socket = ALL," the packet matches and is discarded. A non-IPX packet is not subjected to those IPX policies, and the action taken depends upon the setting of the DefaultAction parameter. The default value of the DefaultAction parameter is Forward.

In general, an IPX policy using a user-defined IPX mask with the value of ALL is evaluated last among the list of IPX policies.

SAP and RIP packets are not subjected to IPX mnemonic filtering on an IPX router.

Example 2 For a bridge, to forward all SAP packets shorter than 100 bytes and discard all others, follow these steps:

1 Create the user-defined IPX mask by entering:

```
ADD -FIlter MASK IPXM1 ipx.PacketLen < 100
```

2 Create the filter policies by entering:

```
ADD -FI POLicy IPXP1 discard SAP
ADD -FI POLicy IPXP2 forward SAP IPXM1
```

The policies are applied as follows:

- An IPX packet is evaluated against policy IPXP2. If it matches, then this packet (a SAP packet with IPX length less than 100 bytes) is forwarded. If it does not match, it is evaluated against policy IPXP1.
- If it matches policy IPXP1 (a SAP packet with IPX length equal or greater than 100 bytes) then this packet is discarded.
- If it matches none of the policies, then the action taken depends upon the setting of the DefaultAction parameter.

Policies that are more specific (with a greater number of masks or matching criteria) are applied ahead of less specific policies that have fewer matching criteria or masks. In this example, an IPX packet is evaluated against policy IPXP2 first, because IPXP2 uses a superset of the IPXP1 masks and is therefore more specific.

Example 3 To discard IPX packets from all clients except the client with the node address of %00608c37c0ba, follow these steps:

1 Set the DefaultAction parameter to forward by entering:

```
SETDefault -FIlter DefaultAction = Forward
```

2 Create the user-defined IPX mask by entering:

```
ADD -FIlter MASK IPXM1 ipx.SrcNodeAddr != %00608c37c0ba
```

3 Create the filter policies by entering:

```
ADD -FIlter POLicy IPXP1 discard IPXM1
```

In this example, an IPX packet is evaluated against policy IPXP1. If it matches (an IPX packet that does not contain the source node address of %00608c37c0ba), then this packet is discarded. If it does not match, then the DefaultAction parameter is applied. In this example, the packet is forwarded.

Example 4 You can use a combination of policies, for example BRidge and IPX, to manage IPX traffic. To forward only IPX WAN Broadcast packets with the destination network of %45469220 and discard all other IPX packets, follow these steps:

1 Create a user-defined IPX mask by entering:

```
ADD -FIlter MASK IPXM1 ipx.DstNETwork = %45469220
```

2 Create the IPX filter policy by entering:

```
ADD -FIlter POLicy IPXP1 forward WANBC IPXM1
```

3 Create the BRidge filter policy by entering:

```
ADD -FIlter POLicy BRP1 discard IPX
```

The policies are applied as follows:

- An IPX packet is evaluated against policy IPXP1. If it matches, then this packet (a WANBroadcast packet containing the destination network of %45469220) is forwarded. If it does not match, then it is evaluated against BRidge policy BRP1.
- If it matches policy BRP1 (an IPX packet), then this packet is discarded.
- If the packet does not match any policies, the action taken depends upon the setting of the DefaultAction parameter.

Example 5 To discard NetWare security packets going out on a dial-on-demand port, enter:

```
ADD -FIlter POLicy IPXP1 DodDiscard NWSEC
```

In this example, an IPX packet is evaluated against policy IPXP1. If the packet matches a NetWare security packet and is going out on a dial-on-demand port with its dial-up path down, the packet is discarded. If the dial-up path is up, the packet is forwarded but tagged so that it does not hold up the dial path. If a packet does not match, the action taken depends upon the setting of the DefaultAction parameter.

Example 6 This example illustrates how to count the number of IPX packets in each of the following IPX length categories:

Byte length of packets:
<= 100
> 100 and <= 200
> 200 and <= 300
> 300 and <= 400
> 400

To create the masks and policies, follow these steps:

1 Create the user-defined IPX masks by entering:

```
ADD -FIlter MASK IPXM1 ipx.PacketLen <= 100
ADD -FIlter MASK IPXM2 ipx.PacketLen > 100
ADD -FIlter MASK IPXM3 ipx.PacketLen <= 200
ADD -FIlter MASK IPXM4 ipx.PacketLen > 200
ADD -FIlter MASK IPXM5 ipx.PacketLen <= 300
ADD -FIlter MASK IPXM6 ipx.PacketLen > 300
ADD -FIlter MASK IPXM7 ipx.PacketLen <= 400
ADD -FIlter MASK IPXM8 ipx.PacketLen > 400
```

2 Create the filter policies by entering:

```
ADD -FIlter POLicy IPXP1 count IPXM1
ADD -FIlter POLicy IPXP2 count IPXM2 IPXM3
ADD -FIlter POLicy IPXP3 count IPXM4 IPXM5
ADD -FIlter POLicy IPXP4 count IPXM6 IPXM7
ADD -FIlter POLicy IPXP5 count IPXM8
```

In this example, an IPX packet is matched against IPXP1. If its length is less than 100 bytes, that count is incremented. If it does not match, then the packet is matched against IPXP2. If its length is greater than 100 but equal to or less than 200, that count is incremented. If it does not match, then it is matched against IPXP3. If its length is greater than 200 but equal to or less than 300, that count is incremented. If it does not match, then the packet is matched against IPXP4. If its length is greater than 300 but equal to or less than 400, that count is incremented. If it does not match, then the packet is matched against IPXP5. If its length is greater than 400, that count is incremented.

This example illustrates the use of multiple masks for the policies. Refer to the next example (example 7) for an alternative configuration.

Example 7 This example illustrates a procedure for configuring IPX mnemonic filters to count various IPX packets by IPX length.

Byte length of packets:
< 101
101 - 200
201 - 300
301 - 400
> 400

To create the masks and policies, follow these steps:

1 Create the user-defined IPX masks by entering:

```
ADD -FIlter MASK IPXM1 ipx.PacketLen < 101
ADD -FIlter MASK IPXM2 ipx.PacketLen 101 - 200
ADD -FIlter MASK IPXM3 ipx.PacketLen 201 - 300
ADD -FIlter MASK IPXM4 ipx.PacketLen 301 - 400
ADD -FIlter MASK IPXM5 ipx.PacketLen > 400
```

2 Create the filter policies by entering:

```
ADD -FIlter POLicy IPXP1 count IPXM1
ADD -FIlter POLicy IPXP2 count IPXM2
ADD -FIlter POLicy IPXP3 count IPXM3
ADD -FIlter POLicy IPXP4 count IPXM4
ADD -FIlter POLicy IPXP5 count IPXM5
```

In this example, an IPX packet is matched against IPXP1. If its length is less than 101 bytes, that count is incremented. If it does not match, then the packet is matched against IPXP2. If its length is between 101 and 200 inclusive, that count is incremented. If it does not match, then it is matched against IPXP3. If its length is between 201 and 300 inclusive, that count is incremented. If it does not match, then the packet is matched against IPXP4. If its length is between 301 and 400 inclusive, that count is incremented. If it does not match, then the packet is matched against IPXP5. If its length is greater than 400, that count is incremented.

Example 8 This example shows how to use mnemonic filtering to prioritize IPX traffic outbound on a WAN serial port 2. IPX packets are to be prioritized into high, medium, and low priorities according to their packet lengths. The following table shows the packet priority and IPX length:

Table 4-8

Priority	IPX Length
High	< 100
Medium	>= 100 and <= 300
Low	>300

To create the masks and policies, follow these steps:

1 Create the user-defined IPX masks by entering:

```
ADD -FIlter MASK IPXM1 ipx.PacketLen < 100
ADD -FIlter MASK IPXM2 ipx.PacketLen 100 - 300
ADD -FIlter MASK IPXM3 ipx.PacketLen > 300
```

2 Create the filter policies by entering:

```
ADD -FI POLicy IPXP1 PRIOritize High IPXM1 to !2
ADD -FI POLicy IPXP2 PRIOritize Medium IPXM2 to!2
ADD -FI POLicy IPXP3 PRIOritize Low IPXM3 to !2
```

In this example, an IPX packet that matches IPXM1 (one that has an IPX length of less than 100 bytes) is placed into the high-priority output queue. An IPX packet that matches mask IPXM3 (one that has an IPX length greater than 300 bytes) is placed into the low-priority output queue. All other IPX packets match mask IPXM2 and are placed into the medium-priority output queue. The packets in the output queues are then sent out in a high:medium:low ratio that is configured using the -PORT QueueInterLeave parameter.

Example 9 To set up protocol reservation using the PROTOcolRsrv <tag> action option of the -FIlter POLicy parameter so that all IPX packets greater than 400 bytes passing through WAN port number 4 get 25 percent of the bandwidth, follow these steps:

1 Add a user-defined mask called IPXMask that sets the following conditions for the passing packets: the packets must be IPX and the packet lengths must be greater than 400 bytes.

Enter:

```
ADD -FIlter MASK IPXMask IPX.PACKETLEN > 400
```

2 Add a policy that includes the policy name IPXPolicy, the mask IPXMask, and the action option PROTocolRsrv <tag>.

The PROTocolRsrv <tag> action option includes entering a tag name IPXlarge to identify those packets that will receive the reserved bandwidth.

Enter:

```
ADD -FIlter POLicy IPXPolicy PROTocolRsrv IPXLARGE IPXMASK
```

3 Select BRidging as the type of packet filtering to occur by entering:

```
SETDefault -FIlter SELection = BRidging
```

4 Enable the FIlter Service by entering:

```
SETDefault -FIlter CONTrol = Enable
```

5 Add the same physical port and the same tag name as was entered in the FIlter Service POLicy command. Also, enter the 25 percent of bandwidth to be reserved for the designated protocol name.

Enter:

```
ADD !4 -PORT PROTocolRsrv IPXLARGE 25
```

6 Specify the PROTocolRsrv option for the -PORT QueueCONTrol parameter by entering:

```
SETDefault !4 -PORT QueueCONTrol = PROTocolRsrv
```

After this configuration, if the system forwards a packet that contains a matching FILter POLicy, the system provides a queue with the percentage of bandwidth reserved for this PROTocolRsrv <tag>.

5 CONFIGURING SOURCE ROUTE BRIDGING

This chapter describes the minimum steps you must perform to configure your source route bridge and various ways to customize the configuration. It also describes how to troubleshoot the source route bridge and provides basic information on how it works.

For conceptual information, refer to "How the Source Route Bridge Works" on page 5-21.

Configuring a Basic Source Route Bridge

This section describes how to configure a source route bridge to operate in a token ring or Fiber Distributed Data Interface (FDDI) environment. (NETBuilder II bridge/routers only support an FDDI environment.) In this section, a network with multiple rings or other network segments is called an *extended network*. For information on how to configure a source route bridge to operate in a wide area networking environment, refer to "Configure Source Route Bridging over a Wide Area Network" on page 5-4.

Source route bridging is supported on token ring, FDDI, Point-to-Point Protocol (PPP), Frame Relay, Asynchronous Transfer Mode (ATM), Asynchronous Transfer Mode data exchange interface (ATM DXI), X.25, Switched Multimegabit Data Service (SMDS), and Integrated Services Digital Network (ISDN). Also, configuring source route bridging can affect IBM-related services such as SDLC or DLSw. For more information, refer to "Configuring LLC2 with Other Services" on page 21-3.

Prerequisites

This section assumes that you have logged on to the system with Network Manager privilege and set up the ports and paths of your source route bridge according to Chapter 1.

Before setting up a source route bridge, you need to examine your network topology and generate the following:

- A unique number for each ring in an extended network. For example, in the topology shown in Figure 5-1, the four rings have been assigned hexadecimal ring numbers (RNs) 0x20, 0x30, 0x40, and 0x50. In Figure 5-2, the two rings have been assigned hexadecimal RNs 0x20 and 0x40. In addition, the serial interfaces have been assigned the hexadecimal RNs 0x30 and 0x50.
- A unique number for each bridge in a set of parallel bridges. For example, in the topology shown in Figure 5-1, parallel bridges B and E have been assigned the hexadecimal bridge numbers (BNs) 0x4 and 0x5. (When more than one bridge interconnects the same networks, the bridges are called *parallel bridges*.)

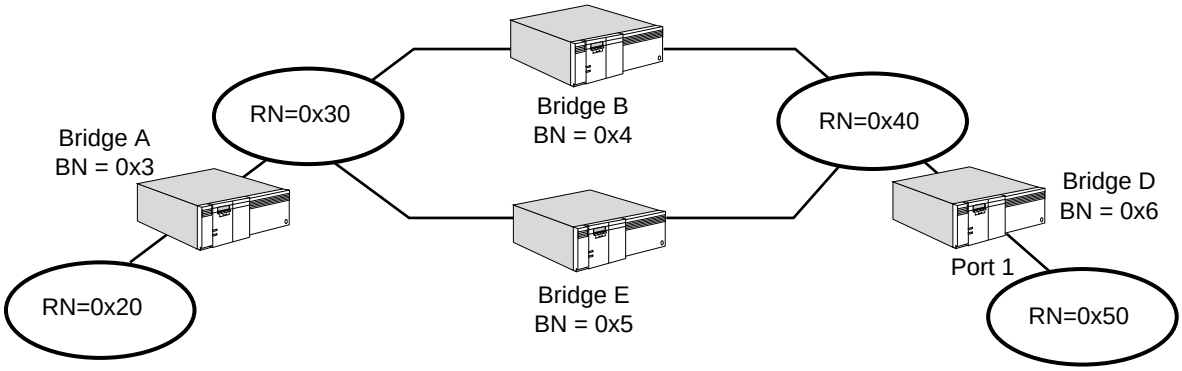


Figure 5-1 Sample FDDI or Token Ring Topology Using NETBuilder II Bridge/Router

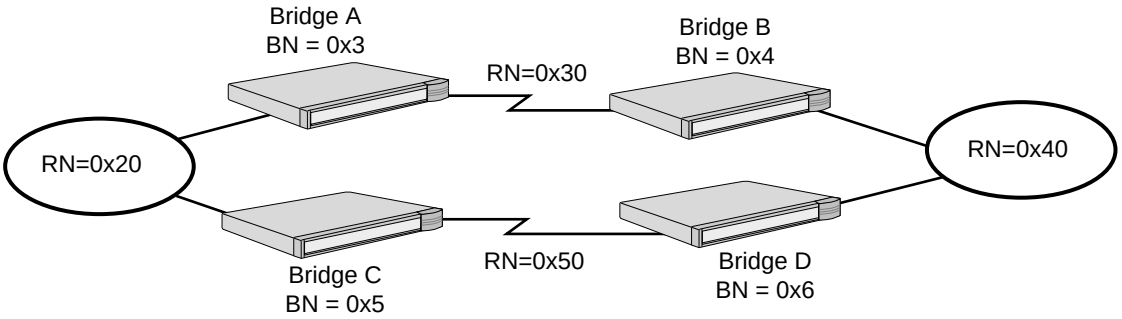


Figure 5-2 Sample Token Ring Topology Using Model 327 or 527 SuperStack II Bridge/Routers

Procedure Figure 5-3 shows a sample token ring topology, which you can refer to while performing this procedure.

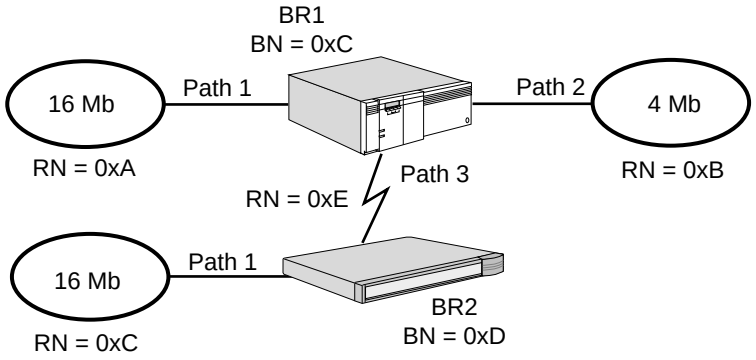


Figure 5-3 Source Route Bridging Sample Topology

To configure a source route bridge, follow these steps:

1 If you are configuring a source route bridge to operate in an FDDI environment, skip this step and go to step 3. If you are configuring a source route bridge to operate in a token ring environment, you may need to set the ring speed of each path.

The default ring speed is 4 Mb. If your source route bridge is a NETBuilder II, you need to perform this step only if your network is composed of 16 Mb rings. If your source route bridge is a model 32x or 52x SuperStack II NETBuilder

bridge/router, the ring speed is automatically detected upon startup. You need to perform this step only if your bridge is connected to an intelligent hub and your network is composed of 16 Mb rings.

For example, to set the ring speed of path 1 of BR1 and BR2 (as shown in Figure 5-3) to 16 Mb, enter the following command on both bridges:

```
SETDefault !1 -PATH BAud = 16000
```

A message similar to the following appears:

```
Note: You must Enable -PATH CONTrol for this Path parameter to
  take effect.
```

2 Enable the paths you set the ring speed for in step 1 using:

```
SETDefault !<path> -PATH CONTrol = Enabled
```

A message similar to the following appears:

```
Thu Jan 1 09:09:14 1995 Path 1 available
```

At this point, connect the DB9 end of the token ring cable that leads from the 16 Mb ring to the token ring interface on your bridge/router.

It will take a minute or two for path 1 to start operating. When path 1 is operational, the system responds with a display similar to the following:

```
Thu Jan 1 09:12:36 1995 Path 1 UP
```

3 Assign each bridge port on your network the ring number of the network it accesses.

If you are setting up a pure router to forward packets to end systems on an extended network, skip this step.

To assign a ring number, use:

```
SETDefault !<port> -SR RingNumber = <number> (1–4095) | 0x<number>
  (1–FFF)]
```

For example, to assign the hexadecimal ring number 0xA to BR1 path 1, as shown in Figure 5-3, enter:

```
SETDefault !1 -SR RingNumber = 0xA
```

To assign the equivalent decimal ring number to BR1 path 1, enter:

```
SETDefault !1 -SR RingNumber = 10
```

A serial line running PPP, Frame Relay, ATM DXI, SMDS, or X.25 is treated as a virtual ring. For information on wide area networking using PPP, Frame Relay, ATM DXI, SMDS, and X.25 refer to Chapter 34, Chapter 42, Chapter 43, Chapter 44, and Chapter 45, respectively.

4 Assign a different bridge number to each bridge in a set of parallel bridges using:

```
SETDefault !<port> -SR BridgeNumber = <number> (0–15) | 0x<number>
  (0–F)
```

If your network is not composed of parallel bridges, you do not need to assign a unique bridge number to each bridge. You can use the default setting of 3.

To assign the hexadecimal bridge number 0xC to a bridge, enter:

```
SETDefault -SR BridgeNumber = 0xC
```

To assign the equivalent decimal bridge number to a bridge, enter:

```
SETDefault -SR BridgeNumber = 12
```

5 Enable global bridging on each bridge.

For example, enable bridging on BR1 and BR2 by entering the following command on each bridge:

```
SETDefault -BRidge CONTrol = Bridge
```

Source route bridging is enabled by default on all ports (the default setting of the -SR SrcRouBridge parameter is SrcRouBridge) and source route bridging should begin to operate after you assign a ring number and enable global bridging.

6 If you do not want to operate in source route transparent (SRT) mode, disable per-port transparent bridging using:

```
SETDefault !<port> -BRidge TransparentBridge = NoTransparentBRidge
```

Transparent bridging is not supported on models 32x and 52x SuperStack II NETBuilder bridge/routers. You do not need to perform this step for this model.

After you complete this procedure, go to "Verifying the Configuration" on page 5-5.

Configure Source Route Bridging over a Wide Area Network

You can configure your source route bridge to forward packets over the following types of wide area networks:

- PPP
- Frame Relay
- Asynchronous Transfer Mode (ATM)
- Asynchronous Transfer Mode Data Exchange Interface (ATM DXI)
- X.25
- SMDS

Source Route Bridging over PPP

For complete information on configuring PPP, refer to Chapter 34.

Source Route Bridging over Frame Relay, ATM, ATM DXI, and X.25

Source route bridging over Frame Relay, ATM, ATM DXI, and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to source route bridge over a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. For complete information on configuring source route bridging over Frame Relay, ATM, or ATM DXI, including a discussion of fully meshed, partially meshed, or nonmeshed topologies and virtual ports, refer to Chapter 42, Chapter 47, and Chapter 43. For complete information on configuring source route bridging over X.25, including a discussion of fully meshed, partially meshed, or nonmeshed topologies and virtual ports, refer to Chapter 45. For information on the number of virtual ports supported per platform, refer to Table 1-1 in Chapter 1.

When creating virtual ports over a heavily trafficked partially meshed or nonmeshed topology, 3Com recommends that each source route bridge on the

Frame Relay, ATM, ATM DXI, or X.25 network have a permanent virtual circuit for the proper operation of the Spanning Tree Protocol. Figure 5-4 shows a network composed of two NETBuilder II bridges and a model 327 SuperStack II bridge connected by virtual ports. The interconnection of the three source route bridges causes a potential loop. The Spanning Tree Protocol can prevent this loop by blocking a route as shown in Figure 5-4.

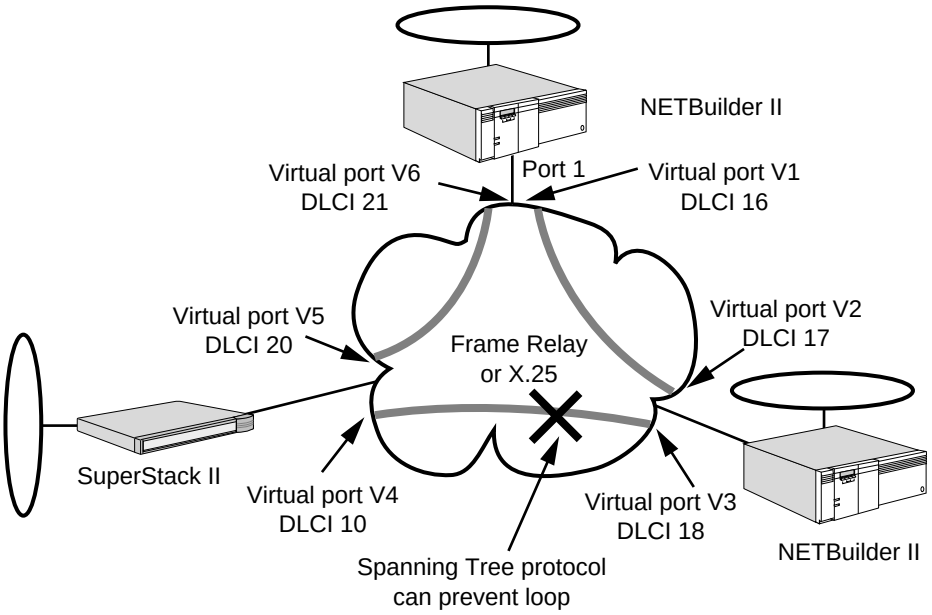


Figure 5-4 Source Route Bridging Over Frame Relay or X.25 in a Nonmeshed Topology with a Potential Loop

Source Route Bridging over SMDS

Source route bridging over SMDS is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure source route bridging over SMDS, refer to Chapter 44.

Source Route Bridging over ISDN

For information on wide area networking using Integrated Services Digital Network (ISDN), refer to Chapter 35.

Verifying the Configuration

After you configure your source route bridge, you need to verify its configuration by following these steps:

1. Check the state of the current configuration by entering:

   SHow -SR DIAGnostics

   The display provides troubleshooting information about source route configuration errors and gives suggestions for corrective actions.

2. Check the configuration of the source route bridge and the status of each port and path by entering:

   SHow -SR CONFiguration

The display shows the source route bridging status on a per-port basis. A Forwarding status indicates that source route bridging is activated. A Down status indicates that source route bridging is not activated because of one or a combination of the following conditions:

- The port or path is disabled.
- The -BRidge CONTrol parameter is set to NoBridge.
- The -SR SrcRouBridge parameter is set to NoSrcRouBridge.
- A ring number has not been assigned to the port.

The display also indicates if end system source routing or source route transparent bridging gateway (SRTG) is enabled. For more information, refer to "Guidelines for Per-Port Route Discovery" on page 5-17 and "Configuring Source Route Transparent Bridging Gateway" on page 5-10.

3 If the display indicates that a port or a path is down, follow these steps:

a Check the configuration of each port by entering:

```
SHow -PORT CONFiguration
```

b Check the configuration of each path by entering:

```
SHow -PATH CONFiguration
```

4 Test the source route bridge by sending packets across it.

For example, make a connection from a device on one attached network to a host on another attached network. If you can successfully make a connection, the source route bridge is ready for normal operation; otherwise, refer to "Troubleshooting the Configuration" on page 5-7.

Getting Statistics

After your source route bridge is up and running, you may want to gather statistics. For information on interpreting the statistics display, refer to Appendix H.

You can collect statistics for a specific time period by using the -SYS SampleTime and -SYS STATistics parameters. For more information, refer to Chapter 58 in *Reference for NETBuilder Family Software*.

To gather statistics, follow these steps:

1 Display source route bridging statistics for all ports by entering:

```
SHow -SYS STATistics -SR
```

2 Display statistics for all ports by entering:

```
SHow -SYS STATistics -PORT
```

3 Check the statistics for all paths by entering:

```
SHow -SYS STATistics -PATH
```

If the display indicates that there are errors (for example, cyclic redundancy check errors) on the attached network, check:

- That the transceiver cable is properly attached to the transceiver.
- That the transceiver is properly attached to the network cable.
- That the network is properly terminated.

If the errors happen on a serial line, check:

- Cable attachments.
- Channel service unit/digital service units (DSU/CSUs).
- Modems on each end of the serial line.

If the line is a leased line, request help from the company that leases the line (for example, the telephone company).

Troubleshooting the Configuration

To troubleshoot the source route bridge, follow these steps:

1 Check for configuration errors using:

```
SHow [!<port>] -SR DIAGnostics
```

The display provides troubleshooting information about source route configuration errors and gives suggestions for corrective actions.

2 Access source route bridge configuration information and check the status of each path. Verify that each path is assigned to the appropriate network by entering:

SHow -SR CONFiguration

Make sure that the status of the source route bridge is Forwarding. Verify that the path is enabled by entering:

SHowDefault -PORT CONFiguration
SHowDefault -PATH CONFiguration.

3 Display all learned remote routes using:

```
SHow [!<port>] -SR WanRoutes
```

SHow displays all the currently learned source routes and the associated DLCI, SMDS individual address, or X.25 DTE address for each learned route. If the port is specified, the display for port-related parameter values is limited to that port.

4 If the display in step 1 indicates that a port or path is down, follow these steps:

a Check the configuration of each port by entering:

SHow -PORT CONFiguration

b Check the configuration of each path by entering:

SHow -PATH CONFiguration

5 Check for other activity on the source route bridge through the statistics display.

a For a detailed accounting of errors on a given port, enter:

SHow -SYS STATistics -SR

If there is no other activity on the source route bridge, check its physical attachments to other networks, including boards, back panel connectors, and transceiver or modem connectors. For lines to wide area bridges, check the DSU/CSU or modem and its configuration.

b If a large number of errors occur on a bridge's local or serial line to a network, check the physical lines.

For a detailed accounting of errors on a given path, enter:

SHow -SYS STATistics -PATH

Some statistics can be set to zero using the FLush -SYS STATistics command to provide a starting point for subsequent analysis of these reports.

6 If possible, replace any bridge you suspect has problems with another bridge or a repeater. Check to see if the problem persists.

If the problem persists, then the bridge is not the cause of the problem.

To determine whether a pair of source route bridges can communicate with each other, use the data link test. This test allows the bridges to exchange test packets and display the related statistics. Use the DLTest command, which is described in Chapter 1 in *Reference for NETBuilder Family Software.*

Related Information

End systems on token ring report soft errors such as frame-copied errors through the media access control (MAC) Report Error frame. End systems may generate a small number of MAC Frame Copy error report packets when a NETBuilder II Bridge is initializing. For the NETBuilder II system to learn addresses on the token ring, it copies the packet with the unknown source address and sets the address-recognized (A) and frame-copied (C) bits in the Frame Status (FS) field (1 byte) located at the end of the frame after the Frame Check sequence and the Ending Delimiter field.

A problem occurs when the FS (A) and (C) bits have been set and the destination of the frame is a local end system. The end system normally sets the (A) and (C) bits, and when it receives a frame with these values already set, it reports an error. These errors are counted until the error threshold is reached; then a MAC Report Error is sent out by the end system.

Customizing the Source Route Bridge

Table 5-1 summarizes the features that allow you to customize your source route bridge and which platforms each feature is supported on.

Table 5-1 Source Route Bridge Features/Platforms Supported

Source Route Bridge Feature	NETBuilder II	Model 32x and 52x SuperStack II NETBuilder
Per-port source route bridging	Yes	Yes
Per-port source route transparent bridging	Yes	No
Source route transparent bridging gateway (SRTG)	Yes	No
IBM connectivity	Yes	Yes
Largest frame size	Yes	Yes
Passive bridging	Yes	No
Spanning tree in a source route bridging environment	Yes	Yes
Parallel bridges	Yes	No
Broadcast traffic reduction	Yes	Yes
Explorer frame propagation	Yes	Yes
Filters	Yes	Yes
Security	Yes	Yes
Configuration as an end system Per-port route discovery for end system source routing Utility for discovering routes to an end system Static routes Aging entries in the routing table Token access priority	Yes	Yes

This section briefly describes and explains how to set up the source route bridging features. Not all available parameters are discussed in this section. For more information on all available parameters, refer to Chapter 56 in *Reference for NETBuilder Family Software*.

Enabling and Disabling Per-Port Source Route Bridging

By default, source route bridging is enabled on all ports. You can disable source route bridging on specified ports using:

```
SETDefault !<port> -SR SrcRouBridge = NoSrcRouBridge
```

To enable source route bridging, use:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
```

For source route bridging to take effect on a port, the port must additionally be enabled (as described in Chapter 1), the -BRidge CONTrol parameter must be set to Bridge (as described in Chapter 3), and ring numbers must be assigned (as described in "Configuring a Basic Source Route Bridge" on page 5-1).

For complete information on the -SR SrcRouBridge parameter, refer to Chapter 56 in *Reference for NETBuilder Family Software*.

Enabling and Disabling Per-Port Source Route Transparent Bridging

This feature is not supported on model 32x and 52x SuperStack II NETBuilder bridge/routers.

If your token ring or FDDI network is composed of users on transparent (non-source route) end systems as well as source route end systems as shown in Figure 5-5, you can enable transparent bridging on your source route bridge. By enabling source route transparent bridging, your source route bridge can forward source route or transparent bridged frames. For conceptual information, refer to "Source Route Bridging" on page 5-22.

By default, source route transparent bridging is enabled on all ports. As shown in Figure 5-5, Bridge 1 has source route transparent bridging enabled, which allows the transparent end systems A, C, and E to communicate. The source route end system B can communicate with the source-route-only end system D. However, the source-route-only end systems B and D cannot communicate with transparent only end systems A, C, or E.

If you want to disable transparent bridging on some ports, use:

```
SETDefault !<port> -BRidge TransparentBRidge = NoTransparentBRidge
```

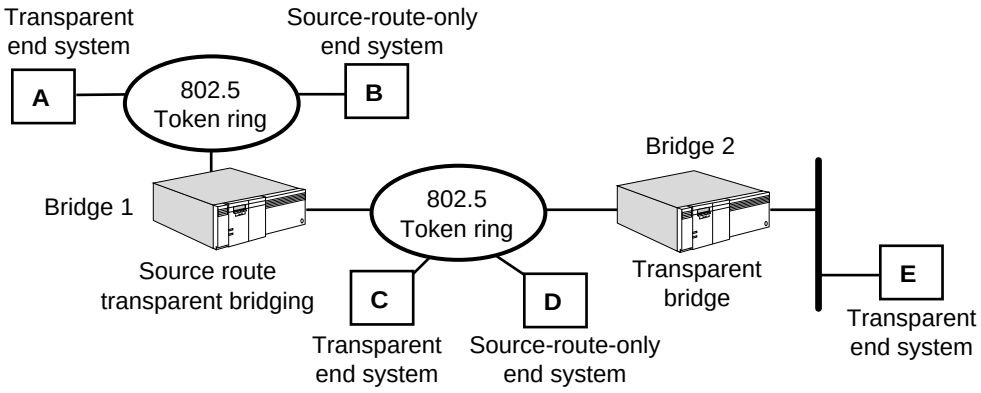


Figure 5-5 Source Route Transparent Bridging

Configuring Source Route Transparent Bridging Gateway

This feature is not supported on model 32x and 52x SuperStack II NETBuilder bridge/routers.

You can connect source route and transparent bridging domains, and allow communication between the two by configuring SRTG.

The SRTG feature is supported on the NETBuilder II platform and on all LAN and WAN media currently offered by 3Com.

The SRTG bridges only logical link control, type 2 (LLC2) and NetBIOS traffic between source route and transparent bridging domains. SRTG supports both 802.3 and Ethernet Version II frames on Ethernet, and supports multiple paths between source route and transparent bridging domains (only one path is active at a time because the SRTG detects and breaks loops according to the spanning tree algorithm).

For conceptual information about SRTG, refer to "Source Route Transparent Bridging Gateway Concepts" on page 5-24.

Prerequisites

This section assumes that you have logged on to the system with Network Manager privilege and set up the ports and paths of your source route bridge according to Chapter 1.

Procedure

To configure the SRTG to support bridging of LLC2 and NetBIOS traffic between source route and transparent bridging domains, refer to Figure 5-6 and follow these steps:

You cannot perform both transparent bridging and source route bridging on a port being used for the SRTG. You can perform either transparent or source route bridging, but not both at the same time.

1. Configure the basic source route bridge on the NETBuilder II source route port connected to the source route domain by referring to "Configuring a Basic Source Route Bridge" on page 5-1.

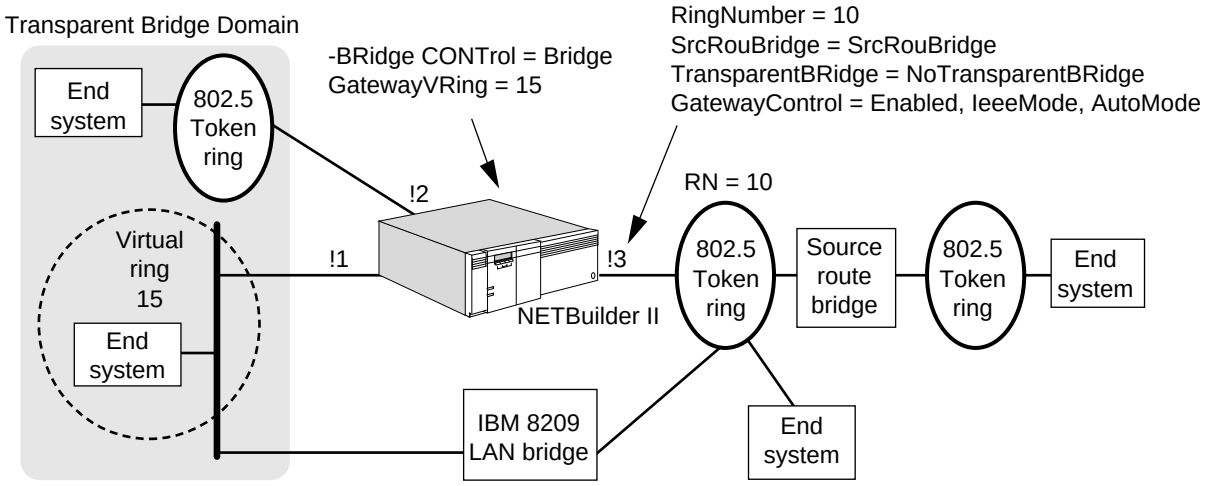


Figure 5-6 Source Route Transparent Bridging Gateway Configuration

2 Verify that source route bridging is enabled on the source route port by entering:

SHow -SR SrcRouBridge

If it is disabled, enable it using:

```
SETDefault !<port> -SR SrcRouBridge = SrcRouBridge
```

3 If no transparent bridging stations exist in the source route domain, disable transparent bridging on the source route port using:

```
SETDefault !<port> -BRidge TransparentBRidge = NoTransparentBRidge
```

4 Verify that transparent bridging is enabled on the transparent bridging port using:

SHow -BRidge TransparentBRidge

If transparent bridging is not enabled on the specified port, use:

```
SETDefault !<port> -BRidge TransparentBRidge = TransparentBRidge
```

5 Configure a virtual ring number for the transparent bridging domain using:

```
SETDefault -SR GatewayVRing = <number>(1–4095) | 0x<number>(1–FFF)
```

You can enter the virtual ring number in decimal or hexadecimal (precede the hexadecimal number with a 0x as indicated in the syntax).

Before forwarding packets from the transparent bridging domain, SRTG adds the virtual ring number and its own bridge number to the source route information of the destination station retrieved from the source route table. From point of view of a source route station, the entire transparent bridge LAN appears as a single source route ring.

6 Enable SRTG on both the source route and transparent bridging ports, and set the encapsulation format on the transparent bridging port (Ethernet only) by using:

```
SETDefault !<port> -SR GatewayControl = ([Enabled | Disabled],
 [IeeeMode | EtherMode], [AutoMode | NoAutoMode])
```

Select Enabled to enable SRTG.

The combination of the next two pairs of settings determines what encapsulation format is used when translating token-ring LLC-based packets.

If NoAutoMode is selected, SRTG does not keep track of the encapsulation format of each transparent bridging station. The final encapsulation method is determined by EtherMode (Ethernet II encapsulation with packet type of 0x80D5) or IeeeMode (IEEE 802.3 encapsulation) settings when the packets are bridged to the Ethernet domain.

Select AutoMode if you want SRTG to automatically keep track of the encapsulation format of each station. If AutoMode is selected, different packet translation rules are used for known stations and unknown stations. For known stations, the IeeeMode | EtherMode settings are ignored and the encapsulation format learned for those stations is used. For unknown stations, LLC-based packets are translated into both 802.3 and Ethernet Version II frames.

The DSAP field in the token ring 802.2 frame must be a multiple of 4s (00, 04, 08, and so forth) except BC and E0, which are reserved for Banyan VINES and IPX, respectively.

For more information about frame conversion, refer to "Frame and Address Conversion" on page 5-27.

After SRTG is enabled, packets are bridged between the source route and transparent bridge domains.

Do not enable both data link switching (DLSw) and SRTG on the same port because packet duplication may occur if both features connect the same areas.

Related Information

If your SRTG topology includes a transparent bridge in the transparent bridge domain and your application involves NetBIOS and Systems Network Architecture (SNA) traffic that uses functional addresses as a destination address, you may have to add a mapping between the functional and multicast address on the transparent bridge if the destination and source media types are different. Use the -BRidge FunctionalAddr parameter. For more information, refer to "Translation Bridging" and "Adding Functional-Address-to-Multicast-Address Mappings to the Default Table" in Chapter 3.

Connecting IBM Bridges to 3Com Token Ring Bridges

This section provides information on connecting 3Com token ring bridges to IBM bridges.

Procedure

For complete information on setting your 3Com token ring bridge to source route or source route transparent mode, refer to "Configuring a Basic Source Route Bridge" on page 5-1 and "Enabling and Disabling Per-Port Source Route Transparent Bridging" on page 5-9.

Related Information

Some IBM bridges support source route-only mode. When configuring these bridges and 3Com token ring bridges in the same network environment, you must configure the 3Com bridge in either source route or source route transparent mode. For more information about source route and source route transparent mode, refer to "Source Route Bridging" on page 5-22 and "Source Route Transparent Bridging" on page 5-22.

IBM bridges support the hexadecimal-only format for bridge and ring numbers. The 3Com token ring bridge supports entry of both decimal and hexadecimal format for the -SR RingNumber and -SR BridgeNumber parameters. A hexadecimal format entry must be preceded by a 0x, as shown in the following examples:

```
SETDefault !1 -SR RingNumber = 0xA
```

The ring number is displayed as decimal 10 with the hexadecimal equivalent in parentheses.

```
SETDefault -SR BridgeNumber= 0xF
```

The bridge number is displayed as decimal 15 with the hexadecimal equivalent in parentheses.

The IBM PC LAN Bridge is not fully compatible with the 3Com token ring implementation of Spanning Tree Protocol in a parallel bridge configuration. In

this configuration, the 3Com token ring bridge forwards single-route broadcast frames. When configuring the IBM PC LAN Bridge in a parallel bridge configuration with a 3Com token ring bridge, set the 3Com bridge as source route-only mode. The IBM PC LAN Bridge sends out a broadcast test packet before it can become fully operational to ensure that IBM bridge adapters are not on the same ring. A parallel 3Com token ring bridge in source route transparent or transparent mode can forward this test packet, confusing the IBM PC LAN Bridge and preventing it from coming up. To ensure that the two parallel bridges come up, the 3Com token ring bridge must be in source route-only mode.

Configuring the Largest Frame Size

The LargestFrameSize parameter specifies the maximum size frame that can be sent and received on a port. The source route bridge negotiates the largest frame size of all transit routes down to this size.

Use this parameter to regulate the amount of data transmitted by end systems to prevent time-outs due to slow network links. If the connected network contains low-speed WAN links, assign a lower largest frame size value.

The base values specified in IEEE 802.1D are supported and are listed in Table 5-2. Extended values listed in the IEEE specification are not currently supported.

Table 5-2 Valid Largest Frame Size Values

LargestFrameSize Parameter Setting	Data Unit Length
0	516 octets
1	1,470 octets
2	2,052 octets
3	4,399 octets
4*	8,130 octets
5*	11,407 octets
6*	17,749 octets
7*	41,600 octets

* These values are not supported.

By default, 3Com bridge/routers use a setting of 3, which is equivalent to a frame size of 4,399 octets.

The value can be changed using:

```
SETDefault !<port> -SR LargestFrameSize = <number>(0–7)
```

The maximum physical frame size that can be received and forwarded by a NETBuilder II system with a Token Ring or Token Ring + module and model 32x and 52x SuperStack II bridge/routers is 4,500 bytes.

Configuring Passive Bridging

This feature is not supported on model 32x and 52x SuperStack II bridge/routers.

To work around the bridge/router hop-count limitation for token ring networks consisting of eight or more rings, you can configure the attached source route bridges for passive bridging and effectively create one logical ring from multiple rings. Creating logical rings allows you to work around the token ring adapter limitation on the maximum number of rings in the route designator fields.

Procedure

To configure passive bridging on a network similar to Figure 5-7, follow these steps:

1 Configure the bridges that are within the logical ring.

a Enable passive bridging.

For example, on bridge 1 and bridge 2, enter:

```
SETDefault -SR Mode = PassiveBridging
```

By setting this parameter to PassiveBridging, all source-routed frames are transparently bridged across the spanning tree paths without examining or updating the routing information field (RIF) in the frame header. For information about the frame header, refer to "IEEE 802.5 Token Ring Frame Format Overview" on page 5-23.

b Configure the same ring number on the bridge ports that are part of the same logical ring.

When you set up passive bridging, the same ring number must be assigned to all physical rings that are part of one logical ring.

For example, to create the logical ring 10 (decimal), on bridge 1 and bridge 2, enter:

```
SETDefault !1 -SR RingNumber = 10
SETDefault !2 -SR RingNumber = 10
```

2 Configure source route bridging for the remaining bridges outside the logical ring.

a Configure ring numbers for the remaining bridges.

For example, to configure the ring numbers in decimal, on bridge 3 ports, enter:

```
SETDefault !1 -SR RingNumber = 10
SETDefault !2 -SR RingNumber = 30
```

b Verify that IEEE bridging is enabled.

By default, the Mode parameter is set to IEEE. Verify its setting by entering:

```
SHow -SR Mode
```

If the setting is not IEEE, configure this parameter by entering:

```
SETDefault -SR Mode = IEEE
```

By setting this parameter to IEEE, the forwarding path of the specifically routed frame (SRF) is determined by the RIF in the frame header. For information about the frame header, refer to "IEEE 802.5 Token Ring Frame Format Overview" on page 5-23.

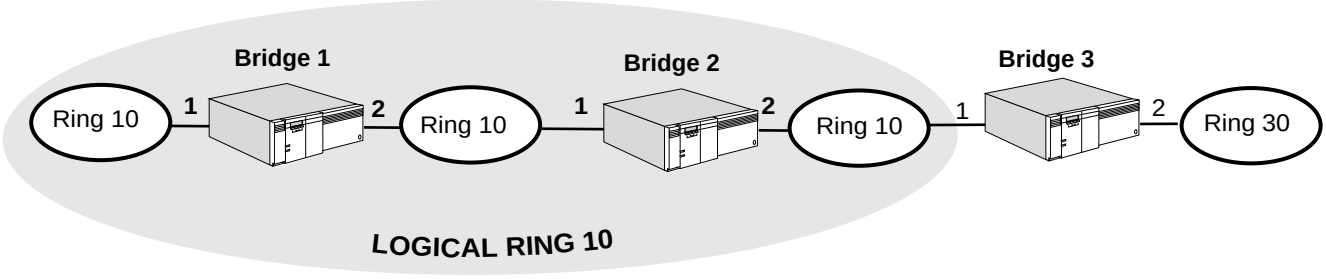


Figure 5-7 Collapsing Multiple Rings into One Logical Ring with Passive Bridging

Setting Up Spanning Tree

In a source route bridging network, an end system can discover a route to a destination system on another ring by sending an All Routes Explorer (ARE) frame that is copied to every ring in the network. If only one path to the destination exists, the destination system only receives and responds to one ARE frame. However, if multiple paths to the destination system exist, the destination system receives and responds to as many copies of the ARE frame as there are paths to it, resulting in heavy network traffic.

To limit the number of ARE frames in a source route bridging environment, 3Com bridge/routers can use the Spanning Tree Protocol (STP) to dynamically establish and maintain a spanning tree across all rings, allowing only a single spanning tree explorer (STE) frame to be forwarded on a ring and preventing duplicate ARE frames from appearing on the same ring. The STP Service is enabled by default so no additional user configuration is necessary. If the STP Service has been disabled, you can enable it by entering:

```
SETDefault -STP CONTrol = Enabled
```

You must disable transparent bridging on all ports before the STP packets are generated for the source route domain. Otherwise, the bridge/router generates STP packets for the transparent domain. To disable transparent bridging, use:

```
SETDefault !<port> BRidge TransparentBRidge = NoTransparentBRidge
```

Transparent bridging is not supported on model 32x and 52x SuperStack II bridge/routers. You do not need to perform this step for those bridge/routers.

For conceptual information about the Spanning Tree Protocol, refer to Chapter 3. For conceptual information about the route discovery process, refer to "Route Discovery Process" on page 5-28.

Configuring Parallel Bridges

This feature is not supported on model 32x and 52x SuperStack II bridge/routers.

If your network is composed of parallel source route bridges to provide redundancy as shown in Figure 5-1, you must assign unique bridge numbers to them using:

```
SETDefault -SR BridgeNumber = <number> (0–15) | 0x<number> (0–F)
```

3Com token ring bridges support both the decimal and hexadecimal format for the bridge number. Hexadecimal format entry must be preceded by a 0x.

As shown in Figure 5-1, bridge B has been assigned a bridge number of 4, and bridge E has been assigned a bridge number of 5.

Reducing Broadcast Traffic

You can reduce the amount of broadcast traffic in your source route bridging environment by regulating the maximum number of broadcast packets per second and setting the broadcast timer threshold to specify when to begin discarding broadcast packets.

To set the maximum amount of broadcast packets per second on a port, use:

```
SETDefault !<port> -BRidge BroadCastLimit = <packets per second>
 (0–100000)
```

To set the broadcast limit timer threshold, use:

```
SETDefault -BRidge BLimitTimer = 400 | 600 | 800 | 1000 |
 Disabled
```

The broadcast limit mechanism works by counting the number of broadcast and multicast packets received during each timer interval. Broadcast and multicast packets are forwarded during a timer interval until the broadcast limit threshold (described later in this chapter) for the port is reached. After the threshold has been reached, no additional broadcast or multicast packets are forwarded on the port until the start of the next timer interval. At that point, broadcast and multicast forwarding is resumed.

To disable the BroadCastLimit parameter, specify 0. To disable the BLimitTimer parameter, specify "Disabled."

Restricting Explorer Frame Propagation

You can restrict the propagation of ARE or STE frames to reduce unnecessary explorer traffic using:

```
SETDefault !<port> -SR MaxAreRDLimit = <number> (0–8)
SETDefault !<port> -SR MaxSteRDLimit = <number> (0–8)
```

Whether you use a value other than the default depends on your network configuration.

The MaxAreRDLimit parameter specifies the maximum number of route designators (RDs) (or hop count) allowed for an ARE frame received on the specified port. The default value of the MaxAreRDLimit parameter is eight, the maximum allowed in a source route bridging environment. This means that the maximum number of bridges or hops that can be daisy-chained in a source route bridge configuration is seven. You can further restrict the hop count by adjusting the MaxAreRDLimit parameter. When the source route bridge receives an ARE frame, it checks the setting of this parameter before forwarding it. If the setting is exceeded, the ARE frame is discarded.

The MaxSteRDLimit parameter specifies the maximum number of RDs allowed for an STE frame received on the specified port. The default value of the MaxSteRDLimit parameter is eight. If the number of route designators in the frame is equal to or greater than the MaxSteRDLimit, the frame is discarded. Otherwise, the STE frame is forwarded.

Configuring Filters

For complete information on configuring filters, refer to Chapter 4.

Configuring Security

You can use the bridge security features to select certain stations whose packets will be forwarded or blocked depending on their source or destination address. For complete information on using the -BRidge SRcSecurity and DStSecurity parameters, refer to "Bridge Security" on page 3-9.

Configuring the Bridge/Router as an End System

The remaining procedures in this section apply to the 3Com bridge/router functioning as an end system for network management purposes or as a level 3 router for routing protocol packets, such as Novell and AppleTalk in a source route environment.

Guidelines for Per-Port Route Discovery

You must configure route discovery on a port if the bridge/router must forward end system protocol packets to other end systems across a source route-only bridge. Use the following guidelines for setting RouteDiscovery in your network environment:

- Enable for IP on applicable ports if you want to network manage your bridges or routers that traverse a source route-only bridge. Enable route discovery for DLTest (Data Link Test) if you want to run DLTest to other 3Com bridges or routers that traverse source-route-only bridging environments.
- For routers in a source route-only environment as shown in Figure 5-8, enable RouteDiscovery for the appropriate protocols on applicable ports to ensure connectivity with source route-only end systems.

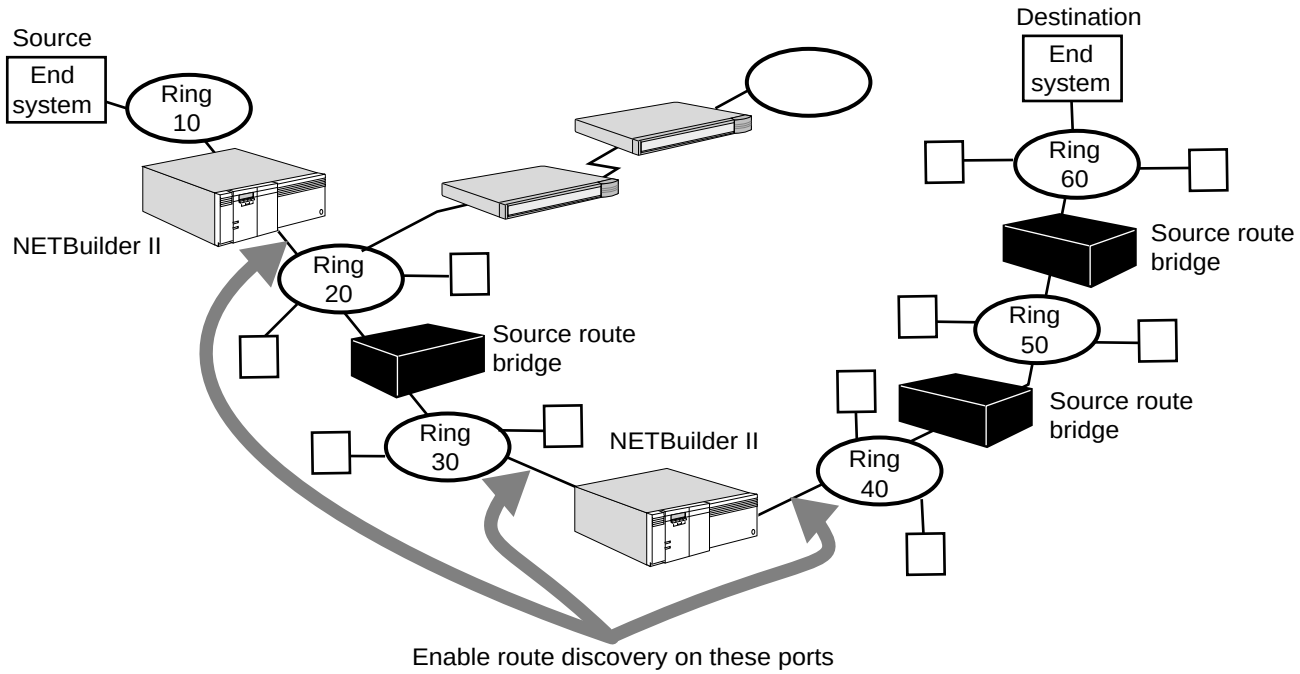


Figure 5-8 Route Discovery for Routers in a Source Route-Only Environment

For the following specific configurations, note the guidelines:

- Connecting a transparent domain with a source route domain

  Unless the SRTG feature is enabled, 3Com bridge/routers do not support the conversion of a transparent bridged frame to a source route bridged frame, or vice versa. You must configure a router to interconnect transparent domains (for example, connected through Ethernet ports) with source route domains (for example, connected through token ring ports). Enable Route discovery on ports that are connected to the source route-only domains or source route transparent domains with source route-only end systems.

- Routing domains connected by a source route-only bridged domain

  When you have two routing domains connected by one or more source route-only bridges, you must enable route discovery on the router ports directly connected to the source route-bridged domain.

Configuring Per-Port Route Discovery

To configure route discovery, use:

```
SETDefault !<port> -SR RouteDiscovery = ([All | None] |
 [AppleTalk | NoAppleTalk], [CLNP | NoCLNP], [DECnet |
 NoDECnet], [DLTest | NoDLTest], [IP | NoIP], [IPX | NoIPX] [LLC2
 | NoLLC2], [VINES | NoVINES])
```

With this command, you can specify different combinations of protocols for end system route discovery to take place over a specific port.

The default for the RouteDiscovery parameter is None, which means that all end system packets are transmitted as transparent frames, and can reach end systems in a transparent bridged or a source route transparent (SRT) bridged environment.

You can specify that route discovery is initiated for all end system packets over a given port if a route to the destination end system does not exist in the local routing table. To specify route discovery for all end system packets, use:

```
SETDefault !<port> -SR RouteDiscovery = All
```

Specifying "All" can significantly impact the performance of the router. The router experiences a significant drop in the maximum packet forwarding rate during route discovery because of the additional CPU overhead required in route lookup and setup of the routing information of the packet. 3Com recommends that you enable RouteDiscovery only for the protocols you use. Increasing the value of the -SR HoldTime parameter will minimize the drop in forwarding rate for these protocols.

If you specify that route discovery is performed only for specific protocol types, you can enhance the performance for other protocols. For example, you can specify that over a given port, route discovery is performed only for AppleTalk and IPX packets using:

```
SETDefault !<port> -SR RouteDiscovery = (AppleTalk, IPX)
```

In this situation, all end-system packets that are not AppleTalk or IPX packets are transmitted as transparent frames over the port.

If the configuration changes, and you no longer want route discovery to take place for specific protocols, you can turn them off using the RouteDiscovery parameter. For example, to turn off route discovery for AppleTalk and IPX packets, use:

```
SETDefault !<port> -SR RouteDiscovery = (NoAppleTalk, NoIPX)
```

You can disable route discovery on a port using:

```
SETDefault !<port> -SR RouteDiscovery = None
```

For more information on end system source routing, refer to "How the Source Route Bridge Works" on page 5-21. For more information on the RouteDiscovery parameter, refer to Chapter 56 in *Reference for NETBuilder Family Software*.

Discovering Routes to an End System

You can discover and optionally save a route to an end system using:

```
DiscoverRoutes <media address> [!<port>] [<timeout (1–120 sec)>]
 [AllRouteExp] [Xid] [Save]
```

where <media address> is [Cmac | Ncmac] %xxxxxxxxxxxx. x is a hexadecimal.

Use Cmac when <media address> is entered in canonical format and Ncmac for noncanonical input.

This command applies only to ports (token ring, FDDI, and HSS running Frame Relay, ATM, ATM DXI, SMDS, X.25, or PPP) with end system source routing enabled with the -SR RouteDiscovery parameter.

The media address should be preceded with the keyword Cmac or Ncmac for canonical or noncanonical format, respectively. The media address should also be preceded by a percent sign (%) and should be 12 hexadecimal digits.

All possible paths to the specified end system are displayed and a preferred route can be chosen and cached in the routing table.

For example, you can cause the bridge/router to issue a route discovery packet over port 1 to address %080000020003 in canonical format by entering:

```
DiscoverRoutes Cmac %080000020003 !1 30 Save
```

A response to the route discovery will be displayed in 30 seconds. If a route is found, the route traversed to reach the specified destination address is saved in the routing table. If one or more routes exist for the remote system, a prompt appears to request the preferred route to save and to determine whether the route is to be cached as a dynamic or a static route. After the route is saved, you can display it using the SHow -SR AllRoutes command.

For more information about the DiscoverRoutes command, refer to Chapter 1 in *Reference for NETBuilder Family Software*.

Adding, Deleting, and Displaying Static Entries in the Routing Table

Routes to a destination end system are discovered using LLC TEST/XID frames. The route associated with the first TEST/XID response is cached in the routing table until its hold time expires. In some topologies, the route that is cached may not be the optimum route to the destination. Some end systems also cannot respond to TEST/XID frames. In these types of situations, you can configure the preferred route as a static (permanent) route using:

```
ADD !<port> -SR ROUte <media address> [Override] [Dec | Hex]
 [<route> [<largestframesize>]]
```

where:

<media address>	is [Cmac | Ncmac] %xxxxxxxxxxxx. 'x' is a hexadecimal. Use Cmac when <media address> is entered in canonical format and Ncmac for noncanonical input.
<route>	is <ring_number>&<bridge_number>[:<ring_number>] ...
<largest frame size>	is: 0 for 516 bytes 1 for 1,470 bytes 2 for 2,052 bytes 3 for 4,399 bytes 4 for 8,130 bytes (not supported) 5 for 11,407 bytes (not supported) 6 for 17,749 bytes (not supported) 7 for 41,600 bytes (not supported)

For example, to configure a static route on port 2 of the bridge/router to the remote system with the MAC address %080002000001 and the manual override option (if the route configured for an end system address becomes invalidated for any reason, the static route is replaced by a learned route if one exists), enter:

```
ADD !2 -SR ROUte Ncmac %080002000001 Override :55&1:56&2:57
```

To display the learned route associated with a specified end system in noncanonical and hexadecimal format, enter:

```
SHow -SR ROUte Ncmac %080002000001 Hex
```

To remove a static route from the routing table, you must remove it manually, unless you specified the Override option when you added the route. To remove a static route, use:

```
DELete !<port> -SR ROUte <media address>
```

You can display routes from the routing table using:

```
SHow [!<port> | !*] -SR ROUte [[Cmac | Ncmac] %<media address>] [Dec | Hex]
SHow [!<port> | !*] -SR AllRoutes [Dec | Hex] [<route>] [Discover | Static]
   [<count>] <route>: ':'<ring number>'&'<bridge number>.... | Transparent
SHow [!<port> | !*] -SR WanRoutes
```

The SHow -SR ROUte command displays static routes in the routing table.

The SHow -SR AllRoutes command displays dynamically discovered, static, and specific source routes or transparent routes depending on the options selected.

For example, to display all discovered routes in hexadecimal format off port 2 that have traversed bridge number 5, enter:

```
SHow !2 -SR AllRoutes Hex &5 Discover
```

To display all static source routes in decimal format off port 2 that have traversed ring number 55, enter:

```
SHow !2 -SR AllRoutes Dec :55 Static
```

To flush all discovered routes in hexadecimal format off port 1 that have traversed the partial route ring number 55, the bridge number 5, and the ring number 77, enter:

```
FLush !1 -SR AllRoutes Hex :55&5:77 Discover
```

The SHow -SR WanRoutes command displays all learned remote networks (bridge number and ring number) and its associated data link connection identifier (DLCI), individual SMDS address, or X.25 DTE address for the Frame Relay, SMDS, or X.25 port, respectively.

For more information about these parameters, refer to Chapter 56 in *Reference for NETBuilder Family Software*.

If you have a static route in a source route environment, LLC will attempt to use that static route.

Aging Out Entries in the Routing Table

You can adjust the time interval (in minutes) that an inactive route entry can reside in the routing table using the HoldTime parameter. This parameter only affects the dynamically learned routes.

To change the default setting of 15 minutes, use:

```
SETDefault !<port> -SR HoldTime = <minutes>(1–1440)
```

Changing the Token Access Priority

The MinAccessPrior parameter determines the minimum access priority used for outgoing frames on a specified port. The lowest priority is 0; the highest is 6. End systems usually have a low-access priority, while bridges have a medium priority (the default is 4). You can configure a source route bridge that typically handles greater amounts of traffic to obtain the token more often than other end systems by adjusting the MinAccessPrior parameter.

To change the default setting of the MinAccessPrior parameter, use:

```
SETDefault !<port> -SR MinAccessPrior = <number>(0–6)
```

How the Source Route Bridge Works

This section provides conceptual information on the following topics:

- Source route, source route transparent bridging, and source route transparent bridging gateway (SRTG) definitions
- IEEE 802.5 token ring frame format
- Source route transparent bridging gateway concepts
- Route discovery process using ARE or STE frames
- End system source routing
- Routing tables

Definitions

This section provides definitions for source route bridging, source route transparent bridging, and source route transparent bridging gateway (SRTG).

Source Route Bridging

Source route bridging is supported on token ring, FDDI, and the following wide area networks: Frame Relay, ATM, ATM DXI, SMDS, X.25, PPP, and ISDN. Source route bridges connect token ring LANs and enable peer-to-peer and terminal-to-host communications across both LAN and WAN token ring networks.

When source route bridging is enabled, the bridge forwards packets based on a route determined by the end system from which the packet originated. The end system initiating the communication is responsible for dynamically determining and then maintaining information about the route to the destination. The source route information is contained within the frame and indicates the path through an extended network from the source to the destination. Because the end system and not the bridge determines the route, a bridge using source route bridging does not record or learn information about addresses on the surrounding networks in the same way that a transparent bridge does. The exception to this rule is on a Frame Relay, ATM DXI, SMDS, or X.25 interface, where the DLCI, VPI.VCI, SMDS, or X.25 address associated with the remote ring is learned.

Source Route Transparent Bridging

This feature is not supported on model 32x and 52x SuperStack II NETBuilder bridge/routers.

Source route transparent bridging is a combination of transparent and source route bridging. The bridge automatically determines whether a packet should be forwarded using transparent bridging or source route bridging. For example, if the bridge receives a frame with routing information, the bridge performs source route bridging. If the bridge receives a frame without routing information, it performs transparent bridging. Source route transparent bridging is used in topologies in which transparent end systems and source route-only end systems coexist on the same network; source route transparent bridging allows the transparent end systems to communicate with transparent end systems and source route-only end systems to communicate with source route-only end systems.

Source Route Transparent Bridging Gateway

This feature is not supported on model 32x and 52x SuperStack II NETBuilder bridge/routers.

With SRTG, you can connect a source-routed network to a transparent bridging network. The SRTG software provides a translation between source route and transparent bridging domains so that token ring network users can communicate with Ethernet network users using source routing; Ethernet network users can communicate using transparent bridging with token ring network users as though they were on the same LAN. Upon receipt of frames from a source route domain, SRTG translates them into transparent bridging frames and removes the source routing information fields (RIFs). The SRTG software also adds appropriate RIF fields to transparent bridging frames before forwarding them to a source route network.

You can configure your bridge to use transparent bridging only, source route bridging only, transparent and source route bridging simultaneously, or SRTG.

When configuring parallel bridges, 3Com recommends that you configure both bridges in the same bridge mode, either source route or source route transparent, to prevent unexpected blocking of one type of traffic due to the Spanning Tree Protocol. For more detailed conceptual information about SRTG, refer to "Source Route Transparent Bridging Gateway Concepts" on page 5-24.

IEEE 802.5 Token Ring Frame Format Overview

Source route bridging requires that each end system in an extended network dynamically determines and maintains the routing information necessary to communicate with other end systems on remote rings in the network. Each frame transmitted by an end system contains the routing information a source route bridge needs to decide whether to forward the frame to an adjoining ring.

This section describes some of the fields in the IEEE 802.5 token ring frame (shown in Figure 5-9) that are important to a general understanding of the route discovery process. Only the destination and source address fields, as well as the routing information field are discussed; not every field in an IEEE 802.5 token ring frame is discussed.

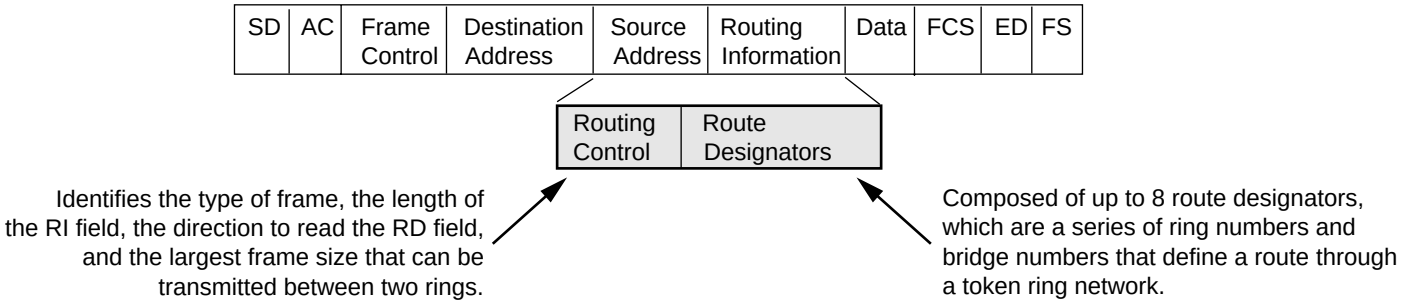


Figure 5-9 IEEE 802.5 Token Ring Frame Format

Destination address field — This 6-byte field identifies the end systems that are intended to receive and copy the frame.

Source address field — This 6-byte field identifies the system from which the frame originated. This field also contains a routing information indicator (RII) bit, which when set to 1, indicates the presence of the routing information field (RIF). If a source route bridge receives a frame with the RII bit = 1, it forwards the frame based on the routing information contained in the route designators (refer to the description of the routing information field). If a source route transparent bridge receives a frame with the RII bit = 0, it forwards the frame based on the destination address using the transparent bridging method.

Routing information field (RIF)

This 0- to 18-byte field contains routing control information and route designators (RD). The routing control information identifies, among other things, the type of source-routed frame, for example, an All Routes Explorer (ARE), Spanning Tree Explorer (STE), or specifically routed frame (SRF).

An ARE frame is transmitted by the source end system to every ring in the extended network. Because the ARE frame is forwarded by a source route bridge to every connected ring, the destination end system receives as many copies of the ARE as there are routes to it. ARE frames are originally transmitted with no route designators; as the frame is forwarded by source route bridges, route designators are added to the frame.

An STE frame is transmitted by the source end system and forwarded only by designated bridges, causing the frame to appear only once on every ring in an extended network. STE frames are originally transmitted with no route designators; as the frame is forwarded by source route bridges, route designators are added to the frame.

An SRF contains the specific route information that allows a source route bridge to forward the frame along a defined network path.

The RD field contains up to eight 2-byte route designators (route descriptors) of ring and bridge number information that describe the path to a destination.

Source Route Transparent Bridging Gateway Concepts

These concepts do not apply to model 32x and 52x SuperStack II NETBuilder bridge/routers.

The SRTG provides translation between source route and transparent bridging domains so that token ring network users can communicate using source routing with Ethernet network users, and Ethernet network users can communicate using transparent bridging with token ring network users. Upon receipt of frames from the source route domain, SRTG translates them into transparent bridging frames by removing the source route information fields (RIFs). SRTG adds appropriate RIF fields to transparent bridging frames before forwarding them to the source route network.

Spanning Tree Considerations

Two different spanning tree schemes exist for transparent bridging and for source routing. In transparent bridging, the Spanning Tree Protocol (STP) ensures that only one active path between any two stations exist in the network. In source routing, STP selects the bridges to forward the spanning tree explorer frames.

When both source route and transparent bridging domains are connected using SRTG, multiple gateways may be installed in parallel, either by mistake or on purpose, creating loops in the network topology. To eliminate loops and

ensure a single active path between two stations, SRTG fully participates in the transparent STP.

To ensure compatibility with IBM 8209 or 8229 LAN bridges, the spanning tree entity on 3Com SRT gateways generates Bridge Protocol Data Units (BPDUs) as STE frames with a destination address set to the group address and the RII bit set. As shown in Figure 5-10, SRTG detects and breaks loops when there are multiple paths between SR and TB domains.

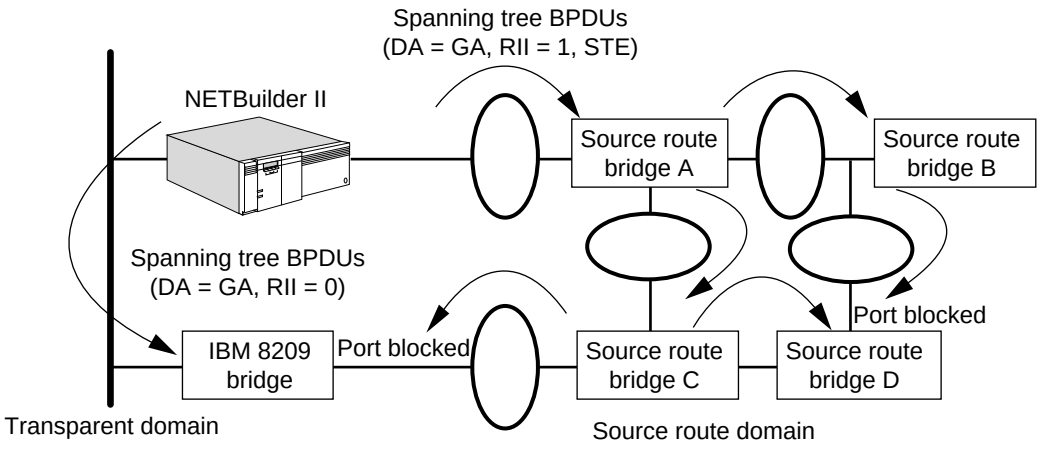


Figure 5-10 Spanning Tree Loop Detection by SRTG

Source route bridge A forwards the spanning tree BPDUs according to the source route spanning tree path to bridge B and C without recomputing the spanning tree algorithm. Bridge B forwards the BPDUs to bridge D, which drops them because the port is in the blocking state. Bridge C forwards BPDUs to bridge D and the IBM 8209 bridge. Bridge D receives them but does not perform the spanning tree computation nor forward them because its other port is in the blocking state. When the IBM 8209 bridge receives the BPDUs, it detects a loop and blocks the source routing port.

When the source routing port goes into a blocking state, all types of frames (ARE, STE, and SRF) are not forwarded. This behavior complies with the transparent bridging behavior but differs from pure source route bridging. When the primary and secondary SRT gateways change their role due to topology changes anywhere in the transparent bridging network (a primary gateway becomes secondary and vice versa), stations using the existing path may experience session disruption. Using parallel SRT gateways does not provide load balancing but does provide a backup path if the primary SRT gateway fails.

Packet Handling between Domains

When a packet is bridged from a source route domain to a transparent bridge domain using SRTG, the source route field of the frame is removed as shown in Figure 5-11. The RIF of the originator is cached with the direction bit in the route control field inverted for use by subsequent return traffic.

When a packet is bridged from a transparent bridge domain to a source route domain using SRTG, the packet is forwarded using the associated routing information from the source route table if the destination is known. If the destination is not known, the packet is immediately forwarded as an STE frame.

The SRT gateway acts as a surrogate source routing station on behalf of all transparent bridge stations and uses a virtual ring number (set with the -SR GatewayVRing parameter) for its transparent bridge domain. Whenever bridging packets from the transparent bridge to source route domain, SRTG adds the virtual ring number and its own bridge number to the source route information of the destination station retrieved from the source route table. From the point of view of a source routing station, the entire transparent bridge LAN appears as a single source routed ring as shown in Figure 5-11.

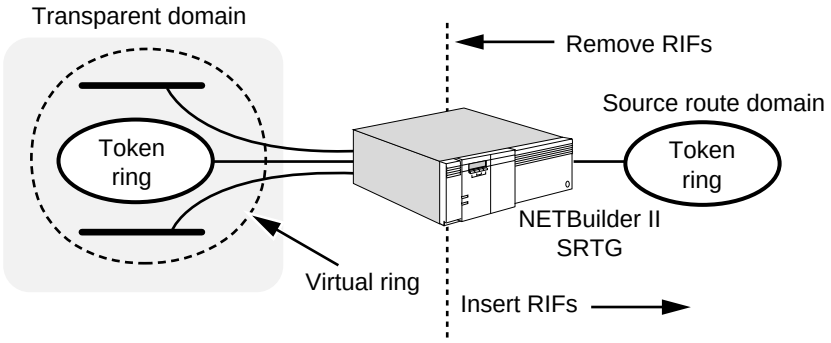


Figure 5-11 Virtual Ring and Frame Translation

Source Route to Transparent Bridge Domain Packets. SRTG handles ARE, STE, and SRF frames as described in Table 5-3.

Table 5-3 Source Route to Transparent Bridge Domain Packet Handling

Frame Type	How Handled
ARE	An ARE frame with a group address (broadcast or functional) is forwarded onto the transparent bridge domain, but its source route information is not cached.
	An ARE frame with a specific destination address is forwarded onto the transparent bridge domain only when the destination address does not exist on the source route domain. If the source address is not found in the source route table, SRTG creates one. If the source address is found in the source route table, SRTG updates the old entry with the new route information if different.
	An ARE frame is copied as many times as available paths, and traverses all possible paths overriding the spanning tree configuration, causing SRTG to receive multiple copies of a packet if there are multiple paths. To prevent multiple copies from being forwarded to the transparent bridge domain, SRTG saves the source route from the first copy, considers it the optimal route, and discards the subsequent copies.
STE	An STE frame with a group (broadcast or functional) address is forwarded onto the transparent bridge domain, but route caching does not occur.
	An STE frame with a specific destination address is forwarded onto the transparent domain if the target station is not on the same source route domain. When forwarding a unicast STE frame, SRTG creates a new entry if an entry is not found in the source route table and the target station is known to exist on the transparent domain. If a source route entry already exists in the source route table, SRTG updates the entry but marks it as temporary. Because routes learned from STE frames may not be optimal, they are overwritten by any subsequent SRF frame from the source station.
SRF	Regardless of the destination address, SRTG forwards any SRF frame to the transparent bridge domain when RIF indicates the virtual ring.
	SRTG checks the ring out number and if it matches the SRTG virtual ring number, the SRF frame is translated and forwarded to the transparent bridge domain. If the destination station is already learned, the SRF frame is sent to a specific port. Otherwise, the SRF frame is flooded on all source route and SRTG ports except the source port. If no source route entry associated with the source station is found, SRTG creates one. If an entry is found but is temporary, SRTG updates the old entry and removes the temporary flag.

Transparent Bridge to Source Route Domain Packets. When SRTG receives a packet from transparent bridge domain, it forwards the packet using the associated routing information from the source route table if the destination

address exists in the database. If the destination does not exist in the source route table, SRTG immediately forwards the packet in a STE frame to reduce possible excessive traffic. Whether the destination address exists or not, SRTG adds the virtual ring number configured for the transparent bridging domain to the RIF field retrieved from the source route table.

Frame and Address Conversion

This section focuses on LAN-specific media (Ethernet, token ring, and FDDI) and the different packet formats. Frame conversions are necessary because Ethernet supports two different formats: Ethernet Version II frame and IEEE 802.3 frame.

In a source route token ring network, there are two ways to form a packet. IEEE 802.2 (LLC) encapsulation is used for LLC2 and NetBIOS packets while other protocols, such as IP, use SNAP encapsulation. To ensure compatibility with IBM's 8209 implementation of delivering bridged packets to a target station in its expected format, SRTG keeps track of the encapsulation format of each Ethernet station.

Ethernet 802.2 Conversion to and from Token Ring 802.2. Because both Ethernet and token ring supports IEEE 802.2 encapsulation, conversion of Ethernet 802.2 frames to token ring 802.2 encapsulation is a simple task. SRTG removes the length field and adds the RIF field when it converts frames from Ethernet 802.2 to Token Ring 802.2. SRTG removes the RIF field and adds the length field (padding may be required for small frames) when it converts frames form Token Ring 802.2 to Ethernet 802.2. These frame conversions are shown in Figure 5-12.

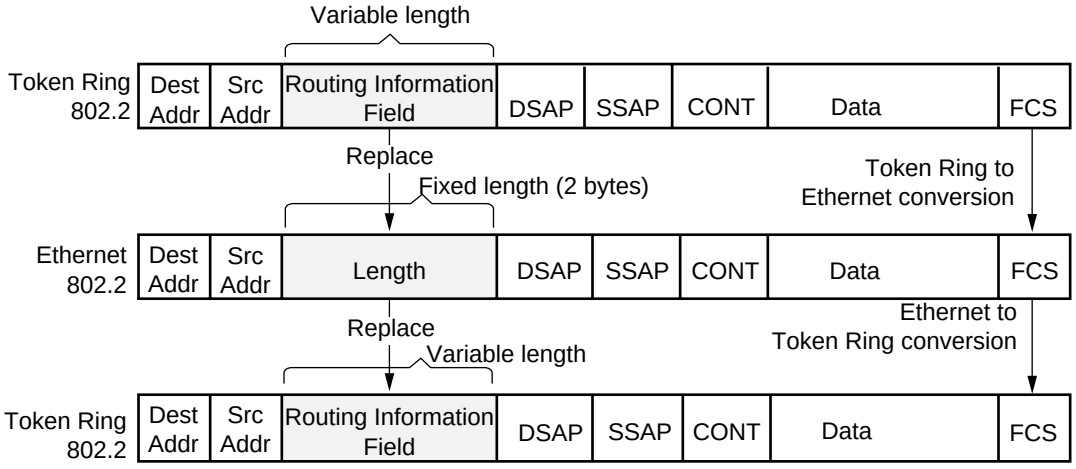


Figure 5-12 Ethernet 802.2 Conversion to or from Token RIng 802.2

LLC-based Token Ring Conversion to and from Ethernet II. To support the coexistence of both Ethernet II and Ethernet 802.3 frames on the same LAN, SRTG provides options in the -SR GatewayControl parameter to translate LLC-based packets to Ethernet II frames as follows:

- The destination service access point (DSAP) field in the token ring 802.2 frame must be a multiple of 4 (for example, 00, 04, 08, and so forth), except 0xBC and 0xE0, which are reserved for Banyan VINES and IPX, respectively.

- If SRTG is configured with the NoAutoMode setting, SRTG does not keep track of the encapsulation type of each transparent bridge station. The final encapsulation format is determined by the IeeeMode or EtherMode setting of the GatewayControl parameter. If it is set to EtherMode, Ethernet II encapsulation with type 0x805D is used. If it is set to IeeeMode, LLC-based packets are translated into the IEEE 802.3 format.
- If SRTG is configured with the AutoMode setting, different packet translation rules are used for known and unknown stations. For known stations, the IeeeMode | EtherMode setting is ignored and the encapsulation format learned for each station is used. For unknown stations, LLC2-based packets are translated based on the IeeeMode | EtherMode setting.

This resulting frame looks like an Ethernet II format. LLC data are not placed inside an 802.3 frame but placed into an Ethernet Version II frame whose type is specified as 0x80D5 and shown in Figure 5-13.

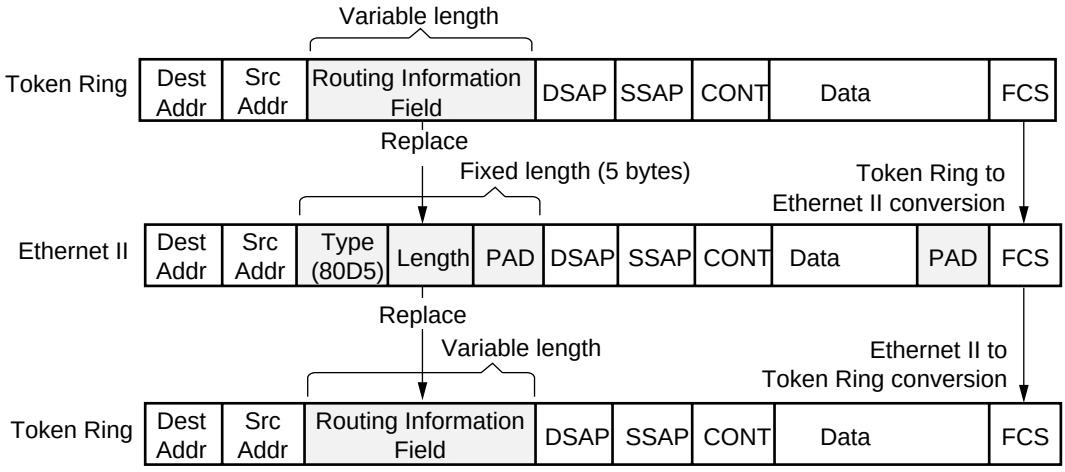


Figure 5-13 LLC-based Token Ring Conversion to and from Ethernet II

Maximum Frame Size

The maximum frame sizes used by Ethernet and token ring networks are different. To solve this frame length mismatch, SRTG automatically sets the largest frame size bit in the Route Control field to 1450 octets whenever it forwards frames to the token ring network (see Table 5-2). SRTG drops data packets from token ring or FDDI if the packets are larger than the Ethernet maximum frame size.

Route Discovery Process

An end system (PCs and workstations) with source route support installed can dynamically determine the routing information it needs to communicate with other end systems on remote rings interconnected by source route or source route transparent bridges. The route discovery process consists of the exchange of messages between the source and the destination end systems. Because no current standard for route discovery exists, the method that the end system uses may be protocol specific; therefore, a general description of the end system route discovery process is provided with details about how the 3Com bridge/router participates in the route discovery process.

The end station sends an explorer packet (for example, a TEST or XID frame, or a protocol-specific frame, in an ARE or STE) with the destination address in the header. If an ARE frame is transmitted by the source system and the source route bridge receives it, the source route bridge adds its bridge number and ring number of the adjoining ring to the RD fields, and forwards the frame to all of its source route bridging interfaces. The next source route bridge repeats the same process until the destination system recognizes its MAC address in the destination address field of the header and copies the frame.

If multiple paths to the destination system exist, the destination will receive as many explorer frames as there are paths and must respond to each explorer frame. The destination system responds to the ARE (each and every one) by sending an specifically routed frame (SRF). The frame contains all the routing information needed to forward the frame back to the source. In fact, when the source route bridge receives an SRF, it forwards the frame according to the embedded source route information in the RD fields. When the source system recognizes its MAC address, it copies the frame and uses the routing information within the frame for all subsequent communications with that destination system.

If an STE frame is transmitted by the source system and the source route bridge receives it, the source route bridge adds its bridge number and ring number of the adjoining ring to the RD fields. The source route bridge only forwards the frame to the source route bridging interfaces that are not blocked because of the Spanning Tree Protocol, resulting in only one STE frame appearing on each ring. Each source route bridge in the spanning tree path follows the same procedure.

When the destination system recognizes its MAC address in the destination address field of the header, it copies the frame and responds to the STE by sending an ARE. The ARE frame is used so that all possible routes to the source can be found. On the return trip to the source system, the source route bridge forwards the ARE frame to all source route interfaces. When the source system recognizes its MAC address, it copies the frame (multiple responses may be received) and uses the routing information from the preferred ARE for all subsequent communications with that destination system.

When the 3Com bridge/router functions as an end system, it initiates the route discovery process by sending a TEST/XID STE frame. Upon receiving the frame, the destination system sends an ARE frame as described in the previous paragraph, except that the 3Com bridge/router caches the first ARE that it receives and discards all the other ARE responses.

End System Source Routing

Route discovery for end system source routing is supported on token ring and FDDI networks and wide area networks using PPP, Frame Relay, ATM DXI, SMDS, or X.25. Using end system source routing, the router acting as an end system can discover end systems not already present in the end system routing table. This is useful in situations in which the router receives a packet, but does not have a source route to the destination station on the source route network. If route discovery is enabled, the router determines the best route to the destination station by initiating a route discovery process and caching the discovered route in the routing table.

You normally use route discovery in configurations where the router is attached to a source route bridged environment. To enable routing in a source route environment, you must configure route discovery on the port directly connected to the source route bridged domain.

For any given port, you can configure the router to initiate route discovery for any combination of the following types of routing protocol packets:

- AppleTalk
- CLNP (OSI)
- DECnet
- IP (route discovery using an ARP packet)
- IPX
- LLC2 (for tunneling LLC2 packets over IP)
- VINES

You can also configure a port to initiate route discovery for DLTest packets.

Routes to a destination end system are discovered using LLC TEST/XID command frames. The route associated with the first TEST/XID response is cached in the routing table until its hold time has expired. Enabling route discovery allows end systems located in transparent-only, source route-only, or source route transparent environments to be reached.

Routing Tables

You can access the routing table of end system by entering the SHow -SR AllRoutes command. For complete information on this parameter, refer to Chapter 56 in *Reference for NETBuilder Family Software.*

A source route bridge forwards a packet based on a route determined by the end system from which the packet originated. Routes are discovered on ports where the -SR RouteDiscovery parameter is enabled for one or more protocol packets. The routes learned by the bridge are cached in a routing table.

The two types of routing table entries are learned (dynamic) entries and user-assigned (static) entries.

- Learned (dynamic) entries are entries that the router learns from route discovery packets received from communicating end systems. The learned entries are subject to dynamic changes or deletion at intervals determined by the -SR HoldTime parameter (the default is 15 minutes).
- User-assigned (static) entries are entries assigned using the ADD -SR ROUte command. The static entries can be changed or deleted only through the ADD or DELete commands. These entries also are referred to as permanent entries.

6 CONFIGURING IP ROUTING

This chapter describes the procedures for configuring your system to perform Internet Protocol (IP) routing. It describes how the router works and gives guidelines for operating, managing, and troubleshooting it.

For conceptual information, refer to "How the IP Router Works" on page 6-38.

Configuring a Basic IP Router

The procedure in this section describes the minimum number of steps required to configure your system to route IP packets. Depending on your network requirements, you can use the default values of the parameters in the various services, or you can further configure the router according to later sections in this chapter.

To configure the IP router, you must set parameters in the RIPIP Service if your network uses the Routing Information Protocol (RIP) or in the OSPF Service if your network uses the Open Shortest Path First (OSPF) routing protocol. If you are using OSI routing for an IP environment, you must configure Integrated IS-IS parameters (IISIS).

The IP parameters enable the routing function and configure the networks connected to the router. The following information describes how to configure IP parameters.

Configuring for Local Area Networks and Point-to-Point Links

Use this procedure to configure basic IP routing over LAN ports and Point-to-Point Protocol (PPP) links.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the ports and paths of your router according to the information in Chapter 1 and log on to the system with Network Manager privilege.
- Become familiar with the protocols supported by the router. This chapter describes the protocols only when the explanation is necessary for interpreting the parameters and screen displays used in the router software.
- Obtain an IP address for each port you want to configure (PPP links can be unnumbered).

  The router fully supports variable length IP subnetting. For information on IP addresses, subnets, subnet masks, and variable length subnet masks, refer to Appendix D. The router also supports multiple IP subnets. For more information on this feature, refer to "Configuring Multiple IP Networks/Subnets" on page 6-7.
- Obtain a subnet mask for the given network address, if it is different from the default or "natural" mask.

Procedure

To set up a basic configuration for your IP router, follow these steps:

1. Assign an IP address for each LAN port that will route IP using:

```
SETDefault !<port> -IP NETaddr = <IP address> [<subnet mask>
  [Ones | Zeros [MTU]]] | UnNumbered
```

CAUTION: *An IP address assigned to port 0 is considered the IP address for all the interfaces. As a result, the bridge/router behaves as an IP host for Telnet access and network management and stops routing IP packets. Do not configure an IP address for port 0 if you want to route IP packets.*

2. Assign an IP address or the value UnNumbered to each wide area port using PPP as the serial line protocol using:

```
SETDefault !<port> -IP NETaddr = <IP address> [<subnet mask>
  [Ones | Zeros [MTU]]] | UnNumbered
```

   If you are configuring your IP router to route over the phone line gateway (PLG) protocol, there are no prerequisites.

   PPP does not require that you assign an IP address to each wide area port. Before configuring your IP router to route over PPP, determine if you want to assign an IP address to each wide area port. (Refer to "Related Information" on page 6-3). If you do not want to assign an IP address to a wide area port, you must set the value of the -IP NETaddr parameter to UnNumbered. An advantage of not assigning an IP address to each wide area port is that you conserve valuable network and subnet numbers.

3. If you are going to be running OSPF as the routing protocol over dial-up circuits, configure a demand interface circuit using:

```
SETDefault !<port> -OSPF DemandInterface = Enable
```

CAUTION: *Do not configure any interface on any router in a single OSPF area as a demand circuit (DC) interface unless all routers in that area have been upgraded to at least software version 8.3.*

   With this setting, the router negotiates with the neighbor at the other end of the point-to-point link. If the neighbor agrees that the point-to-point link is a demand circuit, the router suppresses sending OSPF hello packets and routing refresh information, allowing the data link connection to be closed when not carrying application traffic. For the demand circuit to be cost-effective, make sure that it is isolated from as many topology changes as possible because topology changes bring up the interface.

   For more information, refer to "Reducing Network Costs Using Demand Interface Circuits" on page 6-53.

4. Enable the dynamic routing protocols for IP routing using RIPIP, OSPF, or IISIS.

   - To enable RIP operation on a specified port, set the CONTrol parameter in the RIPIP Service (using its TAlk and Listen values) as follows:

```
SETDefault !<port> -RIPIP CONTrol = ([TAlk | NoTAlk],
 [Listen | NoListen], [Poison | NoPoison], [TRigger |
 NoTRigger], [NetAdvUnn | SubnetAdvUnn], [SubnetBcast |
 All1sBcast], [Aggregate | NoAggregate], [DeAggregate |
 NoDeAggregate], [DynamicNbr | NoDynamicNbr],
 [FullMesh | NonMesh]
```

Setting the CONTrol parameter to the TAlk and Listen values enables the router to send and receive routing information with other routers using RIP.

You can also configure RIPIP for networks with variable length subnet masks using an aggregate/deaggregate scheme or the range table mask scheme. For more information, refer to "Configuring RIPIP for Networks with Variable Length Subnet Masks" on page 6-10.

If you set the value of the -IP NETaddr parameter to UnNumbered for a PPP serial link, make sure that you set the value of the -RIPIP CONTrol parameter to NetAdvUnn or SubnetAdvUnn depending on your network configuration. For more information about NetAdvUnn and SubnetAdvUnn, refer to Chapter 47 in *Reference for NETBuilder Family Software*.

- To enable OSPF on a specified port, set the CONTrol parameter in the OSPF Service using:

  ```
  SETDefault !<port> -OSPF CONTrol = Enable
  ```

  After OSPF is enabled, the router will exchange routing information with other routers using OSPF.

- To configure IISIS for Dual IP and Open System Interconnection (OSI) mode, enter:

  SETDefault -IISIS CONTrol = Enable

5 Enable IP routing by entering:

SETDefault -IP CONTrol = ROute

To complete the configuration for PPP links, refer to Chapter 34.

Related Information

A serial line running PPP can support IP routing without the assignment of IP subnets. This feature is called *unnumbered links*. An unnumbered PPP link is useful only between two routers; in other words, it cannot connect a router to a host.

You must configure a serial line running PPP as an unnumbered link using the -IP NETaddr parameter before the unnumbered link takes effect. When an update is sent over an unnumbered PPP link, the source IP address is borrowed from another interface. For this reason, a router must have at least one IP address configured.

When RIP is run over a PPP link, both ends of the link must be either un-numbered or numbered with the same IP subnet. Half-numbered links, or links with inconsistent IP subnets on both ends, are considered a configuration error.

When OSPF or IIISIS is run over unnumbered PPP links, no limitation exists in the way that the PPP link may be configured. Either end of the link can be numbered independently, or both ends can remain unnumbered. If both ends are numbered, they need not be on the same IP subnet nor have the same subnet masks.

You do not need to assign a network number to a Frame Relay cloud if you are using IISIS.

Configuring for Wide Area Networks

IP routing over Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), X.25, and ATM is supported over fully meshed, partially meshed, and nonmeshed topologies.

If you plan to use RIP over Frame Relay, ATM DXI, X.25, or ATM in a partially meshed or nonmeshed topology, you must enable the next-hop split horizon feature by having a list of neighbors and you must set -RIPIP CONTrol to NonMesh. The list of neighbors can be dynamically generated by the system or manually configured.

If you plan to use OSPF over Frame Relay, ATM DXI, X.25, or ATM in a partially meshed or nonmeshed topology, you can create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. You must run the OSPF NonMesh mode over the Frame Relay, ATM DXI, or X.25 cloud.

If you plan to use IISIS over Frame Relay, ATM DXI, or X.25 in a partially meshed or nonmeshed topology, no additional configuration is necessary. Regardless of the type of topology, when you use IISIS, you do not need to assign a network number to the Frame Relay, ATM DXI, or X.25 cloud.

For complete information on configuring IP routing over Frame Relay, ATM DXI, X.25, and ATM, including a discussion of fully meshed, partially meshed, and nonmeshed topologies, virtual ports, and next-hop split horizon, refer to Chapter 42, Chapter 45, and Chapter 47. For information on the number of virtual ports supported per platform, refer to Table 1-1.

Routing IP over Switched Multimegabit Data Service (SMDS) is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your IP router to perform routing over SMDS, refer to Chapter 44.

For information on configuring PPP and PLG, refer to Chapter 34. For information on wide area networking using ISDN, refer to Chapter 35.

Verifying the Configuration

To verify the configuration, examine network devices and send packets from one network to another using the PING command.

Examining Network Devices

To examine the status of the IP router, follow these steps:

1. Display information on the attached networks by entering:

   ```
   SHow -IP NETaddr
   ```

2. Determine which stations or networks are reachable from the router by entering:

   ```
   SHow -IP AllRoutes
   ```

3. Display information from the Address Translation Table by entering:

   ```
   SHow -IP ADDRess
   ```

Checking with PING To use the PING command to check if packets are being forwarded, use Figure 6-1 as an example, and follow these steps:

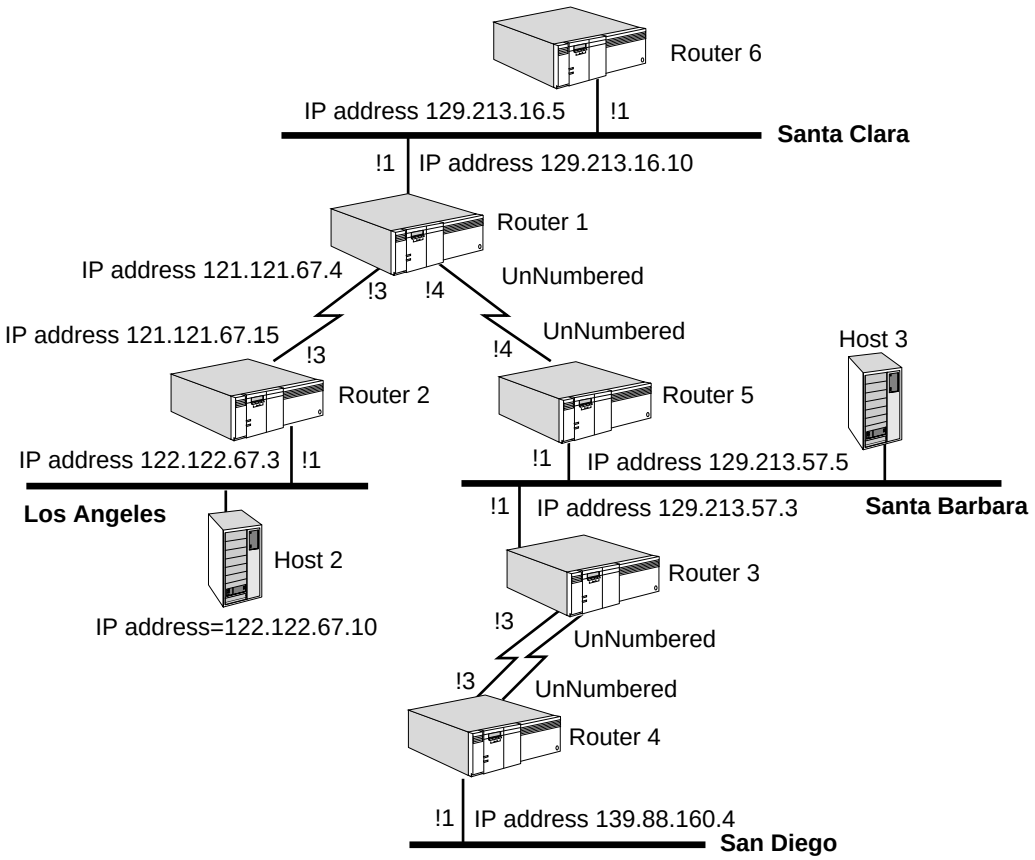


Figure 6-1 Wide Area Router Configuration

1 Determine whether all of the Router 1 network interfaces are up and running by entering the following commands from router 6:

a To check port 1, enter:

```
PING 129.213.16.10
```

If you do not specify the amount of time in seconds that the bridge/router should attempt to ping a device, the bridge/router assumes 20 seconds. For more information on the PING command, refer to Chapter 1 in *Reference for NETBuilder Family Software*.

b To check port 3 of router 1, enter:

```
PING 121.121.67.4
```

If a port is operational, a message similar to the following appears after each PING command:

```
pinging ... 121.121.67.4 is alive
```

If a port is not operational, a message similar to the following appears:

```
pinging ... 121.121.67.4 is not responding
```

If this message appears, check the network connection to see if the cables are properly connected. Contact your network supplier or 3Com for help if you still cannot determine the cause of the problem.

2 After you determine that each port is operational, check that the router can forward packets from one network to another.

   a Enter the PING command on router 6 to check if it can communicate with host 2:

   PING 122.122.67.10

   If host 2 is operational and the router functions properly, a message similar to the following appears:

   ```
   pinging ... 122.122.67.10 is alive
   ```

   b If you do not get this message, use the TraceRoute command on router 6 to trace a path to your intended destination. Specify the IP address of the destination you want to trace.

   c Follow the steps in "Checking the Overall Status" on page 6-6 to verify the following items:

   - Port 1 is properly configured on router 1.
   - Port 3 is properly configured on router 1.
   - Port 1 is properly configured on router 2.
   - Port 3 is properly configured on router 2.
   - The address used in the PING command is the correct address of host 2.
   - Routing is enabled on router 1, router 2, and router 6.
   - The routing protocol is properly configured.

If you cannot determine the cause of the problem, contact 3Com or your network supplier for help.

Getting Statistics

After you have followed the necessary setup and checking procedures using the PING command, examine the statistics by entering:

SHow -SYS STATistics -IP

You can collect statistics for a specific period by using the SampleTime and STATistics parameters. For more information, refer to Chapter 58 in *Reference for NETBuilder Family Software*.

Checking the Overall Status

The following information pertains to checking the status of the router.

Procedure

To check the overall status of the IP router, follow these steps:

1 Examine the path configurations by entering:

SHow -PATH CONFiguration

2 Examine the port configurations by entering:

SHow -PORT CONFiguration

3 Examine the IP configurations by entering:

SHow -IP CONFiguration

4 Examine the RIPIP configurations by entering:

SHow -RIPIP CONFiguration

5 Examine the OSPF configurations by entering:

```
SHow -OSPF CONFiguration
```

6 Examine the ARP configurations by entering:

```
SHow -ARP CONFiguration
```

7 Examine the BGP configurations by entering:

```
SHow -BGP CONFiguration
```

8 Examine the ISIS configurations by entering:

```
SHow -ISIS CONFiguration
```

Related Information

You may also want to verify that routing protocols and static routes are configured properly by using the TraceRoute command. For example, at router 1 in Figure 6-1, you can trace the route between routers 1 and 4, which will verify that routers 5 and 3 relayed the packets sent by router 1. For complete information on the TraceRoute command, refer to Chapter 1 in *Reference for NETBuilder Family Software*.

Customizing the IP Router

After you set up and check the configuration of the basic IP router, it is ready to perform packet routing. If desired, you can further customize your IP router by doing the following tasks:

- Configure UDP Broadcast Helper, if necessary
- Configure multiple subnets
- Configure logical networks over IP
- Configure RIPIP for networks with variable length subnet masks
- Configure static routes
- Configure packet filtering
- Configure routing policies
- Use the IP security parameters
- Configure interautonomous system routing using the Border Gateway Protocol (BGP)

Configuring UDP Broadcast Helper

UDP Broadcast Helper allows applications in the TCP/IP stack to forward broadcast packets through a gateway and to another network segment. For information on configuring UDP Broadcast Helper, refer to Chapter 20.

Configuring Multiple IP Networks/Subnets

Your IP router supports multiple IP subnets. You can configure more than one IP network or subnet on any media.

The procedure for configuring multiple IP subnets on all interfaces is the same. The following paragraphs present an example of configuring multiple IP subnets on an Ethernet network.

Figure 6-2 is an example of a topology where two IP networks can be configured on an Ethernet. The first IP network is called 128.1.0.0; the second is called 128.2.0.0. Use the following example to configure these IP networks on Ethernet 1. Configure the two networks on port 1 of router 1.

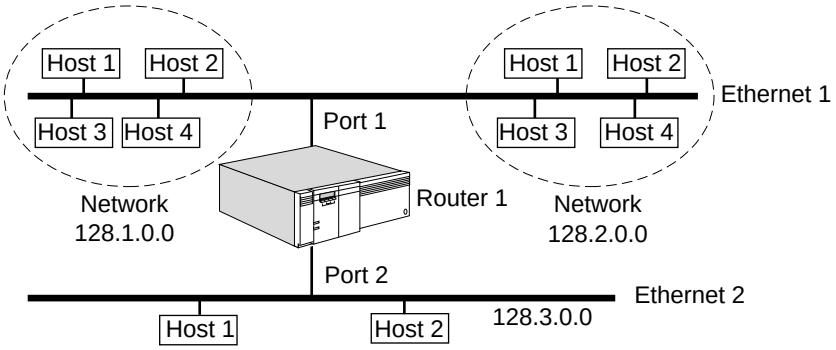


Figure 6-2 Two IP Subnets Configured on the Same Ethernet

To configure the two networks on port 1 of router 1, follow these steps:

1 Configure the first network.

The first address that you configure is known as the primary address. This address is indicated by an asterisk when you enter the SHow -IP NETaddr command. For example, to set up network 128.1.0.0 with the IP address of 128.1.0.5, enter:

```
SETDefault !1 -IP NETaddr = 128.1.0.5
```

2 Configure any subsequent networks.

To configure a subsequent network, for example, network 128.2.0.0 with the IP address of 128.2.0.5, enter:

```
ADD !1 -IP NETaddr 128.2.0.5
```

To delete an address, use:

```
DELete !<port> -IP NETaddr <IP address>
```

In the topology shown in Figure 6-2, the systems on Ethernet 1 have been divided into two IP networks. Direct communication takes place among the hosts in network 128.1.0.0 and among the hosts in network 128.2.0.0. However, router 1 must forward packets between a host on network 128.1.0.0 and a host on network 128.2.0.0.

Related Information

The ability to configure multiple IP subnets gives you the following advantages:

- You can maximize the use of your network media.

  The structure of the IP address limits the number of systems that can be addressed on an IP network. Configuring multiple IP subnets on a single network media allows you to increase the number of systems that you can address on a single media.

- You can break down systems on the network media into subsets of virtual private networks (VPNs). Direct communication occurs within these VPNs.

You must factor the advantages of being able to configure multiple IP subnets against the fact that traffic on a segment containing multiple IP subnets can increase significantly. Traffic from one network to another on the same segment must first go to the router then back out on the same segment.

Configuring Logical Networks over IP

You can assign the same IP address to several ports, and bridge among those ports while routing to other ports, by creating multiple logical networks (MLN). Software version 8.3 supports MLN over IP only on Ethernet media.

MLN offers the following benefits for IP routing:

- Simplifies network protocol address administration on large networks. Instead of configuring each port individually, you need to configure only the group port.
- Reduces the number of IP addresses you need, making more efficient use of the limited hierarchical IP address space.
- Allows you to move stations from one LAN to another LAN without having to reassign hierarchical IP addresses, as long as both LANs belong to the same logical network.
- Allows you to integrate a number of bridged networks by routing them from the bridged environments (configured as logical networks) across a LAN or WAN backbone.
- Allows you to restrict broadcasts by grouping the target range into a port group since bridging of a logical network occurs only within the port group.

In Figure 6-3, ports 1, 2, and 3 and the LANs attached to them have been grouped together into logical network V1 by entering:

```
ADD !V1 -PORT LogicalNET ETHernet 1,2,3
```

Ports 5 and 6 have been grouped into logical network V2 by entering:

```
ADD !V2 -PORT LogicalNET ETHernet 5,6
```

Port 4 is an ordinary port that does not belong to a logical network.

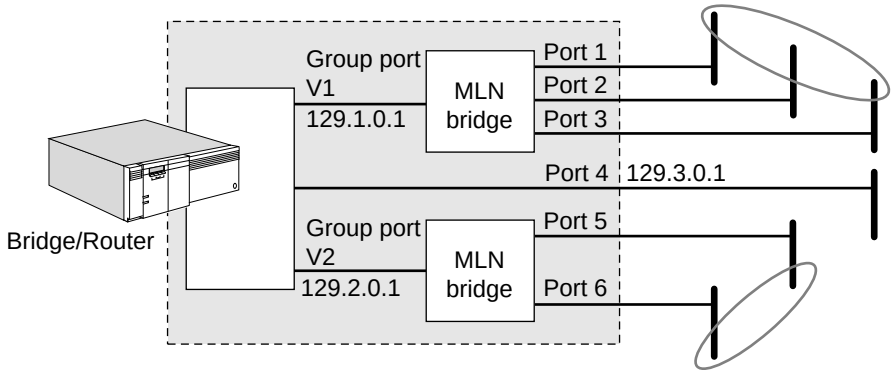


Figure 6-3 Logical Networks over IP

For more information about configuring group ports, refer to "Configuring Multiple Logical Networks" on page 1-22.

You can now assign IP addresses to ports V1, V2, and 4, for example:

```
SETDefault !V1 -IP NETaddr = 129.1.0.1
SETDefault !V2 -IP NETaddr = 129.2.0.1
SETDefault !4 -IP NETaddr = 129.3.0.1
```

You cannot assign IP addresses directly to the member ports 1, 2, 3, 5, and 6.

You can also assign subnet masks and enable dynamic routing protocols, as explained in "Configuring a Basic IP Router" on page 6-1, or customize the router in the other ways explained in this chapter. Ports 1, 2, and 3 share the IP address and other IP properties that you assign to port group V1, and ports 5 and 6 share the IP address and other IP properties that you assign to port group V2.

With this configuration, traffic between any port in group V1 and any port in group V2 is routed. Traffic between group V1 and port 4, or between group V2 and port 4, is also routed.

Traffic among ports within a port group is bridged, not routed. Traffic for the network protocol configured on group port V1 is bridged among ports 1, 2, and 3 (as indicated by the MLN bridge in the figure). Traffic on group port V2 is bridged between ports 5 and 6. To configure this bridging, you must enable global bridging and per-port transparent bridging on all member ports. For more information, refer to "Bridging over Multiple Logical Networks" on page 3-2.

Adding a Static IP Address

When you add a static IP address for a group port using the ADD -IP ADDRess parameter, and the group port has a member in one of the 6-port Ethernet cards, then the IP traffic will not be forwarded to those member ports. *However, if the MAC address is learned dynamically through the Address Resolution Protocol (ARP) on those member ports, then traffic is forwarded.*

Configuring RIPIP for Networks with Variable Length Subnet Masks

Using RIP version 1 in software release 8.0 and later, you can use variable length subnet masks in your network and use RIPIP as the routing protocol. By using variable length subnet masks, you can eliminate the need for additional IP addresses, which are increasingly difficult to obtain because of the rapid growth of IP-based networks.

Because RIP packets do not carry explicit subnet information, you can configure the router that receives or transmits a routing update to determine the appropriate subnet mask. You can implement the following schemes:

- Aggregate/deaggregate scheme

  The aggregate/deaggregate scheme primarily addresses the transmitter function and how the transmitting router translates routes from one mask length to another so as not to confuse receiving routers.

- Range table mask scheme

  The range table mask scheme addresses the receiver operation and how the receiving router interprets an incoming route advertisement and assigns an appropriate subnet mask to it.

These two schemes are independent of each other; they can be used individually or together.

The routes can be RIP-learned routes, directly attached networks, static or dynamic routes learned from other protocols (if the appropriate policy is enabled).

Using the Aggregate/Deaggregate Scheme

To enable the aggregate/deaggregate scheme on a specified (outgoing) port, use:

```
SETDefault !<port> -RIPIP CONTrol = Aggregate
SETDefault !<port> -RIPIP CONTrol = DeAggregate
```

Do not use the aggregate/deaggregate scheme with unnumbered PPP links. Use the SubnetAdvUnn | NetAdvUnn values with the -RIPIP CONTrol parameter or use the range table mask scheme.

Procedure When both values are selected, RIPIP performs route conversion using the following algorithm (steps 1, 2, 3a, and 3d comprise the algorithm for software release 7.1 and earlier).

To use the Aggregate/Deaggregate scheme, follow these steps:

1 If it is a host route (with mask 255.255.255.255), propagate as is.

2 If it is a network route using the natural mask (for example, 10.0.0.0 255.0.0.0), propagate as is.

3 If it is a subnet route, do the following:

When NoAggregate is selected, step 3b is skipped. When NoDeAggregate is selected, step 3c is skipped.

a If the outgoing interface is not subnetted, or if the outgoing interface belongs to a different IP network number, then zero out the bits in the subnet portion and propagate the natural IP network number (aggregate to the natural mask).

For example, if the routing update is sending 10.1.0.0 to 11.1.0.0, then the subnet portion is zeroed out and the natural IP network number (10.0.0.0) is propagated.

b If the outgoing interface has the same IP natural network number as the route being propagated and if the mask of the route is longer than the mask of the outgoing interface, adopt the shorter mask and zero out all the bits in the host field (aggregate to a shorter mask).

This step of the algorithm is used if Aggregate is selected on the outgoing port. For example, if the mask of the route is 255.255.255.0 and the mask of the outgoing interface is 255.255.0.0, the shorter mask is used and the host field bits are zeroed.

c If the outgoing interface has the same IP natural network number as the route being propagated and if the mask of the route is shorter than the mask of the outgoing interface, adopt the longer mask and convert the route into a series of route advertisements that cover the full address space.

This step of the algorithm is used if DeAggregate is selected on the outgoing port. For example, if the route 10.1.0.0 255.255.0.0 is sent to neighbors with a longer mask (255.255.255.0), the route is expanded into a sequence of subnets 10.1.0.0 255.255.255.0 to 10.1.255.0 255.255.255.0. In this example, each route from the shorter side translates into 256 routes on the longer side.

d If a, b, and c are not true, propagate as is (the outgoing interface is subnetted from the same IP network as the receiving interface and has the same mask).

Related Information Use the aggregate/deaggregate scheme in simple network topologies; for example, you may have a single router between the corporate backbone and stub networks (or remote offices) as shown in Figure 6-4.

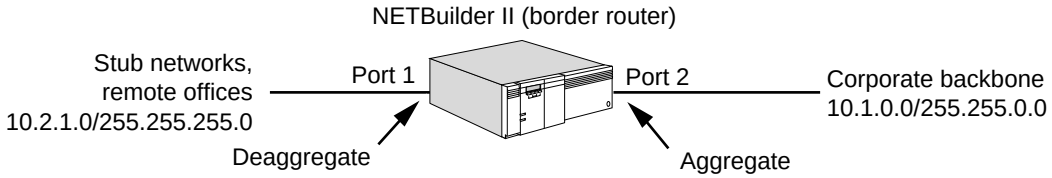


Figure 6-4 Route Aggregation/Deaggregation with RIPIP

In this configuration, the backbone has a shorter mask and no overlapping routes (10.2.0.0 255.255.0.0 and 10.2.2.0 255.255.255.0 are overlapping and cannot coexist). All subnets with the same aggregate must be fully connected and contiguous; subnets with different aggregates can be located independently. For example, all 10.2.X.X subnets must be connected and contiguous, but 10.2.X.X and 10.3.X.X can be independent.

The aggregate/deaggregate scheme provides the following benefits:

- The aggregate scheme reduces the number of routes in the backbone routing table; a smaller routing table leads to smaller routing overhead and smoother network operation.
- All the work is performed by the border router that has interfaces to subnets with different masks. It is responsible for translating routing updates and using the correct subnet mask. No other routers need to be aware that variable length subnet masks are used.
- The scheme can be used easily and quickly in a simple network topology as shown in Figure 6-4 with minimal impact; it is compatible with older routers and major upgrades are not necessary.

Using the aggregate/deaggregate scheme has the following disadvantages:

- Deaggregation may not be a beneficial scheme in some topologies (do not use it for unnumbered PPP links or topologies more complex than shown in Figure 6-4); the default route advertisement should be used as an alternative.
- If more than one router connects the longer subnets to the backbone, the RIP advertisements, after aggregation or deaggregation, may confuse each other. You may need to configure the -RIPIP ReceivePolicy parameter to filter out this type of information.

Using the Range Table Mask Scheme

To configure the range table mask scheme, use:

```
ADD -RIPIP RcvSubnetMask <IP address>-<IP address> <subnet mask>
```

For example, you could specify that all subnets between 10.2.0.0 and 10.2.255.0 use subnet mask 255.255.255.0 by entering:

```
ADD -RIPIP RcvSubnetMask 10.2.0.0-10.2.255.0 255.255.255.0
```

You can configure any type of subnet mask to any network number. The subnet mask can be longer or shorter than its natural mask. For example, the range 128.4.0.0–128.4.0.0 with subnet mask of 255.252.0.0 means that network 128.4.0.0 is assigned 255.252.0.0 as the subnet mask.

You cannot assign a subnet mask to the default route (0.0.0.0).

Procedure For each route received from a neighbor, RIPIP determines the appropriate mask using the following algorithm:

1 If the route belongs to the same IP network number of the receiving interface, use the mask of the interface.

For example, if the router receives route 10.1.0.0 on interface 10.2.0.0 255.255.0.0, the router adopts the same subnet mask for the received route.

2 If the route belongs to the same IP network number of any other interface, adopt the mask of that interface.

This action is useful when receiving routes over an unnumbered PPP link.

3 Use the natural mask based on the class (A, B, or C) of the received route.

After the mask has been determined, the software checks to see if the route falls within any of the network ranges in the range table. If there is a match, the software overrides the mask with a user-configured subnet mask. If there are multiple matches in the table, the software picks the most specific match. The entries in the range table are organized from the more specific to the less specific. For example, range 10.1.0.0–10.1.255.0 with a mask of 255.255.255.0 is more specific than 10.0.0.0–10.255.0.0 with a mask of 255.255.0.0. With these ranges in the table, network 10 is assigned subnet mask 255.255.0.0; networks that begin with 10.1.X.X are assigned a subnet mask of 255.255.255.0.

As the final step, the software compares the received route with the mask. If there are non-zero bits in the host field, the route is a host route, and the software converts the subnet mask to 255.255.255.255.

Related Information Use the range table mask scheme for more complex topologies not covered by the aggregate/deaggregate scheme (overlapping routes exist, routes learned over unnumbered PPP links). With the range table mask scheme, there is no limit on the number of potential subnet masks.

The topology in Figure 6-5 has overlapping routes. In this situation, you need to configure the range table mask on NETBuilder bridge/routers A and B. You prevent bridge/router A, for example, from adopting the shorter mask (255.0.0.0) of network C when receiving route updates from network B.

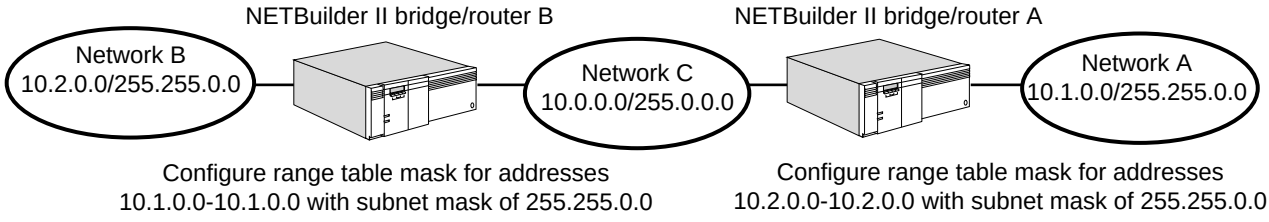


Figure 6-5 Range Table Mask

The range table mask scheme provides the following benefits:

- Classless addressing can be supported.
- No topology limitations exist.
- Overlapping routes are supported.

Using the range table mask scheme has the following disadvantages:

- Extensive configuration may be required on every router, leading to increased administrative overhead.
- When the scheme is first used, all routers must be upgraded and synchronized at the same time. Because all routers must be configured with identical information, the coexistence of non-NETBuilder bridge/routers may not be possible.

Configuring Static Routes

A static route is a user-defined route by which a network can be reached. You can configure as many static routes as desired.

Procedure

To set a static route, use:

```
ADD -IP ROUte <IP address> [<mask>] {<gateway> | !<port>} <metric>
  [Override]
```

To delete a static route, use:

```
DELete -IP ROUte <IP address> {<gateway> | !<port>}
```

Related Information

The following information pertains to static and dynamic routes.

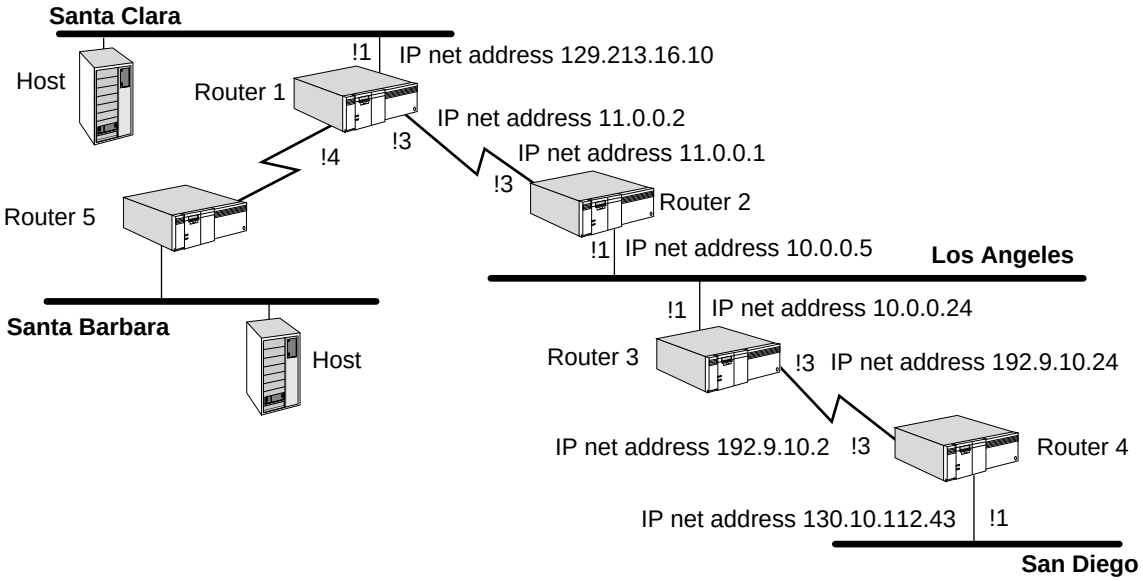


Figure 6-6 Routing Between Gateways

Refer to the example in Figure 6-6. On router 1, you can add a static route for the Los Angeles network by entering:

```
ADD -IP ROUte 10.0.0.0 11.0.0.1 1
```

This example shows that network number 10.0.0.0 (the Los Angeles network) is reachable through gateway 11.0.0.1. The gateway address is the Internet address of port 3 on router 2. Because a packet routed from router 1 to the Los Angeles network has to go through one gateway, the metric is 1.

The gateway must be located on a network directly connected to the router on which you add the static route. For example, in Figure 6-7, routers 1 and 2 both have an interface to a common network.

If the outgoing interface is a PPP link (either numbered or unnumbered), you can add a static route using the outgoing port number instead of the next-hop gateway address. For example, on router 1, you can add a static route for the Los Angeles network by entering:

```
ADD -IP ROUte 10.0.0.0 !3
```

This command achieves the same results as the command in which you entered the gateway address 11.0.0.1 as explained in the previous example.

If the PPP link is unnumbered (no IP network address is configured), you *must* provide the outgoing interface port number because the next-hop gateway address is not available.

Subnet Masks See Figure 6-7.

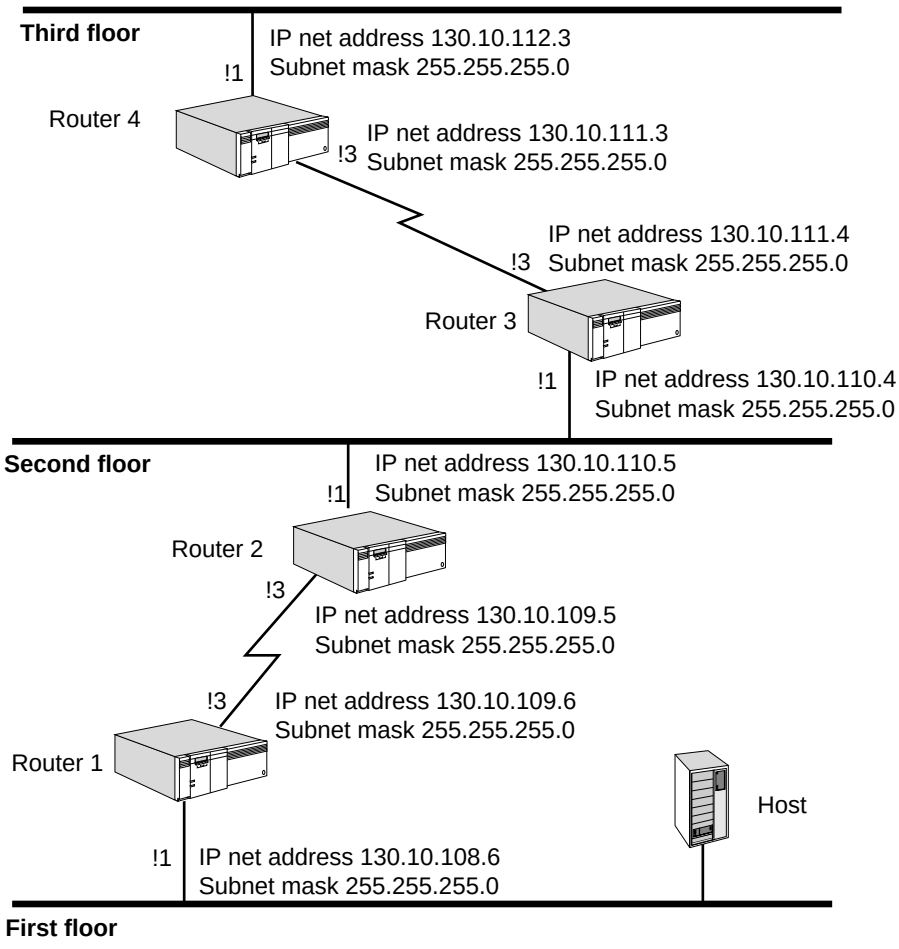


Figure 6-7 Adding a Route Statically in a Subnet Masked Environment

You can also add a route to a subnet in router 1 using a mask by entering:

```
ADD -IP ROUte 130.10.112.0 255.255.255.0 130.10.109.5 3
```

This command adds the address 130.10.112.0 with subnet mask 255.255.255.0 to the routing table. If a destination network is reachable with both a static route and a learned route, the router uses the static route unless you specify the optional Override value in the ADD ROUte command. In that case, if a learned

route of higher precedence is available, it overrides the static route. (For information on precedence, refer to "Multipath Routing" on page 6-40). The Override value is entered at the end of the command.

To add the same static route as described earlier with the Override (o) value included, enter:

```
ADD -IP ROUte 130.10.112.0 255.255.255.0 130.10.109.5 3 Override
```

Configuring Packet Filtering

The IP router supports packet filtering, which controls traffic on your IP network.

Procedure

To configure filters for your IP router, follow these steps:

1. Set up a filter policy or policies using:

```
ADD -IP FilterAddrs <adr1> [<dir>] <adr2> [<action> [<protocol>
  [<filterID>]]]<action> = {PROTocolRsrv=<tag>}|
  Discard | DODdiscard | Forward | {QPriority = H | M | L} |
  X25Profile = <profile>} <protocol> = DLSW | FTP | IP | IPDATA |
  ICMP | SMTP | TCP | TELNET | UDP
```

2. Create a filter or filters, if required, using:

```
ADD !<filterid> -IP FIlters <condition> [,<condition...]
  <condition> = <%offset>:[<operator>]<%pattern>
```

3. Set the FilterDefAction parameter using:

```
SETDefault -IP FilterDefAction = [Forward | Discard]
```

4. Enable packet filtering by entering:

```
SETDefault -IP CONTrol = Filtering
```

For complete information on the parameters used in this procedure, refer to Chapter 29 in *Reference for NETBuilder Family Software.*

Related Information

This section describes the two components of IP packet filtering. It also describes protocol reservation and the PROTocolRsrv=<tag> action option.

IP Packet Filtering Components The IP packet filtering feature is composed of two components: setting up a filter policy and creating a filter. To configure this feature, you must set up a filter policy and depending on your filtering needs, you may or may not need to create a filter. If you want to filter packets based on general criteria such as protocol or IP address, you can configure this type of filtering by setting up a filter policy only. If you want to filter packets based on more specific criteria that requires the system to examine the bytes of a packet, you need to set up a filter policy and create a filter.

If you configure a filter policy only, use the protocol field of the ADD -IP FilterAddrs command to specify a protocol you want to filter. If you configure both a filter policy and a filter, use the protocol field to specify the starting point for the offset of a condition. For complete information on the -IP FilterAddrs parameter, refer to Chapter 29 in *Reference for NETBuilder Family Software*.

Protocol Reservation One of the action options for the -IP FilterAddrs parameter is PROTocolRsrv=<tag>, which is used to set up protocol reservation. Protocol reservation assigns a percentage of bandwidth to designated packets that pass through a specified port and meet certain conditions. The conditions can be protocol type, packet length, packets destined for specified address, and so on.

Protocol reservation is set up with different procedures for different packet types. The IP filtering procedure, is applied only to IP-routed packets. IP-routed packets are also filtered using the IP firewall feature. Refer to Appendix 7 for detailed information about IP firewall.

For a detailed description of all the procedures to configure protocol reservation for the different packet types, refer to Chapter 38.

For examples of protocol reservation for designated IP-routed packets using the IP filtering procedure and the PROTocolRsrv=<tag> action option, refer to "IP Filtering Examples" example 8 on page 6-20, and example 9 on page 6-21.

The flowchart in Figure 6-8 describes how the IP packet filtering feature works. This figure assumes that IP filtering is enabled and at least one filter policy has been configured.

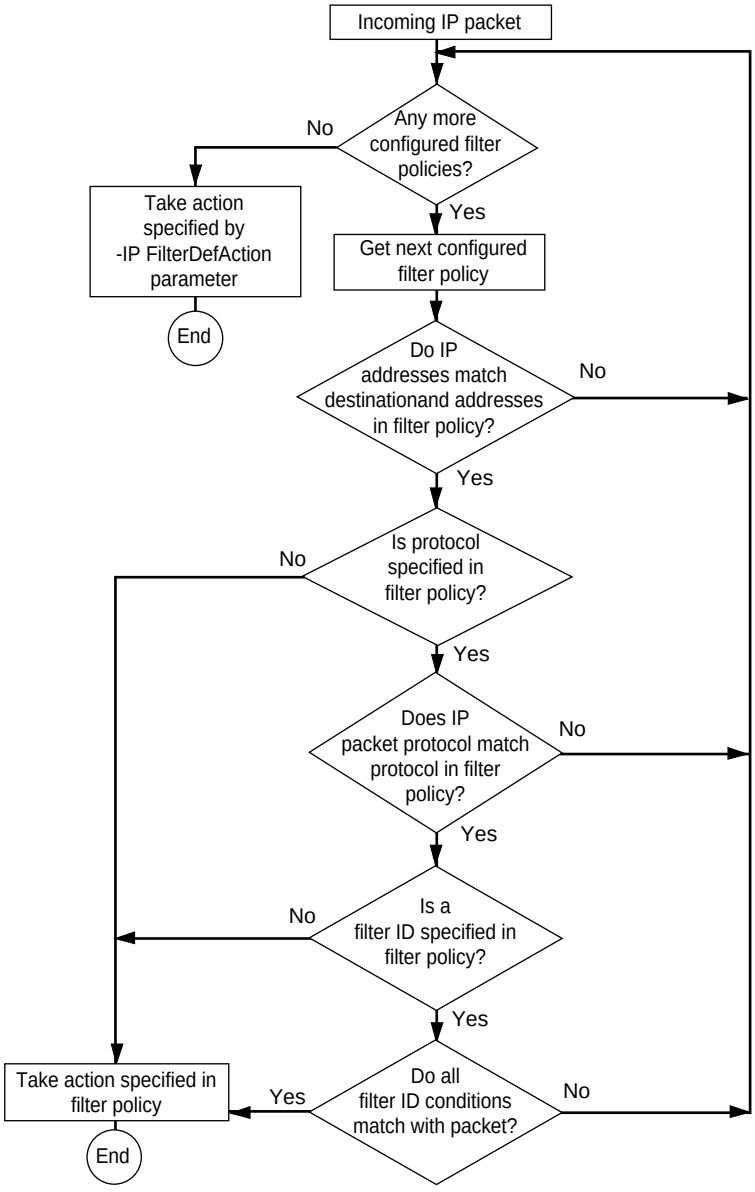


Figure 6-8 IP Packet Filtering

Enabling the packet filtering feature can have a significant impact on IP router performance. The performance is affected because the verification and decision-making process that takes place after each packet is received requires significant amounts of processing power.

IP Filtering Examples The following examples show how to configure the IP packet filtering feature.

Example 1 A router with the IP address of 129.213.16.0 and the subnet mask of 255.255.252.0 connects a local company network to the Internet. You want these router operations:

- Allow outgoing Transmission Control Protocol (TCP) connections from hosts on the local network to any host on the Internet.
- Not allow incoming TCP connections except for electronic mail (Simple Mail Transfer Protocol (SMTP), destination port %19) from a host on the Internet to a mail server (129.213.16.9) on the local network.
- Allow Internet Control Message Protocol (ICMP) messages from the Internet for feedback.
- Not allow any other packets to pass through this router.

To establish a filter that allows hosts on the local network to make TCP connections to hosts on the Internet, enter:

```
ADD -IP FilterAddrs 129.213.16.0/0.0.3.255 > ALL Forward TCP
```

To create a filter policy and filter (filter 1) that allows TCP connections from the Internet only to the local mail server provided that conditions specified in filter 1 are met, enter:

```
ADD -IP FilterAddrs ALL > 129.213.16.9 Forward TCP 1
ADD !1 -IP FIlters %2:%0019
```

The ADD -IP FilterAddrs command specified above tells the system to look for TCP packets. The system automatically adjusts for any IP options. Once the system determines that it has received a TCP packet, it looks at the packet's TCP header. The system looks at the 2-byte offset (in the destination port field) for the hexadecimal value %19 (SMTP port) as specified by the ADD -IP FIlter command above. Figure 6-9 shows the TCP header of a packet that meets the criteria established by these commands.

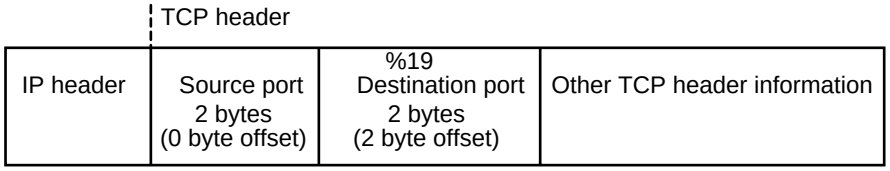


Figure 6-9 TCP Header of Packet that Meets Filtering Criteria

To establish a policy that allows ICMP messages from the Internet for feedback, enter:

```
ADD -IP FilterAddrs all > 129.213.16.0/0.0.3.255 Forward ICMP
```

To discard all other packets that do not meet any of the criteria discussed previously, enter:

```
SETDefault -IP FilterDefAction = Discard
```

Example 2 You want to set up the following conditions for a host with the address of 129.213.128.1:

- Allow incoming Telnet connections for destination port %17.
- Do not allow incoming TCP connections to the host for well-known ports (ports less than %400).
- Allow all other packets.

To set up a filter (filter 1) and corresponding policy that allows incoming Telnet connections for destination port hexadecimal %17 at a 2 byte offset, enter:

```
ADD !1 -IP FIlters %2:%0017
ADD -IP FilterAddrs ALL > 129.213.128.1 Forward TCP 1
```

To set up a filter (filter 2) and corresponding policy that does not allow incoming TCP connections for ports with a value of less than hexadecimal %400 at a 2 byte offset, enter:

```
ADD !2 -IP FIlters %2:<%400
ADD -IP FilterAddrs ALL > 129.213.128.1 Discard TCP 2
```

To forward all other packets that do not meet any of the criteria discussed previously, ensure that the -IP FilterDefAction parameter retains its default setting of Forward. If this parameter has been set to Discard, enter:

```
SETDefault -IP FilterDefAction = Forward
```

This example demonstrates that discard and forward filters can be combined. It also highlights the fact that the order filters are configured in is important. As soon as the system finds the first match, it stops searching. In this example, it is important that the system examine and forward packets with destination port %17 (filter 1) before examining and discarding packets with a destination port of less than %400 (filter 2).

Example 3 You want a router to do the following operations:

- Discard all User Datagram Protocol (UDP) packets with a source port of %161 and a destination port of %162.
- Discard all UDP packets if the tenth byte of data has the value of %60.
- Forward all other packets.

To set up a filter (filter 1) and corresponding policy that discards all UDP packets with a source port of hexadecimal %161 at a 0-byte offset and a destination port of hexadecimal %162 at a 2-byte offset, enter:

```
ADD !1 -IP FIlters %0:%161, %2:%162
ADD -IP FilterAddrs ALL> ALL Discard UDP 1
```

When creating a filter using the ADD -IP FIlters command, separating conditions with a comma (,) as shown in this example, indicates the creation of multiple conditions. If a filter has multiple conditions, all conditions must be satisfied for a match to take place.

To set up a filter (filter 2) and corresponding policy that discard all UDP packets if the tenth byte of data has the value of hexadecimal %60 at an offset of hexadecimal %a, enter:

```
ADD !2 -IP FIlters %a:%60
ADD -IP FilterAddrs ALL > ALL Discard UDP 2
```

To forward all other packets that do not meet any of the criteria discussed previously, ensure that the -IP FilterDefAction parameter retains its default setting of Forward. If this parameter has been set to Discard, enter:

```
SETDefault -IP FilterDefAction = Forward
```

Example 4 You want your router to do the following operations:

- Forward all UDP packets from a host with the address 129.213.16.9 except for RIPIP packets (destination port %208).
- Forward all other packets.

To set up a filter (filter 2) and corresponding policy that forwards all UDP packets from host 129.213.16.9 except for RIPIP packets (destination port hexadecimal %208) with a 2-byte offset, enter:

```
ADD !2 -IP FIlters %2:%0208
ADD -IP FilterAddrs 129.213.16.9 > ALL Discard UDP 2
```

To forward all other packets that do not meet any of the criteria discussed previously, ensure that the -IP FilterDefAction parameter retains its default setting of Forward. If this parameter has been set to Discard, enter:

```
SETDefault -IP FilterDefAction = Forward
```

Example 5 To assign a low priority to FTP packets going to and coming from host 129.0.0.2., enter:

```
ADD -IP FilterAddrs ALL < 129.0.0.2 QPriority Low FTP
```

Example 6 To assign 8 as the X.25 profile ID when sending IP traffic over X.25 to host 129.0.0.3, enter:

```
ADD -IP FilterAddrs ALL > 129.0.0.3 X25PROFileid=8
```

The ID of the profile created from the PROFile Service is 8.

Example 7 If the DodDiscard action in the FilterAddrs parameter is enabled, specified traffic is discarded if a dial-up path is down. If the dial-up path is up, the specified traffic is forwarded.

To mark ICMP traffic from host 10.0.0.1 to host 129.0.0.3 as DodDiscard, enter:

```
ADD -IP FilterAddrs 10.0.0.1 > 129.0.0.3 DodDiscard ICMP
```

Example 8 You want to add and set up the following filtering for your bridge/router:

- Add an IP filter that assigns 20 percent of reserved bandwidth for all Telnet sessions, and 30 percent of reserved bandwidth for all FTP packets, sent out through port 2.
- Set the IP FilterDefAction parameter so that all packets that do not meet the filtering conditions are forwarded.

To set up these filtering operations, follow these steps:

1 Add an IP filter that assigns 20 percent of reserved bandwidth to a PROTocolRsrv tag of "Telnet-tag" for all Telnet packets being sent out through port 2, and 30 percent of reserved bandwidth to a PROTocolRsrv tag of "FTP-tag" for all FTP packets being sent out through port 2, by entering:

```
ADD -IP FilterAddrs all all PROTocolRsrv = Telnet-tag Telnet
ADD -IP FilterAddrs all all PROTocolRsrv = FTP-tag FTP
```

2 Add an IP filter default action that forwards any packets that do not satisfy the filter requirements by entering:

```
SETDefault -IP FilterDefAction = Forward
```

3 Enable the IP filtering feature by entering:

```
SETDefault -IP CONTrol = Filtering
```

4 Assign 20 percent of bandwidth to the PROTocolRsrv name tag "Telnet-tag" and 30 percent of the bandwidth to the PROTocolRsrv name tag "FTP-tag" for port 2 by entering:

```
ADD !2 -PORT PROTocolRsrv Telnet-tag 20
ADD !2 -PORT PROTocolRsrv FTP-tag 30
```

5 Set PROTocolRsrv as the option for port 2 by entering:

```
SETDefault !2 -PORT QueueCONTrol = PROTocolRsrv
```

After you have made these entries, any packet sent out by the system through port 2 that has the name tag "Telnet-tag" will be allocated 20 percent of the bandwidth, and all packets with the name tag "FTP-tag" will be allocated 30 percent of the bandwidth.

Example 9 You want to add and set up the following filtering for your bridge/router:

- Add an IP filter that assigns 10 percent of reserved bandwidth for all FTP packets being sent out to the IP address 50.0.0.1 through port 3.
- Set the IP FilterDefAction parameter so that all packets that do not meet the filtering conditions are forwarded.

To set up these filtering operations, follow these steps:

1 Add an IP filter that assigns reserved bandwidth to a PROTocolRsrv tag of "FTP-tag" for all FTP packets being sent out to the IP address 50.0.0.1 by entering:

```
ADD -IP FilterAddrs all 50.0.0.1 PROTocolRsrv = FTP-tag FTP
```

2 Add an IP filter default action that forwards any packets that do not satisfy the filter requirements by entering:

```
SETDefault -IP FilterDefAction = Forward
```

3 Enable the IP filtering by entering:

```
SETDefault -IP CONTrol = Filtering
```

4 Assign 10 percent of bandwidth to the PROTocolRsrv name tag "FTP-tag" for port 3 by entering:

```
ADD !3 -PORT PROTocolRsrv FTP-tag 10
```

5 Set PROTocolRsrv as the option for port 3 by entering:

```
SETDefault !3 -PORT QueueCONTrol = PROTocolRsrv
```

After you have made these entries, any packet forwarded by the system to port 3 that has the name tag "FTP-tag" will be allocated 10 percent of the bandwidth.

Configuring RIP Routing Policies

The routing policies supported by RIP allow you to control the reporting of routing information on a per-port basis. This section describes the various routing policies you can configure and the parameters associated with configuring each policy, and provides examples of configuring policies.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Familiarize yourself with the various policies that are available, then determine which policies you want to configure. Table 6-1 lists and briefly describes each policy and its associated parameter.
- If you plan to access or receive information on routes from specific networks as opposed to all or no networks, determine the IP addresses of these specific networks.

Table 6-1 RIP Routing Policies

Policy	Description	Parameter
Advertise	Controls which routes are reported regardless of the route source.	AdvertisePolicy
Static	Controls which static routes are reported in the IP routing environment.	StaticPolicy
Exterior	Controls which BGP is reported.	ExteriorPolicy
Interior	Controls which OSPF or IISIS routes are reported.	InteriorPolicy
Receive	Controls which RIP routes are received by a trusted neighbor.	ReceivePolicy

For more information on the parameters listed in this table, refer to Chapter 47 in *Reference for NETBuilder Family Software*.

Procedure

To configure a routing policy, follow these steps:

1 Establish an advertise policy that controls the advertisement of routes through RIP regardless of the source from which the route is learned. Use:

```
ADD !<port> -RIPIP AdvertisePolicy All | None | [~]<IP address>
  [<metric> (0–15)]
```

For example, to configure a policy on port 1 that forwards information on all routes to network 10.0.0.0, enter:

```
ADD !1 -RIPIP AdvertisePolicy 10.0.0.0
```

In this example, a metric associated with network 10.0.0.0 was not specified. If you decide not to specify a metric with the AdvertisePolicy parameter or to specify a metric of zero, a route is reported with a metric calculated from the routing table.

2 Establish a receive policy that accepts or refuses to accept information on routes learned by RIP from a trusted neighbor. Use:

```
ADD !<port> -RIPIP ReceivePolicy All | None | [~]<IP address>
  [<metric> (0–15)]
```

For example, to configure port 1 so that it accepts information on routes learned by RIP for network 10.0.0.0, enter:

```
ADD !1 -RIPIP ReceivePolicy 10.0.0.0
```

In this example, a metric associated with network 10.0.0.0 was not specified. If you decide not to specify a metric with the ReceivePolicy parameter or specify a metric of zero, a route with the originally reported metric is stored in the routing table.

3 To control the reporting of routes learned from specific sources, establish the following policies:

- Exterior policy for routes learned from BGP
- Interior policy for routes learned from OSPF or IISIS
- Static policy for reporting static (user-) configured routes

Use the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters to complete this step. The syntax for the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters is the same as the syntax for the AdvertisePolicy parameter. For the AdvertisePolicy parameter syntax, see step 1.

For example, to configure a policy on port 1 that forwards routing information learned from BGP, OSPF or IISIS, and about static routes configured on network 10.0.0.0, enter:

```
ADD !1 -RIPIP ExteriorPolicy 10.0.0.0
ADD !1 -RIPIP InteriorPolicy 10.0.0.0
ADD !1 -RIPIP StaticPolicy 10.0.0.0
```

In this example, a metric associated with network 10.0.0.0 was not specified in each of the commands. If you decide not to specify a metric with the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters or specify a metric of zero, a route is reported with a metric calculated from the routing table. If you decide to have routes reported with a metric calculated from the routing table, you can manipulate the conversion formula that RIP uses to convert a metric from the routing table into one that it understands. To manipulate the formula, go on to step 4; otherwise, you have finished configuring RIP routing policies.

The metric that you configure with the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters is static or unchanging. This is in contrast to the metric that is calculated from the routing table. You can additionally manipulate the formula that is used to calculate the metric. For these reasons, 3Com recommends using the metric that is calculated from the routing table.

4 If you configured the ExteriorPolicy, InteriorPolicy, or StaticPolicy parameters and want to manipulate the formula that is used to calculate the metric, use:

```
ADD -RIPIP ImportMetric <from protocol> Multiply | Divide <operand>
```

For example, to manipulate the conversion formula used to report OSPF routes so that the metrics reported with these routes are imported into RIP without being changed, enter:

```
ADD -RIPIP ImportMetric OSPF Divide 1
```

To manipulate the conversion formula used to report OSPF routes so that the metrics reported with these routes are divided by 16, enter:

```
ADD -RIPIP ImportMetric OSPF Divide 16
```

For complete information on the commands and parameters discussed in this section, refer to Chapter 47 in *Reference for NETBuilder Family Software*.

Configuring OSPF Routing Policies

The routing policies supported by OSPF allow you to control the reporting of routes learned from other sources. This section describes the various routing policies you can configure and the parameters associated with configuring each policy, and provides examples of configuring policies.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Determine which policies you want to configure. Table 6-2 lists and briefly describes each policy and its associated parameter.
- If you plan to access or receive information on specific routes as opposed to all or no routes, determine the IP addresses of these specific routes.

Table 6-2 OSPF Routing Policies

Policy	Description	Parameter
Exterior	Controls whether exterior routing protocol (BGP) learned routes are further advertised into the OSPF domain.	ExteriorPolicy
Interior	Controls whether interior routing protocol (RIP or IISIS) learned routes are further advertised into the OSPF domain.	InteriorPolicy
Static	Controls which static routes are further advertised into the OSPF domain.	StaticPolicy
Direct	Controls whether a locally attached network (with OSPF disabled on such interface) should be further advertised into the OSPF domain.	DirectPolicy

For more information on the parameters listed in this table, refer to Chapter 41 in *Reference for NETBuilder Family Software*.

Procedure

Assume your network topology is similar to that shown in Figure 6-10. Use the following procedure to control the reporting of routes learned from other sources and advertised into the OSPF domain.

To control the reporting of routes learned from other sources and advertised into the OSPF domain, follow these steps:

1 Enable the OSPF Protocol on the appropriate ports on the backbone routers.

For example, on Routers 1, 2, and 3, enter:

```
SETDefault !1 -OSPF CONTrol = Enable
SETDefault !2 -OSPF CONTrol = Enable
```

2 Configure the backbone routers to learn routes from other interior routing protocols (such as RIPIP) within the same autonomous system.

The default setting of the -OSPF InteriorPolicy parameter is None. This means that if a router runs both the OSPF and RIPIP Protocols, the routes learned by one of these protocols are not reported to the other.

For example, to configure Router 1 to learn routes from RIPIP domain #1, on Router 1, enter:

```
ADD -OSPF InteriorPolicy All
```

You could also specify an IP address of the network in RIPIP domain #1 using:

```
ADD -OSPF InteriorPolicy <IPaddress>
```

3 Configure the backbone routers to learn routes from other exterior routing protocols, such as BGP, in another autonomous system.

For example, to configure Router 3 to learn routes from autonomous system 2, on Router 3, enter:

```
ADD -OSPF ExteriorPolicy All
```

4 In a Boundary Routing environment, configure the backbone router to advertise routes from the remote domain into the OSPF domain.

For example, on Router 2, OSPF is disabled on wide area ports 3, 4, and 5. In order for Router 2 to advertise these routes, on Router 2, enter:

```
SETDefault !3 -OSPF DirectPolicy = Advertise
```

Enter the same command for ports 4 and 5.

The DirectPolicy parameter applies to directly attached networks and only applies to ports where the -OSPF CONTrol parameter is set to Disable.

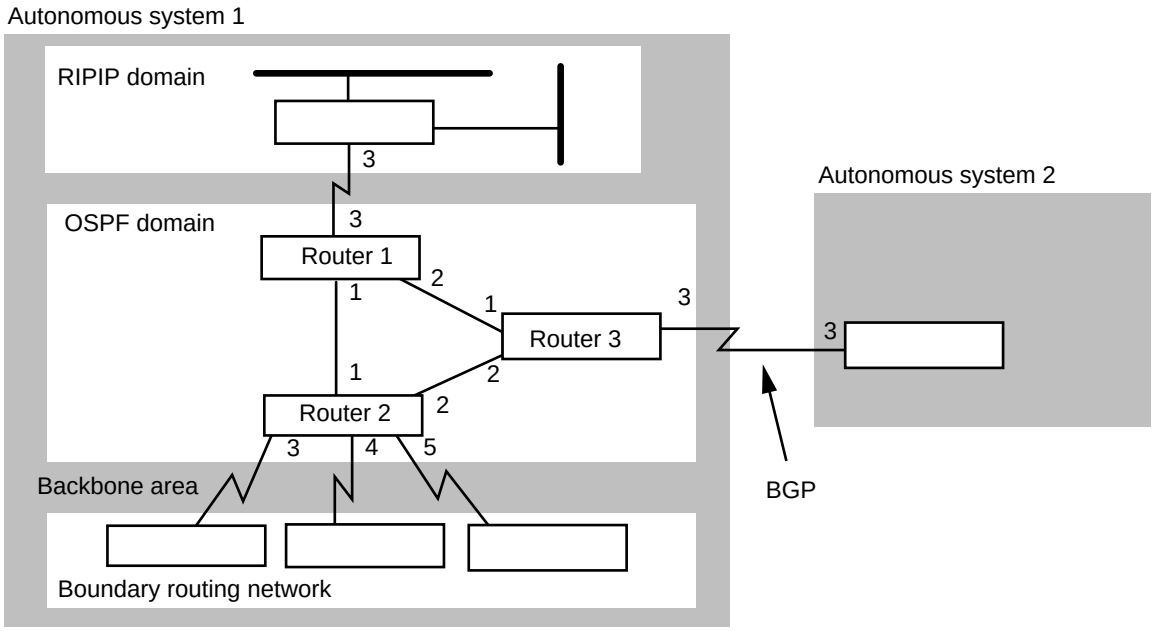


Figure 6-10 OSPF Routing Policies

Configuring IISIS Routing Policies

The routing policies supported by IISIS allow you to control the reporting of routes learned from other sources. This section describes the various routing policies you can configure, the parameters associated with configuring each policy, and examples of configuring policies.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Determine which policies you want to configure. Table 6-3 lists and briefly describes each policy and its associated parameter.
- If you plan to access or receive information on specific routes as opposed to all or no routes, determine the IP addresses of these specific routes.

Table 6-3 IISIS Routing Policies

Policy	Description	Parameter
Exterior	Controls whether exterior routing protocol (BGP) learned routes are further advertised into the IISIS domain.	ExteriorPolicy
Interior	Controls whether interior routing protocol (RIPIP or OSPF) learned routes are advertised into the IISIS domain.	InteriorPolicy
Static	Controls which static routes are advertised into the IISIS domain.	StaticPolicy

For more information on the parameters listed in this table, refer to Chapter 28 in *Reference for NETBuilder Family Software*.

Procedure

Assume your network topology is similar to that shown in Figure 6-11.

To control the reporting of routes learned from other sources and advertised into the IISIS domain, follow these steps:

1 Enable the IISIS Protocol on the backbone routers.

For example on routers 1, 2, and 3, enter:

```
SETDefault -IISIS CONTrol = Enable
```

2 Configure the backbone routers to learn routes from other interior routing protocols (such as RIPIP or OSPF) within the same autonomous system.

The default setting of the -IISIS InteriorPolicy parameter is None. This means that if a router runs both the IISIS and RIPIP Protocols, the routes learned by one of these protocols are not reported to the other.

For example, to configure router 1 to learn routes from RIPIP domain, on router 1, enter:

```
ADD -IISIS InteriorPolicy All
```

You can specify an IP address of the network in RIPIP domain using:

```
ADD -IISIS InteriorPolicy <IPaddress>
```

3 Configure the backbone routers to learn routes from other exterior routing protocols, such as BGP, in another autonomous system.

For example, to configure router 3 to learn routes from autonomous system 2, enter the following command on router 3:

```
ADD -IISIS ExteriorPolicy All
```

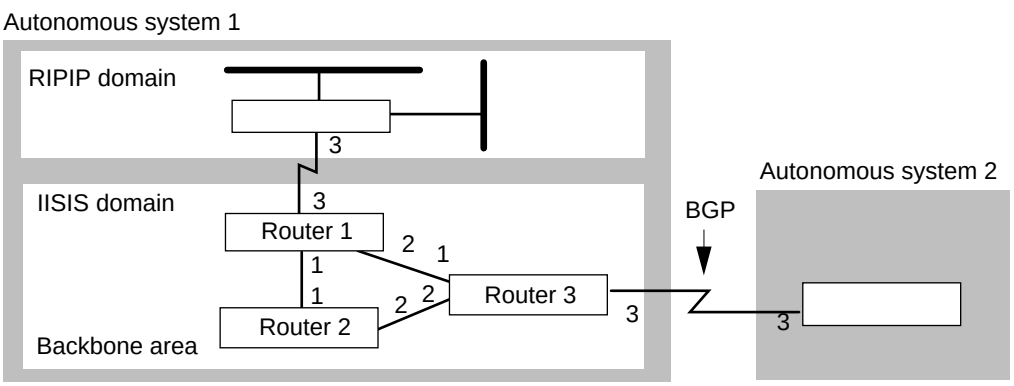


Figure 6-11 IISIS Routing Policies

Using the IP Security Option

For more information on using the IP security option, refer to Chapter 8.

Configuring Interautonomous System Routing Using BGP

This section describes how to configure BGP as your interautonomous system routing protocol. You will need to configure the following items:

- BGP peers
- Default route
- Route aggregation
- Route importing from an IGP to BGP domain (interior policy)
- Route importing from a BGP to IGP domain (exterior policy)
- Network number policies
- AS-path policies (permit, deny, and weight)

For conceptual information on interautonomous system routing, refer to "Autonomous System Routing Using BGP" on page 6-57.

Configuring BGP Peers

For BGP to learn routes between autonomous systems (ASs) and determine the reachability of networks outside of its AS, you must configure BGP peers. For information on peers, refer to "External and Internal Peers" on page 6-58.

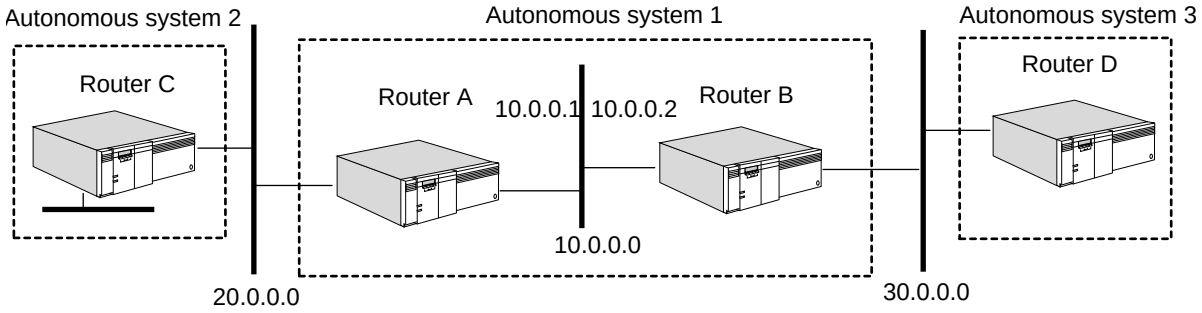


Figure 6-12 BGP Peers

To configure router A and router B in Figure 6-12 as peers, follow these steps:

1 Define the local AS number for the routers using:

```
SETDefault -BGP LocalAS = <AS Number>(1–65536)
```

In this example, routers A and B are internal peers and part of AS 1; on each router, enter:

```
SETDefault -BGP LocalAS = 1
```

This parameter defines the AS number used by this BGP speaker in the OPEN message and in all routing updates as the originating AS number. The local AS number also determines whether a peer is connected through an internal or external BGP session.

2 Add a peer to each router using:

```
ADD -BGP PEER <IP address> <AS Number>
```

AS numbers range from 1 to 65535.

On router B, specify router A's IP address (10.0.0.1) and AS Number (1):

```
ADD -BGP PEER 10.0.0.1 1
```

On router A, specify router B's IP address (10.0.0.2) and AS Number (1):

```
ADD -BGP PEER 10.0.0.2 1
```

The router must know the AS number for itself and the peer that is being added before it can establish a BGP session with its peer.

3 Enable BGP routing by entering the following command on both routers:

```
SETDefault -BGP CONTrol = Enable
```

4 Enable each peer that you added with the ADD -BGP PEER command using:

```
SETDefault [!<IP address>] -BGP PeerControl = Enable
```

On router B, specify router A's IP address for <IP address>. For example, enter:

```
SETDefault !10.0.0.1 -BGP PeerControl = Enable
```

On router A, specify router B's IP address for <IP address>. For example, enter:

```
SETDefault !10.0.0.2 -BGP PeerControl = Enable
```

Router A establishes a TCP connection with the router B (peer-to-peer communication). After the connection is established, both peers exchange BGP update packets indicating the networks each peer can reach.

5 Display routes learned through BGP by entering the following command on any BGP router:

```
SHow -BGP ROUte
```

For information on how to read the display, refer to "ROUte" on page 12-18 in *Reference for NETBuilder Family Software*.

6 Display peer information by entering the following command on any BGP router:

```
SHow -BGP PEER
```

The display shows the current mapping of peer ID to IP address to AS number and shows the current state of the peer (disabled, open, connecting).

Configuring a Default Route

You can configure a default route in the BGP Routing Table to provide the IP address of a network that can be used as the default network to destinations that are not explicitly listed in the routing table. Configuring a default route is helpful under the following circumstances:

- The routing policy of a peer does not permit the advertisement of a default route.
- When the local router is unable to maintain the complete BGP Routing Table due to memory limitations.

If a route for a particular destination address is not contained in the BGP Routing Table, BGP checks for a default route. To configure a default route, refer to Figure 6-13, and use:

```
ADD -BGP DefaultNet <IP address>
```

The configured IP address does not have to be a directly connected network. As long as the local router has a route to the IP address, it can forward all default route traffic to the IP address. The next-hop address in the BGP Routing Table is automatically calculated by the system software.

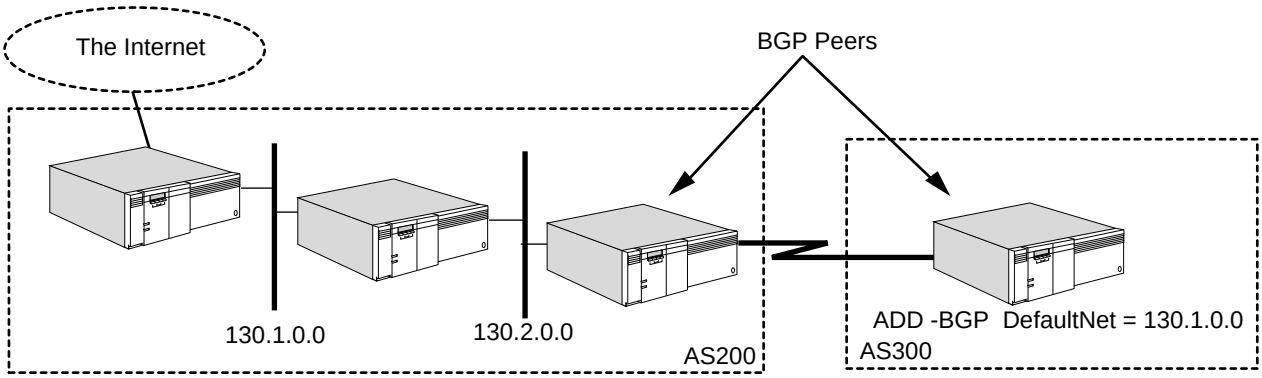


Figure 6-13 BGP Default Route

Configuring BGP Route Aggregation

BGP route aggregation uses the Classless InterDomain Routing (CIDR) address aggregation strategy to combine the characteristics of several different routes so that a single route can be advertised (see Figure 6-14). By combining several networks into one supernet, the number of BGP messages sent to peers and the size of the routing table are reduced. Unnecessary details about subnets are hidden from peers.

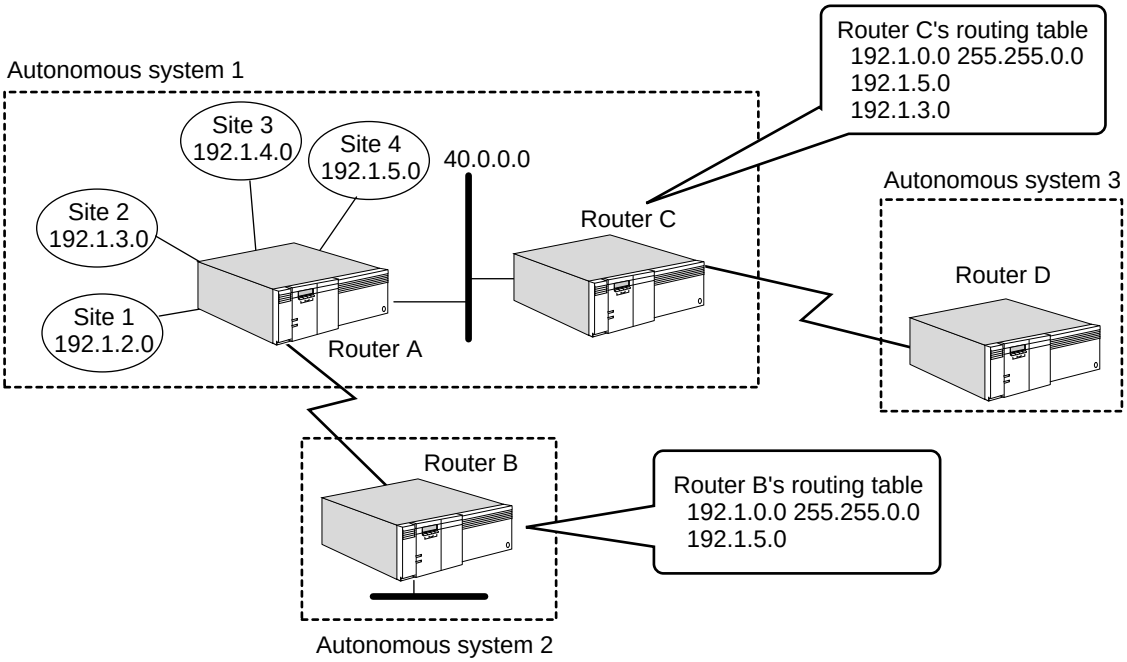


Figure 6-14 BGP Route Aggregation

Before beginning the procedure, make sure you have completed "Configuring BGP Peers" on page 6-27 (making routers A and B peers, and A and C peers), referring to Figure 6-12.

To configure BGP route aggregation, see Figure 6-14 and follow these steps:

1. Specify a list of networks that BGP advertises as a single supernet route by using:

   ```
   ADD -BGP AggregateRange <IP address> <mask>
   ```

   For example on router A, combine the routes to sites 1, 2, and 3 into a range so that only a single route is advertised. Enter:

   ADD -BGP AggregateRange 192.1.0.0 255.255.0.0

   Aggregation should never enclose Class D address space (224.0.0.0 through 239.255.255.255).

2. Specify a list of routes that BGP explicitly advertises using:

   ```
   ADD -BGP AggregateExcept <IP address> <mask>
   ```

   For example, if you do not want site 4 included in the aggregation range, enter:

   ADD -BGP AggregateExcept 192.1.5.0 255.255.255.0

3. Enable route aggregation and the BGP routing protocol by entering:

   SETDefault -BGP CONTrol = (Enable, AGgregate)

   As shown in the Figure 6-14, router A can advertise a single network (192.1.0.0/255.255.0.0) that summarizes each of the three connected sites and also explicitly advertises the exception route (192.1.5.0). Without the use of CIDR, router A advertises each route with a separate entry, and router B's routing table grows in size. With route aggregation, router B's routing table has an entry for 192.1.0.0 and 192.1.5.0.

   Explicit routes within an aggregate can be advertised by the following optional configuration. If 192.1.3.0 should be explicitly advertised to all internal peers, follow these steps:

   Add a network filter (network address and mask) using:

   ```
   ADD -BGP NetworkFilter <NetfilterID> <network address> <mask>
   ```

   For example on router A, configure route 192.1.3.0 as network filter 1 by entering:

   ADD -BGP NetworkFilter 1 192.1.3.0 255.255.255.0

4. Apply a network policy for internal peers to advertise outgoing routes using:

   ```
   ADD -BGP NetPolicyInt <NetfilterID> Explicit
   ```

   For example on router A, associate network filter 1 (192.1.3.0 255.255.255.0) to be explicitly advertised to router C by entering:

   ADD -BGP NetPolicyInt 1 Explicit

   The router C routing table has entries for 192.1.0.0, 192.1.5.0, and 192.1.3.0.

   The NetPolicyAll, NetPolicyExt, NetPolicyPeer parameters can also be configured for explicit policies that are applied to outgoing constituent routes of aggregates. For more information, refer to "Route Aggregation" on page 6-66 and Chapter 12 in *Reference for NETBuilder Family Software*.

Importing Routes from IGP to a BGP Domain

To control how route reachability information is shared between routers in different domains, the BGP router can be configured to accept or reject Interior Gateway Protocol (IGP) routing information. You can configure an interior policy on your BGP router to control the import (also known as *route leaking*) and advertisement of routes from an IGP domain. IGP refers to protocols (such as RIP, OSPF, and IISIS) that operate within a domain.

You must control how you import routes from an IGP domain to BGP domain when your network is connected to the Internet. 3Com recommends that you statically map valid routes and have only these routes imported into the BGP domain. Otherwise, dynamically changing routes in the IGP domain are constantly imported into the BGP domain causing increased load on all core routers to process unnecessary route flaps (routes coming up and going down).

To import a route from an IGP domain into a BGP domain, follow these steps:

1. Define the network filters specifying the network address and mask using:

   ```
   ADD -BGP NetworkFilter <NetfilterID> <network address> <mask>
   ```

   For example, to identify networks that have a value of 11.5.7 in the first three octets, enter:

   ```
   ADD -BGP NetworkFilter 2 11.5.7.0 255.255.255.0
   ```

   To identify networks that have a value of 193.4 in the first two octets, enter:

   ```
   ADD -BGP NetworkFilter 5 193.4.0.0 255.255.0.0
   ```

   To identify networks that have a value of 193.7.8 in the first three octets, enter:

   ```
   ADD -BGP NetworkFilter 6 193.7.8.0 255.255.255.0
   ```

2. Advertise only the specified networks and block all others, or block only the specified networks and advertise all others using:

   ```
   ADD -BGP InteriorPolicy <NetfilterID> <Permit | Deny>
   ```

   By default, BGP imports all IGP-derived, directly connected, and static routes for advertisement by BGP. To avoid this, selectively configure only those networks to be imported using the -BGP InteriorPolicy parameter. Refer to the examples that follow.

   - For example, to import only networks that have a value of 11.5.7 in the first three octets (filter 2), all networks that have a value of 193.4 in the first two octet (filter 5), and all networks that have a value of 193.7.8 in the first three octets (filter 6), enter:

     ```
     ADD -BGP InteriorPolicy 2 Permit
     ADD -BGP InteriorPolicy 5 Permit
     ADD -BGP InteriorPolicy 6 Permit
     ```

   - For example, to block the import of only the specified networks and import all others, enter:

     ```
     ADD -BGP InteriorPolicy 2 Deny
     ADD -BGP InteriorPolicy 5 Deny
     ADD -BGP InteriorPolicy 6 Deny
     ```

To avoid an invalid configuration and the interior policy from being ignored, do not configure the InteriorPolicy parameter with a mixture of permit and deny policies. You must specify the policy as either all permit or all deny policies.

3 Restart all BGP sessions to recompute the route selection process by entering:

```
SETDefault -BGP CONTrol = Enable
```

For more information, refer to "Interior Policies" on page 6-65.

Importing Routes from a BGP Domain to an IGP Domain

To control how route reachability information is shared between routers in different domains (BGP to IGP), the router can be configured to accept or reject injection of BGP routing information into the IGP routing domain. For stub and multi-homed ASs, you can configure a default route on your IGP router. For transit ASs, you can configure an exterior policy to control the import (also known as route leaking) and advertisement of routes from BGP, or just run an IGP between ASBRs. In most cases, importing routes is probably not required.

Stub Autonomous Systems A stub AS has only one connection to another AS.

In a stub AS using RIP, RIP cannot advertise both the network number and the mask. As a result, some BGP-derived routes may not be understood by RIP. You need to configure the -RIPIP DefaultMetric parameter on the border router to advertise a default route (to network 0.0.0.0) using:

```
SETDefault !<port> -RIPIP DefaultMetric = <metric> (0-15)
```

In a stub AS using OSPF or IISIS, these protocols can advertise both the network number and the mask, so you can import BGP-derived routes and advertise them by OSPF or IISIS. However, the simplest solution is to configure the OSPF or IISIS DefaultMetric parameter on the Autonomous System Boundary Router (ASBR) to generate an external link state advertisement (LSA) for network 0.0.0.0 using:

```
SETDefault -OSPF DefaultMetric = [Disable | <metric>(1-65535)
  [Type1 | Type2]]
SETDefault -IISIS DefaultMetric = Disable | <metric> (1-63)
  [Internal | External]
```

Multi-homed Autonomous Systems A multi-homed AS has connections to more than one AS but does not carry transit traffic. All of the traffic in a multi-homed AS is considered local.

In a multi-homed AS executing RIP, some BGP-derived routes may not be understood by RIP; the -RIPIP DefaultMetric parameter should be configured on *each* border router to advertise the default route. The default route allows routers that are internal to the AS to select the least-cost default route to forward traffic destined for another AS.

In a multi-homed AS executing OSPF or IISIS, all BGP-derived routes can be advertised by OSPF, but it is not always necessary or desirable to import them into a multi-homed autonomous system. The easiest solution is to configure the OSPF or IISIS DefaultMetric parameter to Type1 on *each* ASBR to generate an external LSA for network 0.0.0.0. When each internal router constructs its shortest path tree, the router selects the least cost default route.

Transit Autonomous Systems A transit AS has connections to more than one AS and carries both local traffic and transit traffic. Transit traffic is any traffic that does not originate or terminate within the local AS.

Every router within a transit AS must have explicit routing information for all networks that make up the internetwork. Each internal router must be able to forward a packet to any destination without relying on a default route. As a result, RIP cannot be used as the IGP for a transit AS. A transit AS requires a protocol that scales and supports the advertisement of BGP-derived routes. A transit AS must use BGP on all border routers, and OSPF, IIISIS, or other protocols capable of conveying network masks. An IGP is the best choice, but may not scale well.

To allow the import of BGP routes into the IGP, use:

```
ADD -OSPF ExteriorPolicy All | None | [~]<IP address> <metric>
  [Type1 | Type2]
ADD -IISIS ExteriorPolicy All | None | [~]<IP address> <metric>
  [Internal | External]
```

For more information, refer to "Exterior Policies" on page 6-65.

Configuring Network Number Policies

You can control the receipt (import) or advertisement (export) of BGP routes based on the network address using permit or deny network policies. Permit or deny policies are applied to incoming or outgoing packets, or to both incoming and outgoing packets.

To configure a network policy, follow these steps:

1 Identify the individual network number or block of network numbers to which the policy will be applied using:

```
ADD -BGP NetworkFilter <NetfilterID> <network address> <mask>
```

For example, to configure network filter 1 for network addresses starting with 192.2.1, enter:

```
ADD -BGP NetworkFilter 1 192.2.1.0 255.255.255.0
```

2 Define the policy: which peer the policy applies to, the type of policy (permit or deny) and whether the policy applies to incoming packets, outgoing packets, or both.

The following syntaxes can be used:

```
ADD -BGP NetPolicyAll <NetfilterID> {Permit | Deny [In | Out |
  Both]} | Explicit
ADD -BGP NetPolicyExt <NetfilterID> {Permit | Deny [In | Out |
  Both]} | Explicit
ADD -BGP NetPolicyInt <NetfilterID> {Permit | Deny [In | Out |
  Both]} | Explicit
ADD [!<IP address>] -BGP NetPolicyPeer <NetfilterID> {Permit |
  Deny [In | Out | Both]} | Explicit
```

For example, to permit the import of routes to 192.2.1.0 from all peers, enter:

```
ADD -BGP NetPolicyAll 1 Permit In
```

To permit the import of routes to 192.2.1.0 from external peers, enter:

```
ADD -BGP NetPolicyExt 1 Permit In
```

To permit the import of routes to 192.2.1.0 from internal peers, enter:

```
ADD -BGP NetPolicyInt 1 Permit In
```

To permit the import of routes to 192.2.1.0 to be accepted from peer 10.0.0.2, enter:

```
ADD -BGP !10.0.0.2 NetPolicyPeer 1 Permit In
```

If this is the only policy defined, all other routes from peer 10.0.0.2 are discarded.

You can configure deny policies using the same syntaxes by specifying Deny instead of Permit.

All policies in a specific direction (in/out) must be either all permit policies or all deny policies. A mix of permit and deny policies causes ambiguity resulting in the entire policy list being ignored.

3 Restart all BGP sessions to recompute the route selection process by entering:

```
SETDefault -BGP CONTrol = Enable
```

For more information, refer to "Network Number-Based Policies" on page 6-65.

Configuring AS-Path Permit or Deny Policies

You can control the receipt (import) or advertisement (export) of BGP routes based on presence or absence of specific AS numbers in the AS-PATH attribute. Recall that the AS-PATH attribute is contained in each update message. For information about this attribute, refer to "Path Attributes" on page 6-59.

Permit or deny policies are applied to incoming packets, outgoing packets, or to both inbound and outbound packets, and help filter incoming and outgoing routes.

To configure a AS-path permit or deny policy, follow these steps:

1 Define the filter to which the policy will apply using:

```
ADD -BGP AsFilter <AsfilterID> "<regular expression>"
```

For examples of regular expressions, refer to "Regular Expressions Examples" on page 6-35.

2 Define the policy: which peer the policy applies to, the type of policy (permit or deny) and whether the policy applies to incoming packets, outgoing packets, or both.

The following syntaxes can be used to assign policies to all peers, external peers, internal peers, or a specific peer:

```
ADD -BGP AsPolicyAll <AsfilterID> [[Permit | Deny [In | Out |
  Both]] [Weight <weight>]
ADD -BGP AsPolicyExt <AsfilterID> [[Permit | Deny [In | Out |
  Both]] [Weight <weight>]
ADD -BGP AsPolicyInt <AsfilterID> [[Permit | Deny [In | Out |
  Both]] [Weight <weight>]
ADD [!<IP address>] -BGP AsPolicyPeer <AsfilterID> [[Permit | Deny
  [In | Out | Both]] [Weight <weight>]
```

To maintain consistent routing information within an AS, do not apply permit or deny policies to internal BGP peers. The only type of policy that should be applied to internal peers is one that changes the preference by adding additional weight to selected paths.

3 Restart all BGP sessions to recompute the route selection process by entering:

```
SETDefault -BGP CONTrol = Enable
```

For examples of deny and permit filters, refer to "Deny Filters Examples" on page 6-35 and "Permit Filters Examples" on page 6-35.

Regular Expressions Examples This section shows examples of regular expressions.

Blank spaces are represented in the examples as underscores (_). When two spaces are shown together, a space has been inserted between the underscores, for example _ _. You must enter a blank space for each underscore shown in the examples.

Example 1 To create filter 1 that identifies an AS-PATH attribute containing AS 25, enter:

```
ADD -BGP AsFilter 1 "_25_"
```

Example 2 To create filter 2 that identifies an AS-PATH attribute containing AS 35 and AS 50 (in this order), enter:

```
ADD -BGP AsFilter 2 "_35_.*_50_"
```

Example 3 To create filter 3 that identifies an AS-PATH attribute containing AS 35 and AS 50 (in any order), enter:

```
ADD -BGP AsFilter 3 "_35_.*_50_|_50_.*_35_"
```

The horizontal bar (|) indicates a logical OR operation.

Example 4 To create filter 4 that identifies an AS-PATH attribute containing the AS Sequence <AS5, AS46, AS32>, enter:

```
ADD -BGP AsFilter 4 "<_5_ _46_ _32_>"
```

Example 5 To create filter 5 that identifies an AS-PATH attribute containing the AS Set [AS5, AS32, AS46], enter:

```
ADD -BGP AsFilter 5 "[_5_ _32_ _46_]"
```

Deny Filters Examples This section provides examples of deny filters.

Example 1 To block the import of routes containing AS 25 from all peers using filter 1, enter:

```
ADD -BGP AsPolicyAll 1 Deny In
```

Example 2 To block the advertisement of routes containing AS 35 and AS 50 to external peers using filter 2, enter:

```
ADD -BGP AsPolicyExt 2 Deny Out
```

Example 3 To block the import and advertisement of routes containing AS 35 and AS 50 to peer 10.0.0.2 using filter 2, enter:

```
ADD !10.0.0.2 -BGP AsPolicyPeer 2 Deny Both
```

Permit Filters Examples This section provides examples of permit filters.

Example 1 To permit the import of routes containing AS 25 from all peers using filter 1, enter:

```
ADD -BGP AsPolicyAll 1 Permit In
```

Example 2 To permit the advertisement of routes containing AS 35 and AS 50 to external peers using filter 2, enter:

```
ADD -BGP AsPolicyExt 2 Permit Out
```

Example 3 To permit the import and advertisement of routes containing AS 25 to peer 10.0.0.2 using filter 1, enter:

```
ADD !10.0.0.2 -BGP AsPolicyPeer 1 Permit Both
```

Configuring AS-Path Weight Policies

You can control the route selection process by assigning a specific weight to an AS, an AS path, or a subset of an AS path. The total weight for a given route is known as the *degree of preference* for the route and is calculated by summing all the individual AS-path weight expressions assigned to the route's AS-PATH attribute. If multiple routes exist for a destination, the route with the highest degree of preference is selected by the BGP route selection process. For more information, refer to "Path Selection" on page 6-63.

Weight policies are only applied to incoming routing updates and help control the route selection process based on AS numbers.

To configure a AS-path weight policy, follow these steps:

1 Define the filter to which the policy will apply using:

```
ADD -BGP AsFilter <AsfilterID> "<regular expression>"
```

For example, the following commands create filters for AS 100, 200, 300, 400, and 500:

```
ADD -BGP AsFilter 1 "_100_"
ADD -BGP AsFilter 2 "_200_"
ADD -BGP AsFilter 3 "_300_"
ADD -BGP AsFilter 4 "_400_"
ADD -BGP AsFilter 5 "_500_"
```

Blank spaces are represented in the examples as underscores (_). When two spaces are shown together, a space has been inserted between the underscores, for example _ _. You must enter a blank space for each underscore shown in the examples.

2 Define the policy (which peer the policy applies) and the weight using:

```
ADD -BGP AsPolicyAll <AsfilterID> [Weight <weight>]
ADD -BGP AsPolicyExt <AsfilterID> [Weight <weight>]
ADD -BGP AsPolicyInt <AsfilterID> [Weight <weight>]
ADD [!<IP address>] -BGP AsPolicyExt <AsfilterID> [Weight <weight>]
```

For examples of weight filters, refer to "Weight Filters Examples" on page 6-37.

In addition, the following syntaxes affect weight-based policies and the degree of preference in the route selection process:

```
SETDefault -BGP DefaultWeight = <number>(-2000 to 2000)
SETDefault [!<IP address>] -BGP PeerWeight = <weight>(-2000 to 2000)
```

The DefaultWeight parameter configures a default weight that is added to each route when computing the degree of preference (LOCAL-PREF attribute) for the route. You can configure this parameter to give priority in the route selection process to routes received by one BGP speaker over routes received by other BGP speakers.The default value of this parameter is 0. For information about the LOCAL-PREF attribute, refer to "LOCAL-PREF" on page 6-62.

The PeerWeight parameter configures a weight that is added to all routes received from the specified peer. The default value of this parameter is 0.

For more information, refer to "Degree of Preference Calculations" on page 6-37.

3 Restart all BGP sessions to recompute the route selection process by entering:

```
SETDefault -BGP CONTrol = Enable
```

Weight Filters Examples This section provides examples of weight filters.

Example 1 To assign a weight of 10 to routes with AS 100 in the AS-PATH (filter 1) received from all peers, enter:

```
ADD -BGP AsPolicyAll 1 Weight 10
```

You can assign a weight of 10 to routes with AS 100 in the AS-PATH (filter 1) received from external peers or internal peers by using the AsPolicyExt and AsPolicyInt parameters, respectively.

Example 2 To assign a weight of 10 to routes with AS 100 in the AS-PATH (filter 1) received from peer 10.0.0.2, enter:

```
ADD !10.0.0.2 -BGP AsPolicyPeer 1 Weight 10
```

Degree of Preference Calculations The degree of preference, which must always be greater than or equal to 0, is calculated before the route selection process using the following formula:

Degree of preference = (Total ASPolicy Weight) + PeerWeight + DefaultWeight

In the following examples, assume that AS 100 has a weight of 10, AS 200 has a weight of 20, AS 300 has a weight of 30, and AS 500 has a weight of 50.

Example 1 The local router receives a route with the following AS-PATH attribute: {_500_ _200_ _600_ _100_ _300_}. Assume that both the PeerWeight and DefaultWeight parameters have a value of 0. The degree of preference is equal to:

50 + 20 + 0 + 10 + 30 + 0 (PeerWeight) + 0 (DefaultWeight) = 110

If an AS in the AS-PATH attribute has not been assigned a weight using the AsPolicyXXX parameter, it is assumed to have a weight of zero (0).

Example 2 Assume that peer 10.0.0.2 has been assigned a PeerWeight of 100 and the local router's DefaultWeight value is 0. The local router receives a route from peer 10.0.0.2 with the following AS-PATH attribute: {_500_ _200_ _100_}. The degree of preference is equal to:

50 + 20 + 10 + 100 (PeerWeight) + 0 (DefaultWeight) = 180

Example 3 Assume that peer 10.0.0.2 has been assigned a PeerWeight of 100 and the local router has been configured with a DefaultWeight value of -50. The local router receives a route from peer 10.0.0.2 with the following AS-PATH attribute: {_500_ _200_ _100_}. The degree of preference is equal to:

50 + 20 +10 + 100 (PeerWeight) - 50 (DefaultWeight) = 130

For more information, refer to "AS-Path-Based Policies" on page 6-66.

How the IP Router Works

This section describes the following concepts involved in IP routing activities:

- Understanding IP network topology
- Multipath routing
- Default routes
- Learning routes within an autonomous system
- Configuring IISIS for dual IP and OSI mode
- Learning routes between autonomous systems using BGP
- Address resolution
- Other global routing configurations

Understanding IP Network Topology

An IP network is configured on each interface where IP packets are received and sent. The interface can be either a local LAN interface or a serial line interface for a wide area network. Figure 6-15 shows a wide area router (Router 1) connecting two local Ethernet networks (Santa Clara buildings 1 and 2) to two wide area networks (Los Angeles and Santa Barbara). The Los Angeles network is connected by a point-to-point line, and the Santa Barbara network is connected by an X.25 link.

Although Figure 6-15 shows that the wide area ports that connect the Santa Clara network to the Los Angeles network are assigned IP addresses, PPP does not require that you assign an IP address to each wide area port. If you do not want to assign an IP address to a wide area port, you must set the SETDefault -IP NETaddr command to UnNumbered. For more information on this topic, refer to "Configuring for Local Area Networks and Point-to-Point Links" on page 6-1.

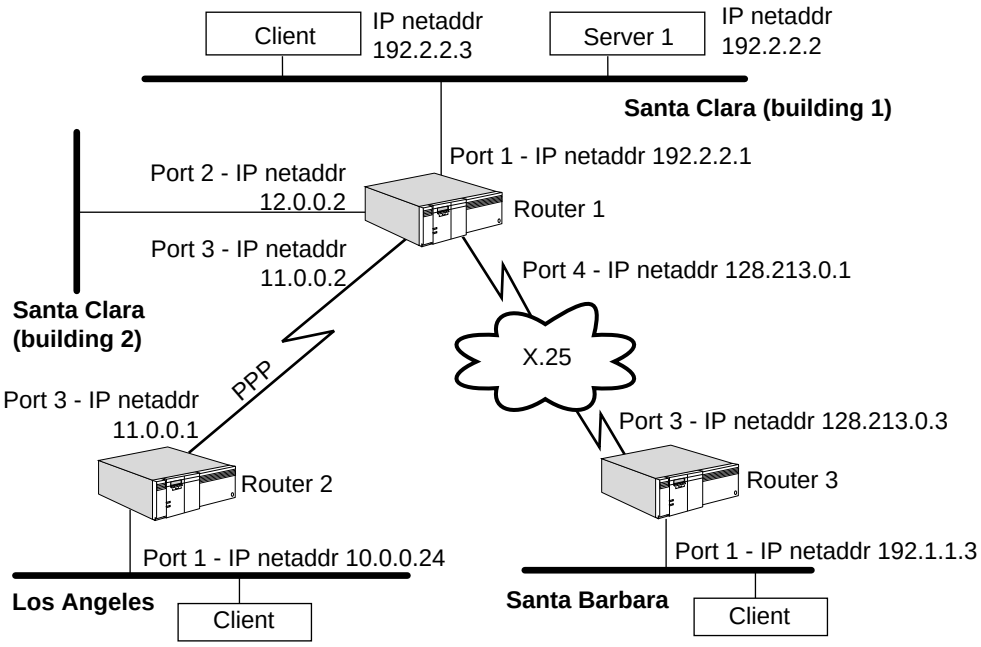


Figure 6-15 Wide Area Router Connecting Four IP Networks

CAUTION: *Each IP address that you assign directly to a port must be unique, that is, you cannot assign the same IP address to different ports. If you want to give several ports the same IP address, define a port group containing the ports,*

and assign the IP address to the group. For information about defining port groups, refer to "Configuring Multiple Logical Networks" on page 1-22, "Configuring Multiple Logical Networks" on page 1-22, and "Configuring Logical Networks over IP" on page 6-9.

A local network is referred to as an *attached network*. When two wide area routers are connected by one or more serial lines, their serial interfaces should be on the same network. For example, in Figure 6-15, port 3 of Router 1 and port 3 of Router 2 are on the same network.

As an IP network grows, contiguous routers can be grouped into areas if using OSPF. Two areas can be interconnected through a backbone area. Areas and backbone areas can be grouped into autonomous systems. An autonomous system consists of routers and networks administered by a single authority. An autonomous system typically runs a single intra-autonomous system routing protocol, such as OSPF.

RIP does not support areas.

Routing can take place between autonomous systems using an interautonomous system protocol, such as the Border Gateway Protocol (BGP). Figure 6-16 shows two areas within an autonomous system being connected by a backbone area. It also shows an autonomous system connected to two other autonomous systems.

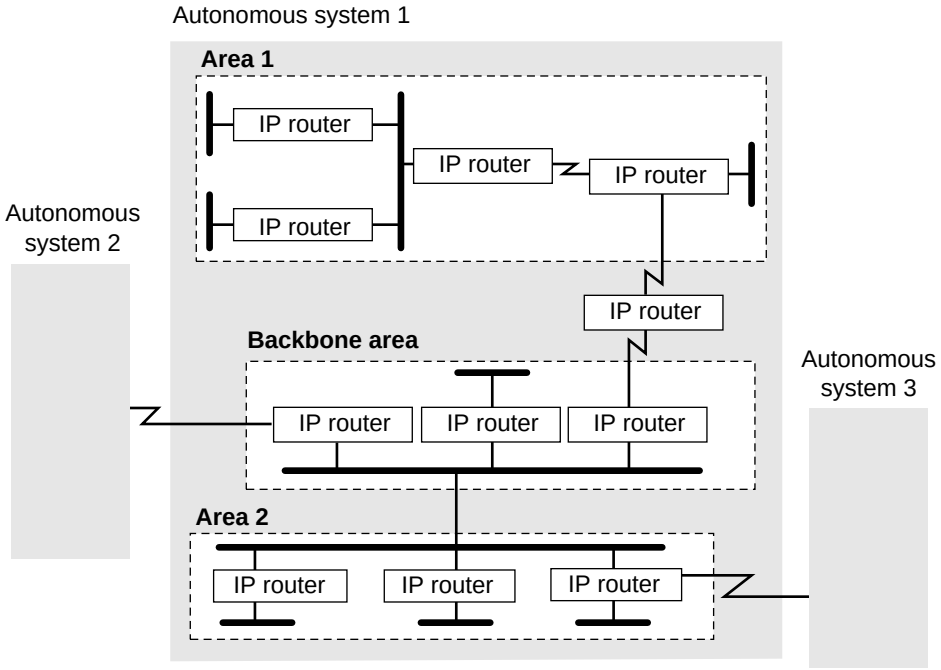


Figure 6-16 Typical IP Network Running OSPF

A router must check its routing table to determine where to route a packet. If the destination is on an attached network, the router can send it directly to the network. But if the destination is farther away in the internetwork, the router must route the packet to another router (called a *gateway*) that is closer to the destination. The route to a wide area network can be statically configured or dynamically learned through RIP, OSPF, IISIS, and BGP. When two routers are located on the same network (that is, each of them has at least one interface to the network), they are considered *neighbors* or *neighboring gateways*.

Multipath Routing

The router supports multipath routing, which means that up to four routes for each destination address can be stored in the routing table. Advantages of multipath routing are as follows:

- The router can still route a packet using an alternative route if the primary one fails; it is more responsive to network topology changes than if only one route to a destination exists.
- The router can distribute the load among the available equal-cost best paths.

Routes learned by different routing protocols are assigned a different precedence in the routing table. Some routing protocols such as OSPF and IISIS have multiple route classes. The classes are also assigned a different precedence.

When multiple routes for a destination exist, the router uses the route with the highest precedence. The types of routes used (listed by decreasing precedence) are listed in Table 6-4.

Table 6-4 Route Precedence

Precedence Level	All Protocols without BGP	All Protocols with BGP
1	Static (added without Override)	Static (added without Override)
2	OSPF Intra Area	OSPF Intra Area
3	OSPF Inter Area	OSPF Inter Area
4	IISIS Intra Area	IISIS Intra Area
5	IISIS Inter Area	IISIS Inter Area
6	RIP	RIP
7	OSPF Type 1 External	OSPF Type 1 External
8	OSPF Type 2 External	OSPF Type 2 External
9	IISIS External	IISIS External
10	ICMP redirect (not applicable for a router; only for host mode)	ICMP redirect (not applicable for a router; only for host mode)
11	Static (with override)	Static (with override)
12a		BGP
12b		Route configured with DefaultNets
13		Non-BGP default route

When the IP Protocol routes a packet and BGP is enabled, the software looks up the route as follows:

1. Looks for a route learned by any protocol except BGP in the All Protocols Routing Table.
   - The software searches in the order specified in the "All Protocols without BGP" column in Table 6-4.
   - BGP has its own routing table separate from all the other protocols.
   - The software does not consider the default route yet.
2. If a route to the destination is not found, the software looks for the route in the BGP Routing Table (precedence level 13a in Table 6-4).
3. If a route to the destination is not found in the BGP Routing Table, the software looks for any configured default networks configured with the ADD -BGP DefaultNet <IPaddress> syntax.

4 If no default networks are configured, the software again searches the All Protocols Routing Table, looking for default routes.

If BGP is disabled, the software follows steps 1 and 4.

The routing table displays routes with a high precedence first.

OSPF Type 1 and 2 external metrics allow you to define how you want Autonomous System Boundary Routers (ASBRs) to report metrics.

A Type 1 external metric is the sum of the metric learned within the autonomous system by OSPF plus the metric learned outside of an autonomous system by BGP. A Type 2 external metric is the metric learned outside of an autonomous system by BGP only.

In an OSPF environment, you can set your ASBR to report Type 1 or 2 external metrics using the InteriorPolicy, ExteriorPolicy, Static Policy, and DefaultMetric parameters. For complete information on these parameters, refer to Chapter 28, Chapter 41, and Chapter 47 in *Reference for NETBuilder Family Software*.

If the route with the highest precedence fails, the route with the next highest precedence will be used. A route in the routing table is deleted in these situations:

- OSPF and IISIS both compute routes based on link state information from all routers. The entire routing table is recomputed each time the topology changes.
- A dynamic route learned through RIP is deleted when a router times out and goes through the HOLD-DOWN and GARBAGE COLLECTION states. A router times out when it fails to hear from a neighbor for a period that is six times the value of the UpdateTime parameter. For example, if the value of the UpdateTime parameter is 45 seconds, the router will time out if it does not hear from its neighbor for 270 seconds.
- A DELete ROUte command removes a static route.
- A lowest precedence route is deleted when four routes of higher precedence are available. This situation occurs when a fifth route is learned and has a higher precedence than the lowest precedence route.

Dynamic routes learned by RIP can be removed by using the FLush -IP AllRoutes command. However, this command does not flush routes learned by OSPF, IISIS, or BGP from the routing table.

Route Selection and Load Splitting

If two or more routes with the same route source precedence are available to reach a destination, the router always selects the route with the lowest metric (measured in hops for RIP and in administrative cost for OSPF, or IISIS). If there is more than one route learned by the same routing protocol with the same equal-cost, low metric, you can split the load between these routes on a round-robin basis. The -IP CONTrol parameter (SplitLoad | NoSplitLoad) determines whether load splitting is performed.

Because load splitting balances the load among different routes, 3Com recommends it if two or more routes are available to reach a destination and the routes have similar metrics. However, if the routes connecting various

networks have different metrics (that is, there is only one route with the fewest hops or lowest cost to a destination), load splitting is not necessary.

Route Selection Examples

Table 6-5 is an example of a routing table and shows how a route is chosen.

Routes are selected on the basis of precedence, lowest metric, or in cases where multiple routes have the same precedence and metric, through load splitting or the first route discovered. The examples in Table 6-5 demonstrate these criteria and can be applied to all types of routes.

Table 6-5 Routing Table Containing Multiple Paths

Network	Gateway	Metric*	Route Source
10.0.0.0	129.213.1.1	100	OSPF—Intra
	129.213.1.2	1	RIP
20.0.0.0	129.213.16.1	1	RIP
	129.214.1.1	2	RIP
30.0.0.0	129.213.16.1	100	OSPF—Intra
	129.213.16.2	100	OSPF—Intra
	129.213.16.3	100	OSPF—Intra

* RIP uses hop count as its metric. The OSPF metric is computed from total administrative cost between router and destination.

Example 1 For network 10.0.0.0, there are two routes available, but these routes are not comparable. The first route is learned by OSPF, and the second route is learned by RIP. Because routes learned by OSPF take precedence over routes learned by RIP, gateway 129.213.1.1 is selected.

Example 2 For network 20.0.0.0, there are two routes available through RIP. Because gateway 129.213.16.1 requires one hop and gateway 129.214.1.1 requires two, the router always selects gateway 129.213.16.1 because it requires the fewest hops or the lowest metric to reach its destination.

Example 3 The routing table entry for network 30.0.0.0 has three available routes to reach it. All are dynamic routes learned through OSPF, and all require an administrative cost of 100. The router chooses the route as described here:

With load splitting — The route is chosen on a round-robin basis. Gateway 129.213.16.1 is used first, then 129.213.16.2, then 129.213.16.3. If one of these routes becomes invalid, it is no longer considered in the selection procedure.

Without load splitting — The route recorded earliest is always used. In this case, the gateway 129.213.16.1 is used.

Default Routes When a router needs to route a packet destined for an address for which there are no entries in the routing table, it uses the default route if one exists. The network 0.0.0.0 represents the default route. The router supports up to four default routes; when more than one default route is available, the same selection rules apply. If load splitting is enabled, the load is distributed among equal-cost best paths. For additional information, refer to "Multipath Routing" on page 6-40.

An advantage of a router using a default route is that network overhead in an autonomous system can be reduced. The reduction in overhead occurs because the router does not need to advertise all external routes.

The following example will help you understand default routes.

Example Router A receives a RIP update packet from router B, which has an entry indicating that network 0.0.0.0 is reachable with metric 3. Router A considers router B its default gateway. That is, if router A needs to route a packet whose destination is not found in its routing table, it sends the packet to router B.

The interior routing protocols for IP (RIP, OSPF, and IISIS) can be configured to advertise a default route by assigning a non-zero value to the DefaultMetric parameter of the routing protocol's service. You do not need to configure the DefaultMetric parameter on every router throughout the domain. The default route learned on one interface is propagated to neighbors on the other interfaces (unless inhibited by the NetworkPolicy parameter).

Each interior routing protocol propagates the advertisement of the default route as the normal operation. For the RIP Protocol, it is possible to suppress propagation of the default route by using the AdvertisePolicy parameter. Since OSPF and IISIS are both link state routing protocols, they cannot suppress any routing information within the bounds of their routing system. However, they can control information that they import from other routing systems.

If more than one interior routing protocol is in operation on a network, the routes from one system can be introduced into the other system by using the InteriorPolicy parameter of the protocol that is importing the routes.

Suppose that RIP and IISIS are both in operation, and that RIP needs to import routes from the IISIS system. Use the InteriorPolicy parameter in the RIPIP Service to achieve this routing. For more information on this parameter, refer to Chapter 47 in *Reference for NETBuilder Family Software*.

When a router is operating both an interior protocol (RIP, OSPF, or IISIS) and an exterior protocol (BGP), then the ExteriorPolicy parameter in the service of the interior routing protocol is used to control the import of routes from the exterior routing protocol into the interior routing protocol. The InteriorPolicy parameter in the service of the exterior routing protocol is used to control the import of routes from the interior routing protocol into the exterior routing protocol.

You can configure the default route in one of two ways:

- On the exit router of a domain, configure a static override default route with a metric of 1 that points to the first hop outside the domain. Then use the StaticPolicy parameter of the selected routing protocol to import this route into the routing protocol and advertise it into the domain.
- At the top level, set the DefaultMetric parameter in the selected routing protocol to instruct the router to originate a default route.

The second method may also be useful at a router that interconnects a RIP domain with an OSPF domain, instead of importing all routing information from each domain into the other.

If the bridge/router is instructed to originate the default route (by setting the DefaultMetric parameter to a nonzero value), it does not accept another router's advertisement of the default route. Consider two routers in parallel, both

originating a default route and each accepting the other's advertisement of the default route. Any packet received by one router is forwarded to the other router, and back again, until the time-to-live timer is exhausted and the packet is dropped.

Learning Routes within an Autonomous System

The router fully supports RIP according to RFC 1058. It also supports OSPF Version 2 according to RFC 1583.

If you are planning to use both RIP and OSPF when expanding your IP network, 3Com recommends that autonomous systems using OSPF make up the core of the network and that autonomous systems using RIP surround those using OSPF. Figure 6-17 is an example of this topology.

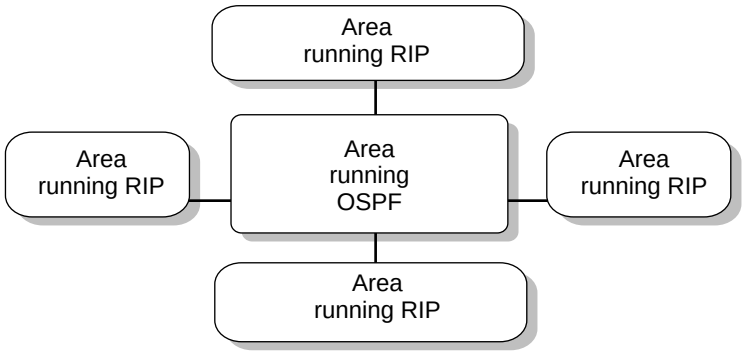


Figure 6-17 Recommended Autonomous System Topology

If a router runs both OSPF and RIP Protocols, the routes learned by one of these protocols are not reported by the other according to the default settings of the InteriorPolicy parameter in both the RIPIP and OSPF Services. If you want cross-reporting between these protocols, set the InteriorPolicy parameter in the RIPIP, IISIS, and OSPF Services accordingly.

If you are using OSPF in a topology with end stations, you need to configure a default gateway on the end stations. Many end stations learn RIP routes dynamically, but they usually do not learn OSPF routes dynamically.

Learning Routes with RIP

Normally, every 30 seconds or every time it learns a route change for a network, the router uses broadcast packets to report to its neighboring gateways the following types of information:

- The networks it can reach
- The metric associated with each network it can reach

By default, the information in update packets pertains only to learned routes. Static route information is not reported.

You can configure some router parameters (refer to "User Configurations" on page 6-47) to determine how the router sends out the updates and what is included in them. For example, you can configure the parameters for the following purposes:

- To change the frequency of the broadcast traffic (UpdateTime parameter)
- To prevent the router from sending or receiving update and request packets (CONTrol parameter)
- To control the set of neighboring routers from which the router receives updates and to which it sends them (AdvToNeighbor and RcvFromNeighbor parameters)
- To prevent the router from sending out a trigger update response upon a route change for a network (CONTrol parameter)
- To enable the router to report static routes (StaticPolicy parameter)
- To enable the router to report routes learned with other interior routing protocols, such as IISIS and OSPF (InteriorPolicy parameter)
- To cause some routes not to be reported or to be reported with the infinity metric, that is, using poison reverse (CONTrol parameter)

Network Reachability

The following types of networks are considered *reachable* when a router broadcasts its RIP update packets:

- All directly connected networks, unless the network is shared by its neighbor and itself
- All static routes (as controlled by the StaticPolicy parameter)
- All dynamic routes learned through RIP and either OSPF or IISIS in the routing table (as controlled by the InteriorPolicy parameter)
- All dynamic routes learned through BGP (as controlled by the ExteriorPolicy parameter)

Solving the Slow Convergence Problem with Split Horizon

Ideally, all routers learn of new routes and discard obsolete routes immediately. That is, the contents of their respective routing tables converge rapidly so that all routing tables always contain correct information. An undesirable side effect of RIP is the possibility that the time is prolonged during which the unreachable network is considered reachable. One solution to this problem of slow convergence is called *split horizon*.

In a WAN environment, the 3Com implementation of next-hop split horizon (-RIPIP CONTrol = NonMesh) eliminates the need for a fully meshed network when using RIP. In next-hop split horizon, the router learning of a network records the IP address of the neighbor from which the network was learned instead of recording the port through which the network was learned. When the router advertises its own reachable networks, it advertises to all neighbors except the one from which it learned of the network being advertised.

Nonmeshed WAN Networks Figure 6-18 shows a nonmeshed Frame Relay, X.25, or ATM network using RIP on which router R is the root router and routers A, B, and C are remote routers that are configured as neighbors on router R. Router R sends RIP updates individually to its neighbors, remote

routers A, B, and C. When sending RIP packets, router R advertises to neighbors all networks it knows about (in this example, networks 20.0.0.0, 30.0.0.0, 40.0.0.0, and 50.0.0.0) if next-hop split horizon is not used. Network 10.0.0.0, being common to all routers in the diagram, is automatically excluded from RIP updates between these routers.

By applying next-hop split horizon, router R does not advertise network 30.0.0.0 to router A, because it learned of 30.0.0.0 from router A. Router R also does not advertise network 40.0.0.0 to router B, nor does it advertise 50.0.0.0 to router C, because it learned of those networks from those routers.

You need to enable the next-hop split horizon feature by setting -RIPIP CONTrol to NonMesh.

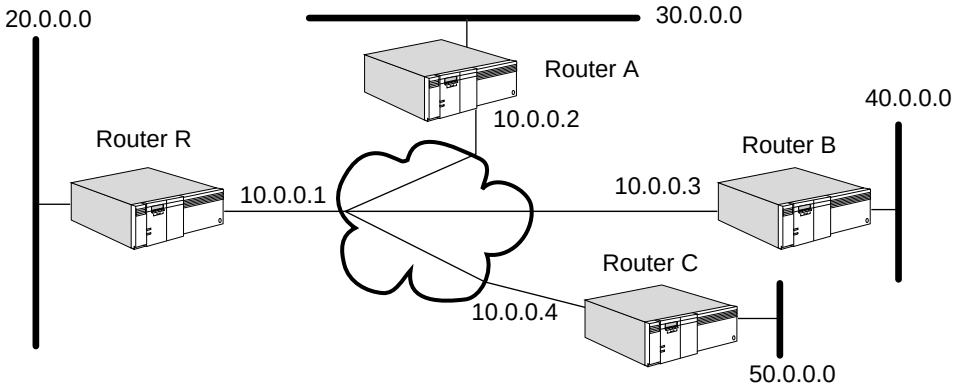


Figure 6-18 Route Advertisement over Nonmeshed Frame Relay or X.25 Network

Meshed WAN Networks In Figure 6-19, the WAN network is meshed because all routers are directly connected to one another. Even if a WAN network is meshed, you must configure routers A, B, and C as neighbors on router R, the root router, for RIP to unicast updates over the WAN. You also need to set -RIPIP CONTrol to FullMesh so that next-hop split horizon is disabled. This example applies to Frame Relay, ATM, and X.25 networks. With Frame Relay networks, RIP neighbors can be dynamically learned.

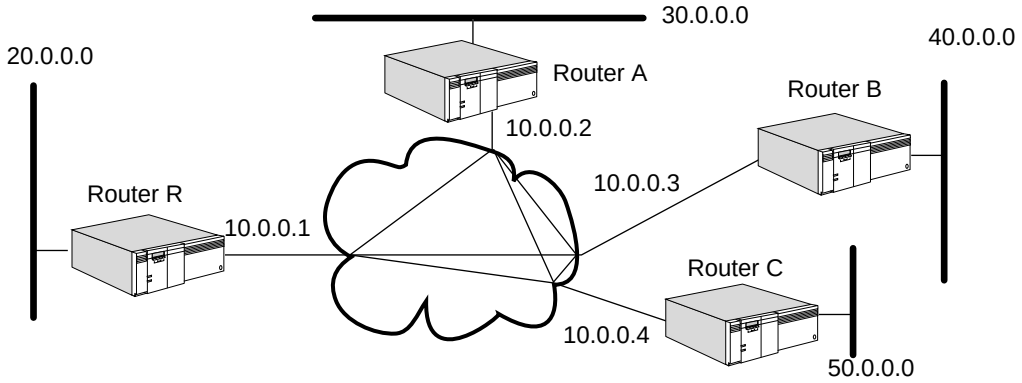


Figure 6-19 Route Advertisement over Meshed Frame Relay Network

LAN Networks On a LAN network, it is not necessary to configure neighbors. If you do not configure neighbors, RIP broadcasts the updates over the LAN. If you configure neighbors, RIP unicasts the updates.

Solving the Slow Convergence Problem with Poison Reverse

Poison reverse or no poison reverse is configurable using the Poison or NoPoison value for the -RIPIP CONTrol parameter.

If poison reverse is enabled, the router advertises all routes to all neighbors, but when advertising a route to a neighbor that has advertised the same route, the router sets the metric to infinity (0xFFFF) to prevent the recipient from adding the route to its routing table. Poison reverse speeds convergence but adds to network overhead.

If poison reverse is disabled, the router omits routes learned from one neighbor from RIP updates sent to that neighbor. No poison reverse has the advantage of minimizing network overhead in large network configurations at the expense of slower convergence.

User Configurations

Table 6-6 shows how you can change the way the router broadcasts or processes RIP update packets. This table includes only the parameters that were not discussed in previous sections. For complete information on the parameters listed in this table, refer to Chapter 47 in *Reference for NETBuilder Family Software*.

Table 6-6 Configuring the IP Router for RIP Updates Using RIPIP Parameters

Parameter	Result
UpdateTime*	Changes the frequency of the update packets.
CONTrol parameter option:	
TRigger | NoTRigger	Determines whether a route change for a network triggers an update packet from the router.
AdvToNeighbor	Determines to which gateways on the directly connected networks the router sends the update packets.
RcvFromNeighbor	Determines to which gateways on the directly connected networks the router should listen for routing information.

* The parameter applies to the entire router. All routers exchanging RIP information should have the same value for this parameter. Otherwise, routing loops or loss of connectivity in the network may occur.

Different States of RIP-Learned Routes

To avoid routing loops, new information about a route is ignored for a designated period before it is used. Figure 6-20 summarizes how a route learned through RIP changes states. Explanations of the different states follow the figure.

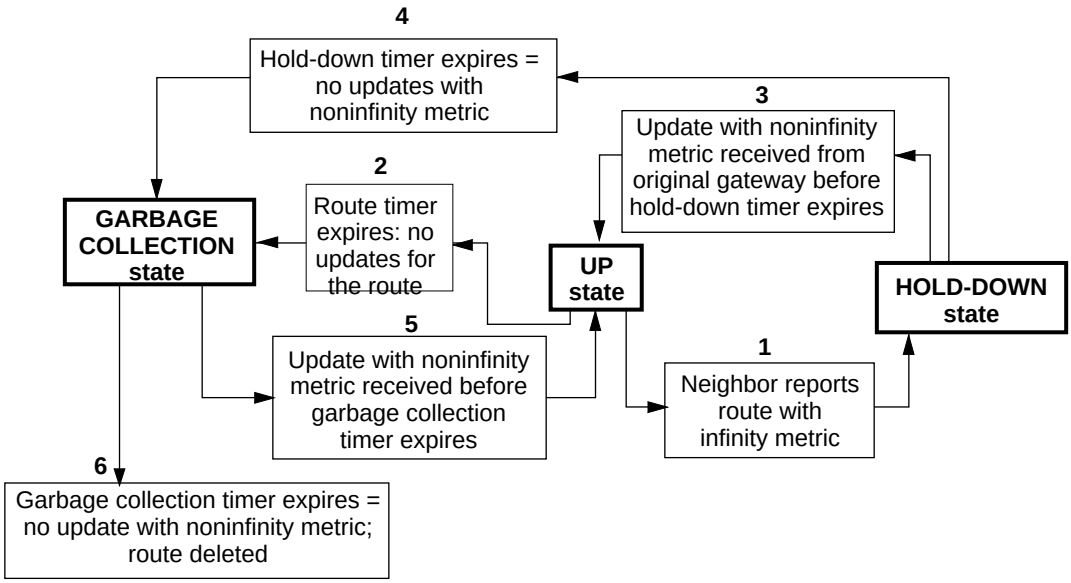


Figure 6-20 Different States of a RIP Route

- GARBAGE COLLECTION state

  When the timer for a route that has been in the HOLD-DOWN state expires, that route changes to GARBAGE COLLECTION state. This happens when no update packets are received to indicate that the route is still reachable. In this state, if a neighboring gateway reports the route with a noninfinity metric within 120 seconds, the route can go back to the UP state. If no updates are received within 120 seconds (garbage-collection timer), the route is deleted from the routing table. It is possible to go into GARBAGE COLLECTION state if no updates are received within 180 seconds.

- UP state

  A route is considered UP if it is reachable with a noninfinity metric (15 or fewer hops). Whether it is reachable is determined by the last update received from the neighboring gateways. It remains UP for 180 seconds (the route timer). The timer is reset each time a new update for the route is received.

- HOLD-DOWN state

  A route in UP state changes to HOLD-DOWN state if an update received from the original gateway indicates that the route is associated with an infinity metric (16 hops). In this state, all update information received from other gateways for that route is ignored.

  However, if an update is received from the original gateway within 60 seconds (the hold-down timer), and it associates a noninfinity metric with the route, the route goes back to UP state.

  If the hold-down timer expires, the route goes from HOLD-DOWN state to GARBAGE COLLECTION state for 120 seconds.

When you display the routing table with the SHow -IP AllRoutes command, the state of each route is displayed under the STATUS heading.

Learning Routes with OSPF

Normally, every 30 minutes or every time the router learns a route change for a network, it uses multicast packets to report to its neighbors the following types of information:

- The networks and the directly connected routers
- The metric associated with each directly connected router and network

In an unchanging topology, OSPF only sends updates every 30 minutes while RIP sends updates every 30 seconds. OSPF provides a significant savings in network overhead when compared to RIP.

Different Functions of OSPF Routers In an autonomous system running OSPF, routers can be assigned several different functions. An OSPF router can be assigned to route within an area (intra-area), between areas (interarea), or between autonomous systems (interautonomous system).

By default, the router performs intra-area routing. A router that routes between areas is an Area Border Router (ABR). Routing between autonomous systems is performed by a router that acts as an Autonomous System Boundary Router (ASBR).

In addition to its routing function, a router can function as the designated router (DR) or backup designated router (BDR) on a multiaccess network. (A multiaccess network is any network other than a point-to-point link, such as SMDS, X.25, Frame Relay, or a LAN.)

Figure 6-21 is an example of an autonomous system running OSPF, with routers configured as described in the preceding paragraphs. Detailed descriptions of ABRs, ASBRs, DRs, and BDRs follow the figure. A stub network is a network that only has one OSPF router; a multiaccess network has more than one OSPF router.

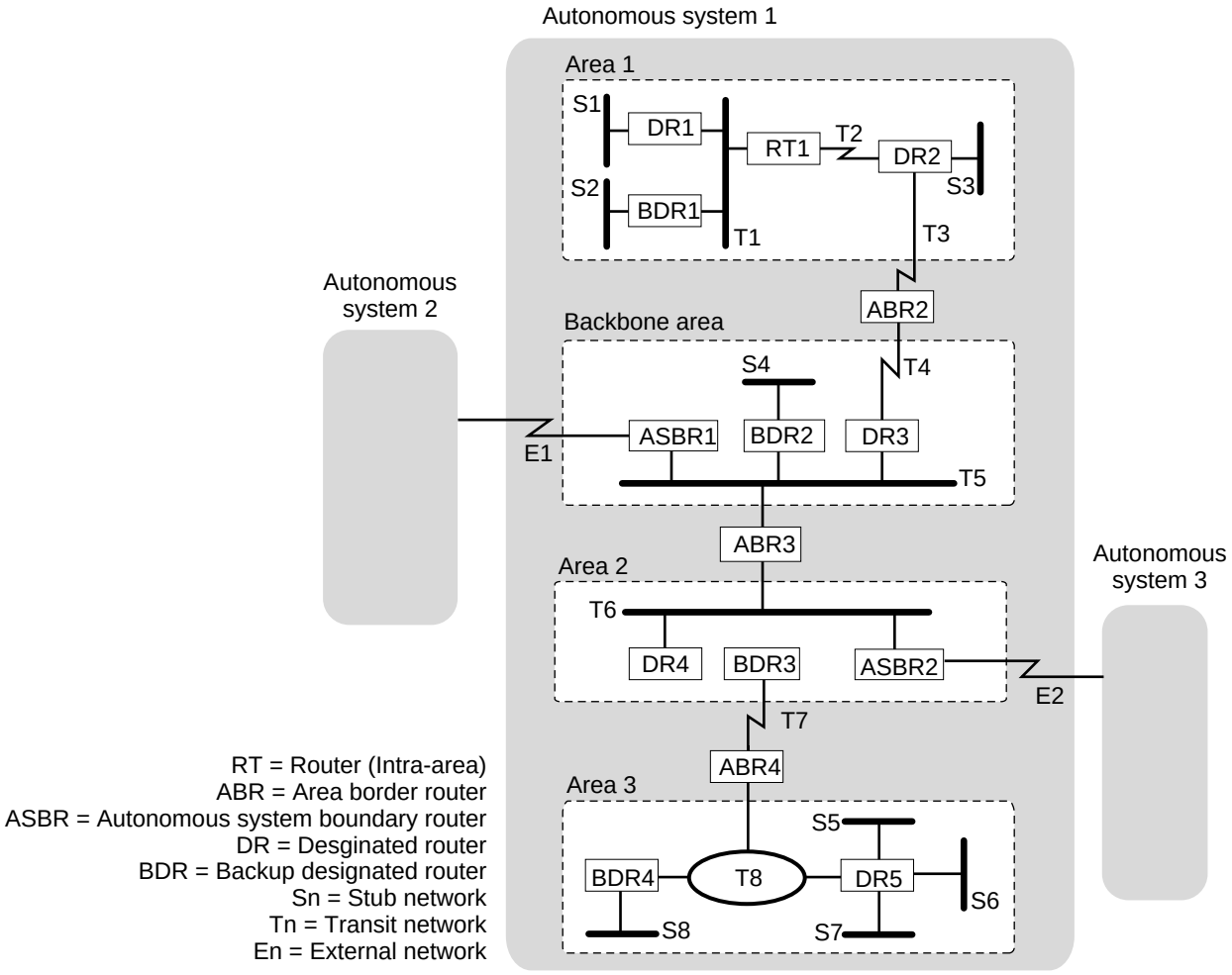


Figure 6-21 Autonomous System with Multifunctional OSPF Routers

Area Border Router An area border router (ABR) is a router that has interfaces in more than one area. For example, in Figure 6-21, ABR2 interfaces network T3, which is part of Area 1. It also interfaces network T4, which is part of the backbone area. (A router automatically acts as an ABR when different area numbers are assigned to different ports.)

Each ABR maintains a distinct database for each area to which it belongs. In the example shown in Figure 6-21, ABR2 maintains databases for Area 1 and the backbone area.

All ABRs should have at least one interface connected to the backbone area. However, if there are no interfaces of an ABR connected to the backbone area, you can configure a virtual link to provide complete connectivity.

A virtual link is established between two ABRs. One of the ABRs must be directly connected to the backbone area, which provides a link for the other ABR. Also, both ABRs must be part of at least one common nonbackbone area for the virtual link to be established.

For example, in Figure 6-22, ABR1 is an ABR for Areas 1 and 2. However, it is isolated from the backbone area. Because ABR1 and ABR2 are both connected

to Area 1 and ABR2 is connected to the backbone area, a virtual link can be established between ABR1 and ABR2. The VirtualLink parameter allows you to establish a virtual link between two ABRs. For more information on this parameter, refer to Chapter 41 in *Reference for NETBuilder Family Software*.

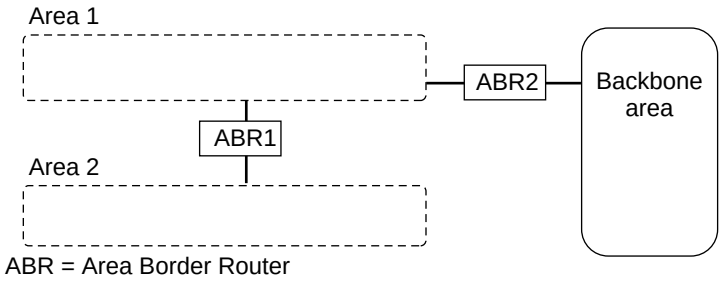


Figure 6-22 Backbone Area with One Isolated ABR

Autonomous System Boundary Router An autonomous system boundary router (ASBR) is a router that interfaces one or more routers in other autonomous systems. For example, in Figure 6-21, ASBR1 interfaces networks in Autonomous Systems 1 and 2. (A router automatically acts as an ASBR when different routing protocols [RIP, OSPF, IISIS, or BGP] are enabled on different ports.)

An ASBR can also function as an ABR if it is connected to more than one area in addition to being connected to another autonomous system.

Typically, an ASBR runs an interautonomous system routing protocol, such as BGP, on the interface that connects the other autonomous systems. On interfaces within the autonomous system that it is part of, the ASBR runs an intra-autonomous routing protocol, such as RIP, OSPF, or both.

In addition to its routing function, a router that is elected as the designated router (DR) on the multiaccess network performs the function of flooding for the network. A router that is elected as the backup designated router (BDR) on the multiaccess network should be adjacent to the same routers that the DR is adjacent to. The BDR has a subset of the DR's responsibilities. The BDR takes over the DR's role in the event of its failure. For details on the functions performed by the DR and the BDR, refer "Learning Routes and Network Reachability" next.

The value of the ROUTerPriority parameter determines which routers on a multiaccess network function as the DR and the BDR. For more information on the ROUTerPriority parameter, refer to Chapter 41 in *Reference for NETBuilder Family Software*.

Learning Routes and Network Reachability In an autonomous system running OSPF, intra-area routes, interarea routes, and interautonomous system routes are learned as described here.

Each router periodically exchanges a hello packet with its neighbor. The hello packet includes a list of all routers from which the originating router has recently received a hello packet. The exchange of hello packets establishes a bidirectional relationship between neighbors. The HelloTime parameter allows you to set the frequency at which hello messages are sent. If you modify the

setting of the HelloTime parameter, you must check the setting of the RouterDeadTime parameter. If the setting of the HelloTime parameter is larger than the setting of the RouterDeadTime parameter, the routers will not become fully adjacent. For more information on the HelloTime and RouterDeadTime parameters, refer to Chapter 41 in *Reference for NETBuilder Family Software*.

After bidirectional relationships are established between neighboring routers, each pair of neighboring routers must decide if they should form an adjacency (a formalized bidirectional relationship). Neighboring routers connected by a point-to-point network always form adjacencies. (A point-to-point network is a network where two routers are connected through a single network connection.) However, on a multiaccess network, adjacencies are formed only between DRs and BDRs and each of their neighbors. For example, in Figure 6-23, DR1 is the DR and BDR1 is the BDR for Network T1. Adjacencies are formed between DR1 and RT1, between BDR1 and RT1, between DR1 and RT3, between BDR1 and RT3, between RT1 and RT2, and between DR1 and BDR1. However, RT1 and RT3 have not fully established an adjacency with each other. They are in a state known as a two-way state.

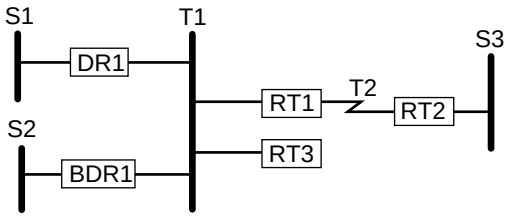


Figure 6-23 Forming Adjacencies on a Multiaccess Network

If adjacencies were formed between each router and its neighbor on a multiaccess network, it can be shown mathematically that the amount of traffic on the network would be significantly heavier. By minimizing these adjacencies, the DRs and BDRs reduce this traffic to manageable proportions.

Use the SHow -OSPF NeighborStatus command to display an OSPF neighbor status table, which shows the status of direct connect neighbor adjacencies for your router. For more information on the SHow NeighborStatus command and an explanation of OSPF neighbor status table entries, refer to Chapter 41 in *Reference for NETBuilder Family Software*.

After an adjacency is formed between a pair of routers, each router on a point-to-point network and the DRs and BDRs on a multiaccess network send out a link state advertisement to its neighbor every 30 minutes or whenever a change in topology occurs. External link state advertisements are flooded throughout the router's autonomous system. Other link state advertisements are flooded within a single area. For details on link state advertisements, refer to "Link State Advertisements" on page 6-54.

Each router in an area maintains an identical database of the area's topology. The database contains both the topology of the router's area and routes to

networks outside of the router's area. This database is used to build a shortest path tree. The router doing the computation uses itself as the root of the tree and builds each node of the tree based on the metric advertised in the link state advertisement.

The router always selects the path with the lowest metric. For details on metrics, refer to "Metrics" on page 6-55.

If there is more than one equal cost path, the router can use multipath routing and load splitting. For more information on these features, refer to "Multipath Routing" on page 6-40 and "Route Selection and Load Splitting" on page 6-41.

The router stores information on all reachable networks in its routing table.

The following types of networks are considered reachable:

- All directly connected networks, unless the network is shared by its neighbor and itself
- All static routes (if configured)
- All dynamic routes learned through RIP, OSPF, and IISIS in the routing table
- All dynamic routes learned through BGP (if configured)

For more information, refer to descriptions later in this chapter.

Reducing Network Costs Using Demand Interface Circuits In a remote office internetworking environment, many remote offices are connected to a central site through *demand circuits*, such as ISDN circuits or analog lines with modems, X.25 SVC or Frame Relay SVC neighbors, or dial-up lines. The cost of these demand circuits depends on the connection time or line usage.

OSPF periodically sends hello packets to refresh routing information, requiring the circuit to be constantly open, which results in unwanted usage charges. To reduce the cost of running OSPF over demand circuits and increase bandwidth, OSPF has been modified to operate more efficiently over demand circuits. When no network topology changes occur, OSPF sends no routing information traffic at all, allowing the data link connection to be closed when not required for application data traffic. As soon as data is sent, a data link connection is attempted. If the connection is successful, the data is sent and the circuit stays open. After a period of inactivity, the circuit is closed again to conserve cost and resources.

Using the -OSPF DemandInterface parameter, you can configure an interface to be a demand interface. The neighboring router must agree that the point-to-point link is a demand circuit by setting the DC bit defined in the OSPF Options field of router LSAs, OSPF hello packets, and database description packets as follows:

- In a router's self-originating LSAs, the DC bit is set if and only if the router can properly process LSAs having the DoNotAge bit set.

  If the DoNotAge bit is set, only truly changed LSAs are flooded over demand circuits. If a newly received LSA is only a periodic refresh, it is not flooded on attached demand circuits.

LSAs are not aged while they are held in the link state database, meaning they do not have to be refreshed, further reducing the routing traffic and the amount of time the circuit must remain up.

CAUTION: *Do not configure any interface on any router in a single OSPF area as a demand circuit (DC) interface unless all routers in that area have been upgraded to at least software version 8.3. Non-DC-aware routers become confused by LSAs using the DoNotAge bit in the link state age field. The LSA appears to expire and those routers are constantly flushing the LSA from their link state database and rerunning the Dijkstra algorithm, as well as informing all the routers they have adjacencies with of the routing changes. This affects every router in an area that cannot understand DC-style LSAs.*

- For hello and database description packets, the DC bit is set in outgoing packets if and only if the router wants to treat the attached network as a demand circuit and tries to negotiate with the neighboring router for the suppression of hellos on point-to-point demand circuits.

  Over point-to-point demand circuits, both end points must agree to suppress sending hello packets by setting the DC bit in OSPF hellos and database description packets. Receiving a packet with this setting indicates agreement, and OSPF hello packets are sent only until initial link state database synchronization is achieved with the neighbor. After the state of the neighbor connection reaches "full," hellos are suppressed and the data link connection to the neighbor is assumed to be available.

  For OSPF broadcast and nonbroadcast multiaccess (NBMA) networks that have been configured as demand circuits, the exchange of hello packets remains periodic for the proper operation of the DR election algorithm.

Link State Advertisements A link state advertisement indentifies the state of a router's interfaces and adjacencies. The types of link state advertisements sent out by a router depends on the function that the router has been configured to perform. Multiple link state advertisements can be contained in a link state update packet. There are four types of link state advertisements:

- Router link state advertisements

  Each OSPF router sends out a router link state advertisement. This advertisement describes its links to stub and multiaccess networks as well as links to other routers for a given area. (A stub network is a network that only has one OSPF router; a multiaccess network has more than one OSPF router.) The advertisement is flooded throughout the area the originating router belongs to.

- Network link state advertisements

  DRs on multiaccess networks send out network link state advertisements. These advertisements describe the routers on the network that are fully adjacent with the DRs. These advertisements are flooded throughout a single area.

- Summary link state advertisements

  ABRs send out summary link state advertisements. These advertisements summarize all the interarea routes for all the areas to which the router is attached. (Each available interarea route is summarized in a separate summary link state advertisement.) These advertisements are flooded throughout the area that the ABR interfaces.

- External link state advertisements

  ASBRs send out external link state advertisements. These advertisements contain information on destinations outside of the autonomous system the router resides in, including static and dynamic routes learned by RIP, BGP, and IISIS. (Each destination outside the autonomous system is described in a separate external link state advertisement.) These external routes are described using Type 1 or 2 external metrics. (For more information on Type 1 and 2 external metrics, refer to "Metrics.") These advertisements are flooded throughout the autonomous system the router resides in.

To view the short version of the link state database, enter:

```
SHow -OSPF LinkStateData
```

Metrics OSPF uses an administrative cost as its metric. The SETDefault -OSPF Cost command allows you to set the cost for a specific path.

The default value of the Cost parameter is 10^8/bandwidth of the medium that interfaces a port. For Ethernet, the default value is 10; for T1 lines, the default value is 65; for FDDI, the default value is 1.

For example, to set the cost on port 3, enter:

```
SETDefault !3 -OSPF Cost = 58
```

In this example, the T1 serial line that interfaces port 3 has been assigned the cost of 58. For more information on the Cost parameter, refer to Chapter 41 in *Reference for NETBuilder Family Software*.

The router running OSPF selects the route with the lowest total administrative cost to reach its destination. For example, imagine that OSPF learns about two intra-area routes to reach a particular destination. Route 1 has the administrative cost of 200, while Route 2 has the administrative cost of 300. The OSPF router will select Route 1 because it has the lowest administrative cost.

User Configurations Table 6-7 summarizes the OSPF parameters that allow you to customize the configuration of your OSPF router. For complete information on these parameters, refer to Chapter 41 in *Reference for NETBuilder Family Software*.

Table 6-7 OSPF Configuration Parameters

Parameter	Operation
AreaID	Determines the area to which a specified port on a router belongs.
Cost	Determines metrics (cost and type of service) associated with a specified port.
DEBUG	Determines the level of OSPF tracing that will be performed at the local console port.
DefaultMetric	Determines the metric for the default route.
Delay	Determines the delay in seconds for the specified port. The delay time is added to all link state advertisements before it is sent on an interface.
DirectPolicy	Determines whether a locally attached network should be advertised into the OSPF domain

(continued)

Table 6-7 OSPF Configuration Parameters (continued)

Parameter	Operation
ExteriorPolicy	Determines which networks learned through BGP are reported in external link state advertisements and what metric and metric type to use.
HelloTime	Changes the frequency at which hello packets are exchanged between neighbors on a network.
InteriorPolicy	Determines which networks learned through RIP and IISIS are reported in external link state advertisements and what metric and metric type to use.
Neighbor	Determines to which router on the directly connected network a router sends packets. Also determines how packet is addressed.
PassWord	Determines the password for a specified port to authenticate packets.
ReceivePolicy	Determines which networks from external link state advertisements are stored in routing tables and what metric to use.
RouterDeadTime	Changes the frequency for determining when a router is down.
ROUTerPriority	Determines the priority for a router on a specified port. The router with the highest priority becomes the DR for a multiaccess network.
StaticPolicy	Determines which static routes are advertised for a particular network and what metric is used.
StubDefaultMetric	Specifies whether or not the router should generate the default route and metric into the stub areas.
VirtualLink	Determines whether the specified port of a router acts as a virtual link between an area and a backbone area.

Configuring Integrated IS-IS for Dual IP and OSI Mode

If you are configuring dual IP and OSI mode, you must enable the IP forwarding process, enable Connectionless Network Protocol (CLNP), and have at least one IP network number or subnet mask configured before configuring IISIS.

IISIS is a protocol that provides integrated OSI-type routing for IP and OSI environments; it is the IP extension added to the original OSI IS-IS Protocol. IISIS routing simplifies network topology, reduces network management complexity, and reduces routing traffic overhead. In IP environments, IISIS is an alternative to other IP routing protocols, such as RIP and OSPF.

The original IS-IS routing protocol was developed by ISO to provide network layer connectivity in OSI environments. IS-IS is designed to work with CLNP and ES-IS. IS-IS is an international standard.

You can use IISIS in OSI mode for routing in pure OSI environments. You can use IISIS in dual IP and OSI mode for routing in environments where both types of networks are being used. In dual mode, for example, one router can serve IP and OSI subnets simultaneously and IISIS routes traffic between the two subnets. You can also use IISIS for IP environments only.

Autonomous System Routing Using BGP

The Border Gateway Protocol (BGP) is an interautonomous system routing protocol that is used to exchange routing information between different autonomous systems (ASs). A router can use BGP to determine the reachability of networks outside of its AS.

The sections that follow describe the following items:

- BGP overview
- BGP external and internal peers
- Peer-to-peer communication
- Path attributes (AS-PATH, ORIGIN, NEXT-HOP, MULTI-EXIT-DISC, LOCAL-PREF, ATOMIC-AGGREGATE, and AGGREGATION)
- Path selection
- Policies (interior, exterior, network number, AS-path)
- Route aggregation

BGP Overview

The BGP provides the following advantages:

- Consumes less bandwidth

  The BGP uses incremental updates to reduce the amount of routing information. When a BGP session is first established, the peers exchange the entire contents of their routing tables. After this initial data exchange, BGP peers only exchange changes to their routing tables, effectively reducing the size of their routing tables and consuming less bandwidth.

- Allows the detection of routing loops

  The BGP minimizes the occurrence of routing loops. In addition to network reachability information, the BGP Protocol requires that update messages contain a list of the ASs the routing information has traversed. Routing loops are eliminated because a router never selects a path that contains its own AS.

- Selects routes based on performance and policy constraints.

  The BGP allows a default weight to be added to all internal and external routes before computing the degree of preference for a route. If there are multiple routes to the destination networks, the route with the highest weight is chosen, allowing some routes to have higher priority than others.

The BGP allows user-configured AS-path policies and network number policies to be implemented. These policies determine whether a BGP speaker accepts and distributes routing information based on an AS-PATH attribute or IP network number.

External and Internal Peers

Two routers that exchange routing information using BGP are *peers*. Two kinds of peers exist:

- Internal peers

  Two routers residing in the same AS are internal peers. Internal peers do not need to be attached to the same network.

- External peers

  Two routers residing in adjacent ASs are external peers. External peers must be attached to the same network. BGP uses the common network to exchange messages between external peers.

Refer to Figure 6-24 as an example. In this figure, routers A and B are configured as internal peers. Routers B and C are configured as external peers.

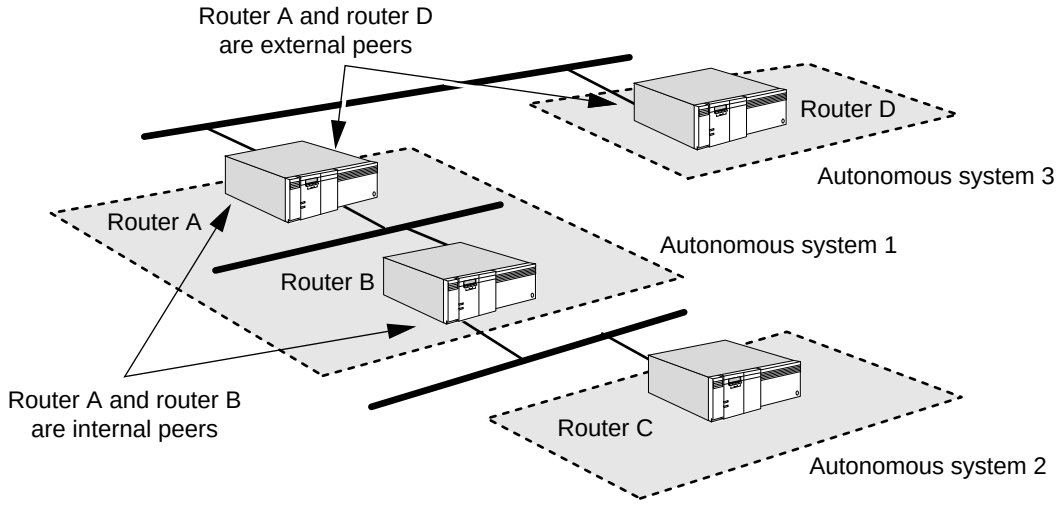


Figure 6-24 Internal and External Peers in an Autonomous System

Each peer establishes a BGP connection with the other peer. After the connection is established, the peers exchange update packets that indicate the networks each peer can reach. A peer also may report the networks that other gateways in other ASs can reach.

For example, in Figure 6-24, router B reports to router C all the networks that are reachable within autonomous system 1, and router C does the same for autonomous system 2. In addition to reporting the networks reachable within autonomous system 1, router B also reports to router C all the networks that are reachable through router A.

Peer-to-Peer Communication

After BGP peers are configured, three peer-to-peer communication states can be established between two peers:

- Connection establishment state
- Confirm state
- Established state

The router enters the connection establishment state immediately after you configure BGP and set up peers. In this state, Router B tries to establish a TCP connection with a configured gateway (Router C).

After a TCP connection is established, the routers exchange open messages in which the following information is exchanged:

- The version of BGP that a router wants to "speak"
- The hold time (maximum time for which a connection is kept open without receiving any keepalive or update packets)
- The AS number
- The Router ID

If the open messages are satisfactory, each peer enters the confirm state in which they exchange keepalive packets. When keepalive packets are received, the peers reach the established state, in which they exchange routing information in update messages.

The connection between peers is assumed to be a reliable TCP connection. Once the peers have exchanged routing tables, the only packets regularly exchanged (every 30 seconds) are keepalive packets. Routing updates occur only when new routes are reachable or previously advertised routes have become unreachable, which greatly reduces the amount of routing traffic and the time required to exchange and process information.

Update messages contain all reachable network addresses and the corresponding distances associated with each gateway, as well as the complete AS path for each network. Update messages may also contain explicit unreachable routes.

Path Attributes

For each route, the BGP uses a set of path attributes to describe the route. These attributes help to eliminate looping of routing information, assist with policy-based routing decisions, indicate the original source of the path information as well as the IP address of the next-hop router to the destination, provide routing metrics, simplify the route selection process, and perform route aggregation.

Path attributes are classified as either well-known or optional as shown in Figure 6-25:

- A well-known attribute must be recognized by all BGP implementations.

  The well-known attributes are further divided into mandatory and discretionary. A well-known, mandatory attribute (ORIGIN, AS-PATH, NEXT-HOP) must be included in every route description. A well-known, discretionary attribute (LOCAL-PREF, ATOMIC-AGGREGATE) may or may not be included in a route description depending on whether the attribute is implemented by the BGP speaker. A BGP speaker that receives a well-known path attribute is required to forward the attribute to its peer in update messages.

- An optional attribute may or may not be recognized by a BGP implementation.

  The optional attributes are divided into transitive and nontransitive. An optional transitive attribute (AGGREGATION) may be passed along unchanged by a BGP router that has not implemented the attribute. An optional nontransitive attribute (MULTI-EXIT-DISC) may not be passed along by a BGP router that has not implemented the attribute.

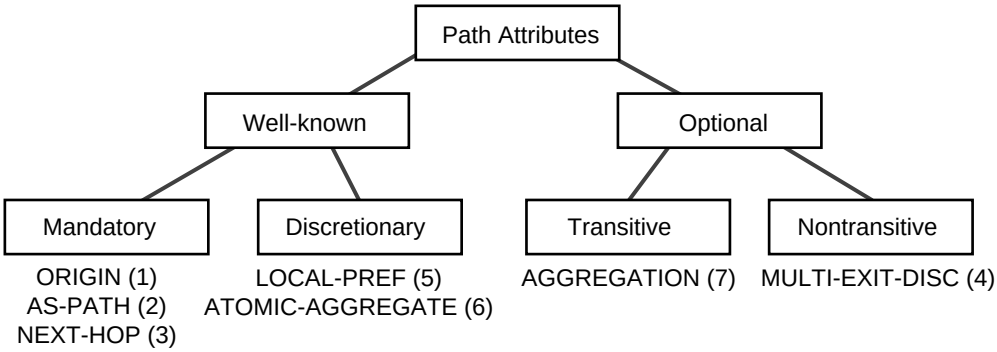


Figure 6-25 Path Attributes

You can display detailed information about path attributes associated with AS paths by entering the SHow -BGP ASPath Debug command.

AS-PATH The BGP Protocol uses the AS-PATH attribute to eliminate the occurrence of routing loops. As reachability information for a network traverses the internetwork, BGP creates a list of the ASs through which the routing information has passed. Each BGP speaker adds its own AS to the list before advertising network reachability to a peer as shown in Figure 6-26. The list of the ASs along the path to a destination network is called the AS-PATH attribute.

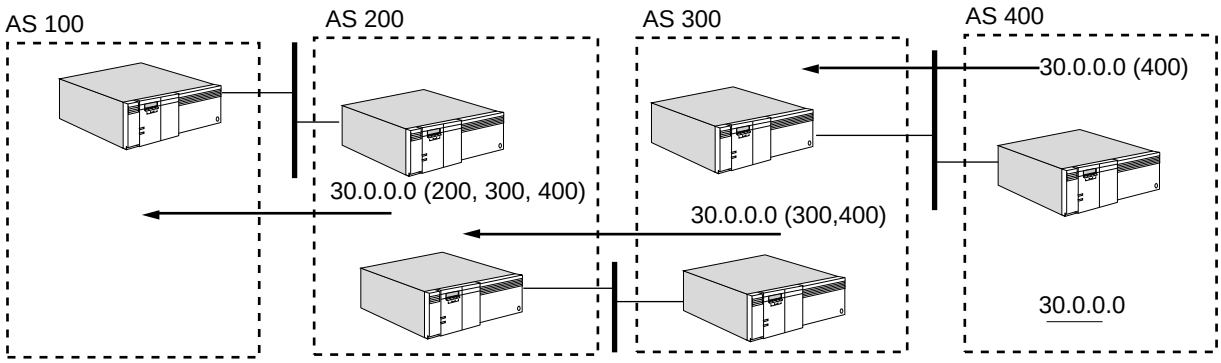


Figure 6-26 AS Path Example

The AS-PATH attribute is composed of a sequence of AS path segments. Each AS path segment may be either an AS SEQUENCE or an AS SET:

AS SEQUENCE — An *ordered* set of ASs that the route in the update message has traversed.

AS SET — An *unordered* set of ASs that the route in the update message has traversed. AS SETs are used by the route aggregation algorithm to reduce the size of the AS path information. An AS SET lists each AS number only once, regardless of how many times it may have appeared in the multiple AS paths that were aggregated.

An AS SET indicates that the destinations can be reached though paths that traverse at least some of the listed autonomous systems. AS SETs provide enough information to eliminate the looping of routing information.

The AS-PATH attribute helps suppress routing loops. A router never accepts a route with its own AS in the AS-PATH list. The AS-PATH attribute can be used to make policy-based routing decisions. For more information, refer to "AS-Path-Based Policies"on page 6-66.

ORIGIN The ORIGIN path attribute defines the original source of the path information. The ORIGIN path attribute may contain the values IGP or Incomplete. IGP indicates that the destination network was learned by the original BGP speaker from the Interior Gateway Protocol (IGP) running in the original AS; this routing information is considered to be trustworthy.

Incomplete indicates that the routing information was obtained from some means other than an IGP. For example, the route may have been learned using a static configuration.

NEXT-HOP The NEXT-HOP path attribute defines the IP address of the border router that should be used as the next-hop to the destination networks. A BGP speaker can use its own IP address or the IP address of another router attached to the same subnet. This attribute allows a BGP speaker to advertise routes through another border router attached to the same subnet. For example in Figure 6-27, routers B and C are both border routers for AS 200. However, only router C is a BGP speaker and has a session with router A. Router C advertises the route to network 130.7.0.0 with router B as the next-hop router.

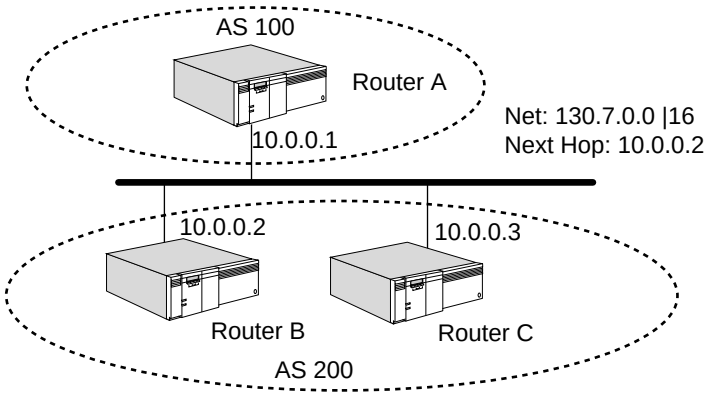


Figure 6-27 Next-Hop Router

MULTI-EXIT-DISC The MULTI-EXIT-DISC attribute provides metric support. It has limited function in the BGP Protocol because routing loops are suppressed using the AS-PATH attribute instead of metrics. As routing information traverses multiple ASs, the ability to select a route based upon the least cost metric is no longer possible. One AS may use hop count, another uses a delay, and a third uses an administratively defined cost. Because no universally accepted metric is used, direct comparison of the combined metrics for routes using different paths has no real meaning.

The MULTI-EXIT-DISC attribute allows a BGP speaker to advertise a metric along with a route only if the network is internal to the same AS as the BGP speaker. If an AS has multiple BGP speakers to a neighboring AS, different speakers may advertise the same network with a different metric. Typically, BGP speakers select their metric based on the inter-AS cost to reach the destination network.

If the border routers of an AS receive different metrics for the same network, they compare the different metrics. The result of the comparison along with other factors determines the "best" route. If the MULTI-EXIT-DISC attribute is received over an external link, it may be propagated over internal links to other BGP speakers. However, it is never propagated to other BGP speakers in neighboring ASs.

In Figure 6-28, routers A and B advertise a route to network 30.0.0.0, which is completely contained within AS 300. Router A advertises the route with a MULTI-EXIT-DISC of 10. Router B advertises the route with a MULTI-EXIT-DISC of 5. The border routers in AS 200 make a comparison between the two metrics to select a better entry point in AS 300. However, the BGP speakers in AS 200 never propagates this metric to AS 100.

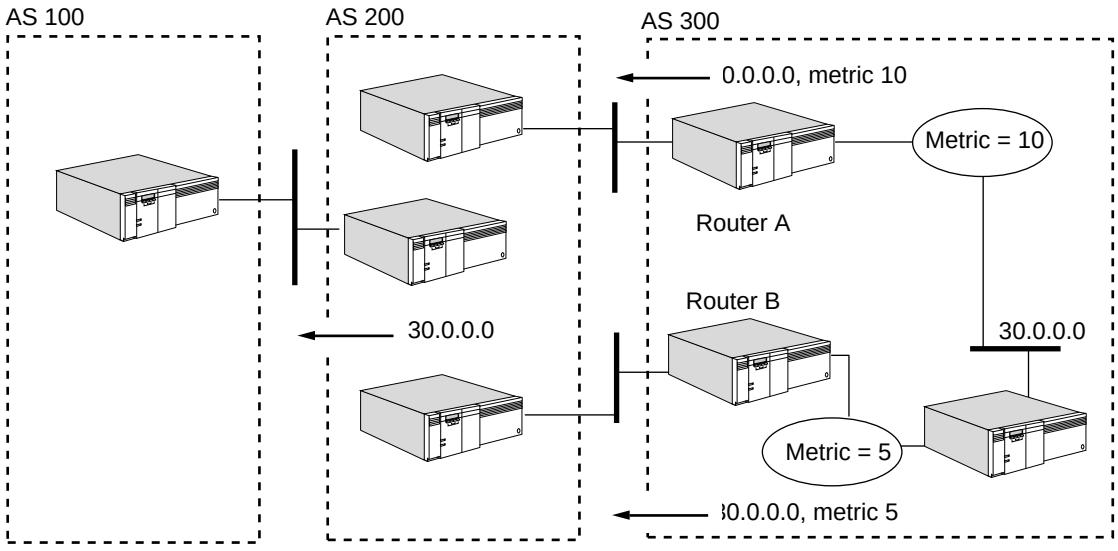


Figure 6-28 BGP Routing Metric

LOCAL-PREF The LOCAL-PREF attribute simplifies the route selection process. It advertises a degree of preference for each external route to BGP peers in the same AS so that a route with a higher degree of preference is selected over a route with a lower degree of preference.

This attribute is included as part of all update messages sent to other BGP speakers located within the same AS and never advertised to BGP peers in an adjacent AS.

ATOMIC-AGGREGATE The ATOMIC-AGGREGATE attribute is attached to a less specific route before propagating it to other BGP speakers to ensure that the aggregate is not deaggregated by other BGP speakers.

If a BGP speaker is presented with a set of overlapping routes from one of its peers, the more specific route takes precedence. If the BGP speaker selects the less specific route, the router attaches the ATOMIC-AGGREGATE attribute.

AGGREGATION The AGGREGATION attribute allows a BGP speaker performing route aggregation to advertise the AS that performed the aggregation. Aggregation is the process of combining several different routes so that a single route with a shorter mask can be advertised.

Path Selection

One of the most important tasks of BGP is to select the best path to a destination network based on the AS topology. In traditional routing protocols, each path has only a single metric to represent its cost. To evaluate two paths, the router compares the two metrics and selects the path with the lowest cost metric.

In interdomain routing, no universally agreed-upon metric among ASs can be used to evaluate different paths to a network. Therefore, each AS may implement its own set of criteria for path selection.

Path selection for the 3Com BGP-4 implementation is based on the following criteria in order of priority:

User-defined policies	Policies that are configured to control the distribution of routing information affect the paths that are available and the path selection process.
AS weight factor	When multiple paths exist to a given network, you can assign weights for AS paths or subsets of AS paths. The weight for a path is calculated by summing all the individual AS-path weight expressions that are applicable for the path. The path having the highest weight is selected.
AS count	If competing paths have the same weight, the path with a lowest total AS count is preferred over paths that have a larger AS count. BGP considers the path with a lower number of AS hops to be shorter.
link type	If competing paths have the same AS count, BGP prefers paths that arrive over external links to those that arrive over internal links.
path origin	If competing paths arrive over the same type of link, paths that are originated by an IGP Protocol has precedence.
MULTI-EXIT-DISC	If competing paths have the same ORIGIN attribute, BGP selects the path with the lowest MULTI-EXIT-DISC attribute value.

BGP-ID — If all of the above result in a tie, the selection process gives preference to the path from the peer with the larger IP address (BGP-ID). The IP addresses are compared as unsigned 32-bit integers.

Policies

Policies determine whether AS-level routing information is accepted and distributed by a BGP speaker. Policies control routing information in two ways:

- Routing information can be received by a BGP speaker but not added to the routing table.
- Routing information can be received and added to the routing table but only advertised to some of the router's BGP peers.

BGP stores its routes in a routing information database (RIB), which is conceptually divided into three distinct parts as shown in Figure 6-29.

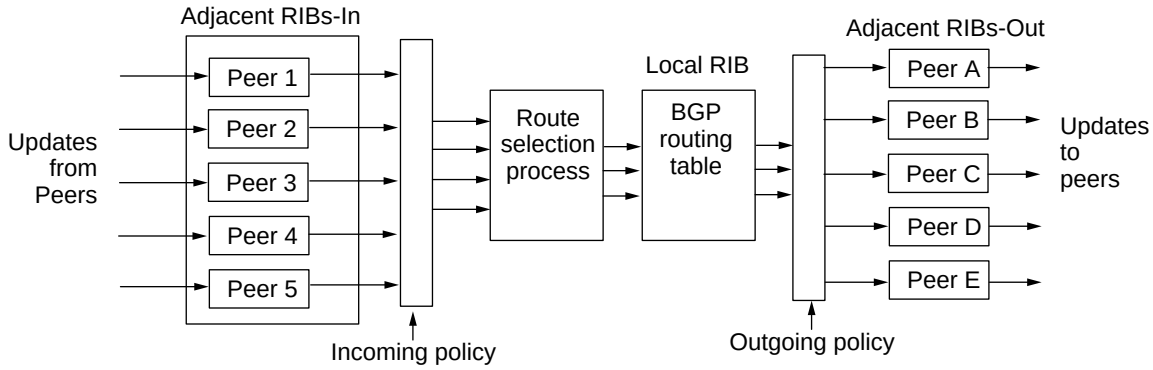


Figure 6-29 BGP Routing Selection and Policies

The Adjacent RIBs-In contains unprocessed routing information that has been received by the local BGP speaker from its peers. The information is learned from inbound update messages and represents routes that are available for input to the route-selection process of the local BGP speaker.

The Local RIB contains routes selected for use by the route-selection process of the local BGP speaker.

The Adjacent RIBs-Out organizes the routes that the local BGP speaker has selected for advertisement to its neighboring BGP speakers using outbound update messages.

Incoming policies are applied as part of the path selection process to manage the flow of information from the Adjacent RIBs-In to the Local-RIB. Outgoing policies are applied to manage the flow of information from the Local-RIB to the Adjacent RIBs-Out. The redistribution of routes is performed only on routes that have been placed in the Adjacent RIBs-Out.

The following list describes the three important effects of this sequence of operations:

- Incoming filters are implicitly applied before re-advertisement takes place.
- Only those routes used by the local BGP speaker are considered for re-advertisement.
- Only one outgoing route is advertised for each network even if many incoming paths for that route are learned from a variety of peers.

Interior Policies BGP learns network reachability information from many sources:

- Internal or external BGP speakers
- IGP speaker (RIP, OSPF, IISIS) residing in the same router
- Static route configurations
- Directly attached networks

To control the flow of routing information into BGP, an interior policy can be configured using the InteriorPolicy parameter. This parameter controls the blocking of IGP routes (RIP, OSPF, IISIS), static routes, and directly connected networks into BGP for advertisement to BGP speakers residing in adjacent autonomous systems.

By default, BGP imports all IGP-derived, directly connected, and static routes for advertisement by BGP. To avoid this, selectively configure only those networks to be imported using the -BGP InteriorPolicy parameter.

Exterior Policies Each IGP Protocol controls the import of BGP routes through the configuration of its ExteriorPolicy parameter (in the RIPIP, OSPF, and IISIS Services).

BGP aggregated routes can only be leaked into IGP domains if the IGP routing protocol supports a mask along with each network route. OSPF and IISIS support this feature, but RIPIP does not provide this information in update packets. You cannot export aggregated routes into domains that run RIPIP.

Network Number-Based Policies Network number policies provide filtering of incoming and outgoing BGP advertisements based on IP network numbers. The following types of network filters can be configured:

- Do not accept routes for network x.x.x.x from BGP peer A.
- Do not advertise a route for network z.z.z.z to BGP speaker B.

The network number specified in these examples can be a single network number or a range of network numbers specified by a CIDR address prefix.

Network number policies are configured using the NetworkFilter, NetPolicyAll, NetPolicyExt, NetPolicyInt, and NetPolicyPeer parameters. The NetPolicy parameters allow you to configure the policy on all peers, external peers, internal peers, or on a specific peer.

AS-Path-Based Policies The AS-path policy provides filtering based on information contained in the AS-PATH attribute in each update message. Typical policies contain a combination of the following elements:

- Source AS
- Destination AS
- AS presence (within the AS-PATH attribute)
- Advertise or receive

Using these elements, the following policies can be configured:

- Distribute routes from AS 2 only to ASs 3, 6, and 7.
- Accept only those routes from AS 4 that have AS 7 contained in the path.
- Do not accept routes requiring a path through AS 3 from AS 10.
- Distribute routes containing AS 5 only to ASs 3 and 4.

To maintain consistent routing information within an AS, do not apply accept or deny policies to internal BGP peers.

AS-path policies are configured using the AsFilter, AsPolicyAll, AsPolicyExt, AsPolicyInt, and AsPolicyPeer parameters.

For more information about the AS-PATH attribute, refer to "AS-PATH" on page 6-60.

Route Aggregation

BGP route aggregation uses the Classless InterDomain Routing (CIDR) route aggregation strategy to combine several different routes so that a single route with a shorter mask can be advertised. By combining several networks into one supernet, the number of BGP messages sent to peers and the size of the routing table is reduced. Unnecessary details about subnets are hidden from peers. CIDR is a method of using IP addresses without regard to traditional address classes that helps reduce routing table growth by summarizing several networks or subnets with a single routing update.

Supernetting can only be understood and supported by protocols that carry mask information along with routes, such as OSPF and IISIS. The RIP Protocol always interpret routes as Class A, B, or C and cannot fully interpret routes with supernet masks. Do not configure BGP route aggregation in this situation.

BGP routers learn all the subnet routes through an intradomain routing protocol, such as OSPF, or static configuration. The BGP router may advertise to its BGP peers a single aggregate route that describes all the destinations connected to it. When a BGP router performs route aggregation, it needs to know the range of block of IP addresses to be aggregated or not aggregated.

The BGP router should aggregate as many routes as possible except those that cannot be treated as part of a single unit due to multi-homing, policies, or other constraints.

Aggregation should never encompass Class D address space (224.0.0.0 through 239.255.255.255).

Address Resolution

To resolve Internet addresses with associated Ethernet addresses when routing, the router uses the Address Resolution Protocol (ARP) as described in RFC 826.

Configure the -ARP CONTrol parameter to decide whether the router supports proxy ARP requests on the specified interface. A proxy request is a request for a target Internet address that is not on the subnet where the request originated. If the router generates proxy replies, it replies with its own Ethernet address, provided that it has a route in the routing table for the target subnet.

ARP determines the destination's Ethernet header format to be used by sending out ARP requests that include the format. Specify the format by configuring the RequestFormat parameter in the ARP Service. The system replying to the request then uses the Ethernet header format it supports, and the router records the IP address, the Ethernet address, and the Ethernet header format of the replying system. The information is valid for the time specified by the HoldTime parameter in the ARP Service. Configure the time if you want the router to hold the information for more than or less than 24 hours, which is HoldTime's default value.

Inverse ARP

Inverse ARP is an adaptation of ARP that resolves DLCIs on Frame Relay networks to IP addresses, as described in RFC 1293.

Extended ARP

Extended ARP is an adaptation of ARP that resolves internet addresses to E.164 addresses on SMDS networks, as described in RFC 1209.

For more information on the parameters discussed in this section, refer to Chapter 6 in *Reference for NETBuilder Family Software*.

Other Global Router Configurations

After you determine how the router should route packets on each of its interfaces, you can influence the global router operation in several areas, as follows:

- Treatment of Internet Control Message Protocol (ICMP) request packets and generation of packets

  Configure the -IP ICMPReply parameter to control whether the router responds to Address Mask Request and Information Request packets. The router supports all the ICMP messages described in RFC 1009.

  Configure the -IP ICMPGenerate parameter to control whether the router originates ICMP ReDirect, Destination Unreachable, and TimeExceed packets.

- How long the IP layer waits for all IP fragments

  Configure the -IP ReassemblyTime parameter to control the length of time the IP layer waits for all IP fragments of an IP datagram to be received. This parameter applies only to packets specifically destined for the local router.

- The level of security

  Use the parameters beginning with "Sec" in the IP Service to configure the system for IP security options processing.

- The value of time-to-live (TTL)

  Configure the -IP DefaultTTL parameter to specify the value the router puts in the TTL field of an IP packet when it generates the packet.

- Prioritization of packets within the IP Protocol

  Some actions of the -IP FilterAddrs parameter enable your bridge/router to do special processing of IP packets over WAN links, improving the IP WAN traffic management. When the FilterAddrs parameter is used with the -IP Filters parameter, you can specify the packets to which the special processing applies.

For more information on the parameters discussed in this section, refer to Chapter 29 in *Reference for NETBuilder Family Software*.

7 BUILDING INTERNET FIREWALLS

This chapter describes how to configure an Internet firewall on a NETBuilder II bridge/router and a model 227 SuperStack II NETBuilder bridge/router. This chapter provides a conceptual overview of a firewall and gives guidelines for operating and managing it successfully.

For conceptual information, refer to "How A Firewall Works" on page 7-8.

Setting Up an Internet Firewall

The procedure in this section describes how to configure an Internet firewall. To configure the NETBuilder II bridge/router and SuperStack II bridge/router to perform firewall functions, you must set parameters in the FireWall Service.

Figure 7-1 shows two levels of firewall protection set up using the NETBuilder II bridge/router and the SuperStack II bridge/router. The first firewall is the model 227 SuperStack II bridge/router, which connects the Internet to a server subnet. The server subnet is where most Internet servers, such as the mail server and the WWW server, are located. The firewall on the SuperStack II bridge/router is enabled on port 3.

The second firewall is the NETBuilder II bridge/router, which connects the server subnet to the internal corporate network. The firewall on the NETBuilder II bridge/router is enabled on port 7. Traffic can flow freely between the other interfaces on the NETBuilder II bridge/router because the firewall will not be enabled on those interfaces. This permits the NETBuilder II to not only perform firewall functions but also perform high-speed routing for internal networks.

Although the NETBuilder II bridge/router is a secondary firewall, it is really the primary defense for internal networks since the internal servers are directly reachable from the Internet and they have a much higher chance of being compromised. Access from these internal servers to internal networks should be limited, and the servers should be configured using secured applications.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Read the information beginning with "How A Firewall Works" on page 7-8 through "Setting Up System Logs" on page 7-15.
- Log on to the system with Network Manager privilege.
- Configure your ports and paths as described in Chapter 1.

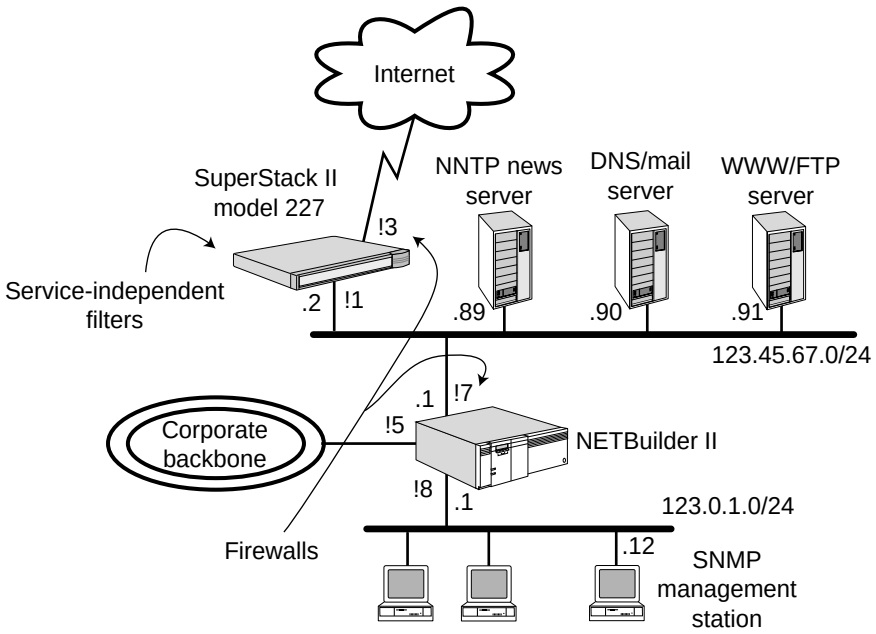


Figure 7-1 NETBuilder bridge/router Firewall Example

Defining Your Firewall Stance

You can choose between two firewall stances: "Everything not specifically permitted is denied" or "Everything not specifically denied is permitted." The stance assumed in this chapter is "Everything not specifically permitted is denied."

To define the basic stance of your firewall and decide whether log messages will or will not be recorded, follow these steps:

1. On the SuperStack II bridge/router, use:

```
SETDefault !<port> -FireWall DefAction = (Deny,NoLog)
```

2. On the NETBuilder II bridge/router, use:

```
SETDefault !<port> -FireWall DefAction = (Deny,NoLog)
```

The deny stance means that after all of the filters have been applied to a packet, and no actions have been taken, the packet must be dropped. You can explicitly deny specific types of traffic within your rules, which would stop traffic that you find dangerous or unnecessary before the system has to check that traffic against all of the other rules.

Refer to the DefAction and Log parameters in Chapter 24 of the *Reference for NETBuilder Family Software* for more information.

Continuing Routing Functions

Even while operating as a firewall, the NETBuilder II bridge/router and SuperStack II bridge/router must continue to perform the functions they were originally designed to perform, that is, they must continue to execute routing protocols so that they can correctly forward packets.

If your bridge/router is running OSPF, refer to the OSPF parameter. If your bridge/router is running RIP, refer to the RIP parameter. Both parameters are in Chapter 24 of the *NETBuilder Family bridge/router Reference Guide*.

Assuming OSPF is the routing protocol in use, to allow OSPF packets to come in and go out on all LAN interfaces, follow these steps:

1 On the SuperStack II bridge/router, enter:

```
ADD !3 -FireWall OSPF Permit
```

2 On the NETBuilder II bridge/router, enter:

```
ADD !7 -FireWall OSPF Permit
```

If RIP is being used, the syntax is the same. If BGP-4 or EGP is being used, you need to write a generic filter to allow any of these protocols to work properly. Refer to "Generic Filters" on page 7-11.

Configuring OAM Procedures

Your operations, administration, and maintenance (OAM) procedures must keep working. Examples of OAM procedures include Telnet, Internet Control Message Protocol (ICMP) (Ping), File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), and Simple Network Management Protocol (SNMP). For information on how to configure a firewall for these services, refer to the TelnetIn and TelnetOut, FTPIn and FTPOut, TFTP, ICMP, and SNMP parameters in Chapter 24 of the *Reference for NETBuilder Family Software*.

In the following procedure steps, 123.0.1.0/24 is the IP address of the management subnet, and 123.0.1.12/32 is the IP address of the SNMP management station. The syntax *"a.b.c.d/x"* denotes "the address *a.b.c.d* with the top *x* bits significant for comparison".

The management station does not need to be directly attached to one of the firewall bridge/routers. The management subnet could be anywhere on the corporate network. For the purpose of this procedure, we assume only that there is such a network with a common IP prefix/mask.

All of the filters used in the following steps are designed to control traffic to and from the bridge/routers. At this stage, no attempt is made to control traffic through the bridge/routers; that procedure is covered in "Blocking Unwanted Traffic" starting on page 7-6.

Configuring Telnet

To configure Telnet on both bridge/routers, complete this step:

- To allow users on the management LAN to Telnet into the SuperStack II bridge/router, on the NETBuilder II bridge/router enter:

```
ADD !7 -FireWall TElnetOut Permit From 123.0.1.0/24 to 123.45.67.2
```

This command guarantees that the NETBuilder II bridge/router will not block traffic from management stations to the SuperStack II bridge/router. No blocking actions are taken on Telnet traffic to the NETBuilder II bridge/router itself, because the firewall is not enabled on port 8.

Configuring TFTP

NETBuilder II and SuperStack II bridge/routers sometimes use TFTP to perform file transfers or network booting. The bridge/router always functions as a TFTP client, while the management station functions as a TFTP server.

To configure TFTP on both bridge/routers, complete this step:

- To accept TFTP packets from the SuperStack II bridge/router to the management station, on the NETBuilder II bridge/router enter:

```
ADD !7 -FireWall TFTP Permit From 123.45.67.2 To 123.0.1.0/24
```

This command allows both TFTP request packets (123.45.67.2 -> 123.0.1.0/24) and TFTP response packets (123.0.1.0/24 -> 123.45.67.2). The NETBuilder II bridge/router will have no problem accessing the management server because the firewall is not enabled on port 8.

Configuring ICMP (Ping)

To configure ICMP on both bridge/routers, complete this step:

- Allow the SuperStack II bridge/router to send and receive ICMP (Ping) messages; on the NETBuilder II bridge/router enter:

```
ADD !7 -FireWall ICMP Permit
```

Configuring SNMP

SNMP (another UDP-based protocol) has two sides. On the receive side, there are management requests. On the transmit side, there are two kinds of packets: responses to management requests and traps. The following two filters allow the SuperStack II bridge/router to send SNMP traffic to the SNMP management station and receive traffic from there. 3Com is explicitly denying SNMP that originates elsewhere on the Internet.

To allow the SuperStack II bridge/router to be SNMP manageable, on the NETBuilder II bridge/router enter:

```
ADD !7 -FireWall SNMP Permit From 123.45.67.1 To 123.0.1.12/32
ADD !7 -FireWall SNMP Permit From 123.0.1.12/32 To 123.45.67.1
```

Configuring FTP

FTP can be used to move files between any bridge/router and an FTP server.

If you do not use the FTP feature on your bridge/routers, you can skip this section.

The SuperStack II bridge/router needs to be able to FTP to and from the management subnet.

On the NETBuilder II bridge/router enter:

```
ADD !7 -FireWall FTPIn Permit From 123.45.67.1 To 123.0.1.0/24
```

Verifying the Configuration

After the OAM filters have been defined, turn on IP Firewall and verify the configuration of both bridge/routers by following these steps:

1. Turn on the firewall on the SuperStack II bridge/router and the NETBuilder II bridge/router respectively by entering:

```
SETDefault !3 -FireWall CONTrol = Filter
SETDefault !7 -FireWall CONTrol = Filter
```

2. Verify current functionality on both bridge/routers.

   Each of the following steps must be completed on the SuperStack II bridge/router and the NETBuilder II bridge/router.

a Ensure that both bridge/routers have all of their routes in the IP routing table by entering:

```
SHow -IP AllRoutes
```

b If you are using OSPF, ensure that both bridge/routers have all of their OSPF adjacencies by entering:

```
SHow -OSPF NeighborStatus
```

c Verify that Telnet works between the bridge/routers and the management subnet.

d Verify that TFTP works between the bridge/routers and the management subnet by entering:

```
COpy 123.0.1.76:test a:test
```

e Verify that PING works between the bridge/routers and the management subnet by entering:

```
PING 123.0.1.17
```

f Verify that SNMP management of the bridge/routers works.

- Are the bridge/routers still "green" in the management application?
- If you take down an interface on the bridge/router by removing a cable, do you see a trap on the SNMP management station?

g Verify that you can use FTP from the bridge/routers.

- Set up firmware.
- Perform a PUT test by entering:

  ```
  PUT <filename>
  ```

- Perform a GET test by entering:

  ```
  GET <filename>
  ```

Checking the Overall Status

To check the overall status of your configuration, follow this step:

- Display the settings of the parameters in the FireWall Service using:

  ```
  SHow !<port> -FireWall CONFiguration
  ```

  A display similar to the following appears:

```
=============Firewall Configuration for Port !2================
CONTrol (1st priority)------------------------------Permit-Deny-Log
Filter                                          -          -          -
IgnoreSrcSpoofing                               -          -          -
DenyTinyFragment                                -          0          0
DenySrcRoute                                    -          0          0
DenyRecordRoute                                 -          0          0
DenyTinyStamp                                   -          0          0
DenyIPTunnel                                    -          0          0
SuppressICMP (0 ICMP msgs issued)               -          0          0
Services (2nd priority)----------Active@-------idle--Permit-Deny-Log
FTPOut                                     Permit Log   Mar25 12:21   19min
Filters (3rd Priority)-----Bytes-Active@-------idle--Permit-Deny-Log
InFilter good                                              34          Mar25 12:21
OutFilter                                                  0           -
DefAction (4th priority)----------------------------Permit-Deny-Log
Deny                                            0          0          0
```

Blocking Unwanted Traffic

To block unwanted traffic on the SuperStack II bridge/router and NETBuilder II bridge/router, follow these steps:

1 Configure external protection for the SuperStack II bridge/router.

TCP-based services such as TELnet, FTP, SMTP, NNTP, DNS, Gopher, and Archie are much safer than non-TCP-based services. Avoid using non TCP-based services on your Internet connections. Due to the CPU requirements of DenySrcSpoofing, 3Com recommends that it be turned on only where absolutely required. Turn on DenySrcSpoofing on interfaces receiving external traffic to repel source spoofing attacks. Internal interfaces may not require such checking.

The filters in this section apply to traffic through the bridge/router, not to and from traffic as described earlier.

a Block hacker tricks.

These are nonintrusive filters that only adversely affect traffic from those individuals who are trying to break into your site. All normal traffic will proceed as usual.

On the SuperStack II bridge/router enter:

```
SETDefault !3 -Firewall CONTrol = DenySrcSpoofing
```

On the NETBuilder II enter:

```
SETDefault !1 -FireWall CONTrol = DenySrcSpoofing
```

b Allow remote secondary name servers to talk to your external name server.

These commands allow TCP-based domain name service (DNS) server-to-server traffic to be sent between 123.45.67.90 (the IP address of the name server in this example) and several remote name servers (for example, two off-site secondary name servers). Multiple name servers can be accommodated by adding more of these commands.

To configure the SuperStack II bridge/router, use:

```
ADD !1 -Firewall DNSSvrSvr Permit From 123.45.67.90/32 To <IPaddr>
ADD !1 -Firewall DNSSvrSvr Permit From 123.45.67.90/32 To <IPaddr>
```

2 Allow mail to be sent from the Internet to an external mail host on both bridge/routers.

SMTP is a particularly vulnerable service on many UNIX workstations because users run it mostly everywhere, and common vendor-supplied versions have many security holes in their implementations (not referring to the protocol here). Because of this vulnerability (and the difficulty of keeping all of the internal machines up-to-date with the latest version of "sendmail"), 3Com only allows hosts on the Internet to establish SMTP connections to the external mail host.

To configure the SuperStack II bridge/router, enter:

```
ADD !3 -FireWall SMTPOut Permit From 123.45.67.90/32
ADD !3 -FireWall SMTPIn Permit To 123.45.67.90/32
```

The NETBuilder II bridge/router must be configured to allow SMTP connections from anywhere on the corporate network (123.0.0.0/8 in this example) to anywhere on the Internet. You can also force all internal mail to go through the external mail host, but that is not usually a requirement. Allow SMTP to come in from the corporate backbone. On the other LAN interfaces, with "special

treatment" for the perimeter network, only accept SMTP traffic in from the main mail host, and only to machines within the corporate address (123.0.0.0/8). On all other interfaces, SMTP is allowed to go out as long as it is coming from within the corporate address.

To configure the NETBuilder II bridge/router, enter:

```
ADD !7 -Firewall SMTPIn Permit From 123.45.67.90/32 To 123.0.0.0/8
ADD !7 -Firewall SMTPOut Permit From 123.0.0.0/8
```

3 Configure Hypertext Transfer Protocol (HTTP) and World Wide Web (WWW) connections on both bridge/routers.

Allow remote HTTP connections to and from the WWW server only; allow internal WWW browsers to go out.

On the SuperStack II bridge/router, allow HTTP traffic from the Internet to the external WWW server only. Block traffic to the rest of the corporate network.

To configure the SuperStack II bridge/router, enter:

```
ADD !3 -Firewall HTTPIn Permit To 123.45.67.91/32
ADD !3 -Firewall HTTPOut Permit
```

The first filter allows the Internet to access your WWW server; the second filter allows your internal users to originate HTTP traffic to anywhere on the Internet.

To configure the NETBuilder II bridge/router, enter:

```
ADD !7 -Firewall HTTPOut Permit
```

This command explicitly permits HTTP out through the NETBuilder II bridge/router. If this step is not taken, the DefaultAction parameter blocks all HTTP traffic in and out on all ports.

4 Allow remote news feeds to get to the external news server.

This step is optional.

Allow the internal Network News Transfer Protocol (NNTP) server to connect over the SuperStack II bridge/router LAN interface in the outgoing direction by entering:

```
ADD !3 -Firewall NNTPOut Permit From 123.45.67.89/32
```

Allow the external NNTP servers to make connections through the SuperStack II bridge/router Internet link, but only to the external NNTP server using:

```
ADD !3 -Firewall NNTPIn Permit From <IPaddr> To 123.45.67.89/32
ADD !3 -Firewall NNTPIn Permit From <IPaddr> To 123.45.67.89/32
```

The <IPaddr> variable in the first command is the IP address of the first external news feeder. The <IPaddr> variable in the second command is the IP address of the second news feeder. If you have 10 external news feeds, then you will need 10 filter rules like the first two.

Allow NNTP traffic to cross the NETBuilder II bridge/router by entering:

```
ADD !7 -Firewall NNTPIn Permit
ADD !7 -Firewall NNTPOut Permit
```

If NNTP is not explicitly permitted using these rules, the DefaultAction parameter settings deny it.

How A Firewall Works

The firewall allows users inside a private network to have outbound access, while restricting outside users from inbound access. The types of incoming and outgoing traffic can be identified as follows:

- Inside-originated request to an outside service
- Outside reply to the inside-originated request
- Outside-originated request to an inside response
- Inside reply to the outside request

Firewalls are typically constructed on bastion hosts and a multiprotocol router. The bastion hosts, usually UNIX, are configured to prevent it from being compromised by outsiders and to provide detailed logging of system activity for security monitoring. The host may serve as an externally accessible server for FTP, e-mail, or the WWW.

Another common firewall component is a packet-filtering router. The multiprotocol router has extensive filtering capabilities to limit the type and direction of traffic that passes through it. The router usually is not the object of an attack, but can serve as a barrier to other, more desirable targets, or as the basis of a denial-of-service attack.

Packet-Filtering Routers

The packet-filtering router can make a permit or deny decision for each packet it receives. The router examines each datagram to determine if it matches one of its packet filtering rules. The filtering rules are based on the packet header information that is made available to the IP forwarding process. This information consists of the following items:

- IP source address
- IP destination address
- Incoming interface of the packet
- Outgoing interface of the packet
- Encapsulated protocol (TCP, UDP, ICMP, or IP tunnel)
- TCP/UDP source port
- TCP connection Establishment packets
- TCP/UDP destination port
- ICMP message type

If a match is found and the rule permits the packet, then the packet is forwarded according to the information in the routing table. If a match is found and the rule denies the packet, then the packet is discarded. If there is no matching rule, a user-configurable "default action" parameter determines whether the packet is forwarded or discarded.

Benefits of Packet-Filtering Routers

The majority of Internet firewall systems use only a packet-filtering router. Other than the time spent designing the filters and configuring the router, little or no cost is required to implement packet filtering because the feature is included as part of standard router software releases. Because Internet access is usually provided over a WAN interface, there is little impact on router performance if

traffic loads are moderate and few filters are defined. A properly designed firewall using a packet-filtering router can be transparent to end users and applications, so it does not require specialized user training or require that specific software be installed on each host.

Firewall Filter Types

You can configure your firewall with the following three types of filters:

- Service-independent filters
- Predefined (service-dependent) filters
- Generic filters

This section describes each type of filter and how each can be used as a component of a firewall.

Service-Independent Filters

Some types of Internet attacks are popular, but difficult or impossible to specify using only generic packet-header information. These attacks are generally service-independent and are difficult to specify because filtering rules require additional information that can only be learned by looking in the routing table. For these types of attacks, a separate control (-FireWall CONTrol) has been created for each of the known attack types. The -FireWall CONTrol parameter works at the IP layer and allows you to control the following items:

- Filtering
- Source IP address spoofing
- TCP/IP tiny fragment attacks
- Packets that contain IP options such as source-route, record-route, and time-stamp
- IP-over-IP tunnels
- ICMP messages (protection against denial of service attacks)

For more information on the -FireWall CONTrol parameter, refer to Chapter 24 in *Reference for NETBuilder Family Software*.

Predefined (Service-Dependent) Filters

This section identifies some of the important services commonly used over Internet (Telnet, SMTP, NNTP, FTP, HTTP, Gopher, and DNS) and describes how the parameters in the Firewall Service relate to each service. These parameters are designed for filtering that service; one parameter controls incoming connections and one parameter controls outgoing connections. Each parameter can be separately configured on a per-interface (per-port) basis.

The filtering operations performed by these service-dependent filters can also be performed using generic filters, which are described in "Generic Filters" on page 7-11.

Service-dependent filters, and their related parameters, provide the following benefits:

- The parameters are designed for connection flows instead of packet flows. Connection flows deal with issues such as outbound FTP sessions or inbound Telnet sessions. Packet flows (the basic concept of generic filter rules) deal with inbound TCP packets or outbound UDP packets. A connection flow

(such as inbound Telnet) usually involves bidirectional packet flows (such as both inbound TCP packets and outbound TCP packets).

- These parameters, such as FTPOut, automatically take care of both outbound and inbound TCP packets.
- These parameters allow a user who is not an expert on packet formats, or who may not be aware of the protocol details and port number schemes, to easily configure a packet-filtering router.
- Some services, such as DNS, are quite complex and they are difficult to specify using the generic filter rules.

Be aware that providing packet-filtering rules for a particular service does not mean the service is secured. Each service has its own weaknesses and security holes that are beyond the ability of a packet-filtering router to control. In general, permitting outbound connections is very safe. Outbound connections are connections initiated from internal networks to the Internet; they do not permit Internet-initiated connections into internal networks. Permitting inbound connections are much more risky. Consider using application-level proxy services that will further enhance your firewall.

TCP-based services such as TELnet, FTP, SMTP, NNTP, DNS, Gopher, and Archie are much safer than non-TCP-based services. Avoid using non TCP-based services on your Internet connections. Because of the CPU requirements of DenySrcSpoofing, it is recommended that it be turned on only where absolutely required. Interfaces receiving external traffic should have DenySrcSpoofing turned on to repel source spoofing attacks. Internal interfaces may not require this type of checking.

Dynamic "Window Management" for FTP

FTP is a TCP-based service, and is unusual because it uses two or more simultaneous TCP connections. The first TCP connection is initiated from client to server. This connection, usually called the *command channel*, carries commands and replies. The second TCP connection, usually called the *data channel*, is dedicated to transferring data. The second TCP connection is made in two ways: regular FTP and passive FTP.

Regular FTP occurs when a client issues a PORT command on the command channel to the server and the server opens the data TCP session. The port number of the client's desired data TCP socket is embedded in the PORT command.

Passive FTP occurs when a client issues a PASV command and, if the server responds positively, the client initiates the data TCP session. The TCP port number is embedded in the command and reply.

The FTPIn parameter and the FTPOut parameter understand both forms of FTP sessions. The FTP filters permit server-to-client data TCP connections when they detect the PORT command. They also permit a client-to-server data TCP connection when they detect a PASV command. The TCP port number is extracted from the PORT (or PASV) command so that only a specific data connection is allowed; no persistent holes in the firewall will occur.

An FTP session may involve several data TCP connections, therefore, the FTPIn parameter and FTPOut parameter constantly monitor the active command

channels for PORT and PASV commands and readjust their permissions window accordingly.

When the command channel is closed, all associated data channels are also closed.

Generic Filters

3Com's simple yet powerful filter language allows you to write your own specialized filters, each of which may be comprised of a number of rules within the generic filters. Some examples are:

- Rules can be specified on a per-interface basis.
- Rules can be applied to incoming traffic, outgoing traffic, or both.
- Rules are based on easy-to-understand names and values, instead of hex numbers, offsets, or bit-masks.
- Rules can provide comprehensive logging for both permitted or denied packets.
- Rules can permit or deny packets based on any combination of source address, destination address, protocols, source TCP/UDP port, destination TCP/UDP port, ICMP message types, and TCP "Establish" keyword to differentiate the direction of TCP connections from the value of the SYN bit.

Managing Filters

This section describes the syntax for creating filters, how to create and delete filters, how to manage filters in your firewall configuration, and the differences between traditional IP filters and firewall filters.

Filter Rule Syntax

For detailed information on rule syntax and corresponding values, refer to the Filters parameter in Chapter 24 of *Reference for NETBuilder Family Software*.

Creating Filters Using Filter Rules

Each filter has a name assigned to it. The syntax of a filter name is the same as a DOS filename; it can be up to eight characters followed by up to a three-character extension. File names are case-insensitive. When a new filter is created with the same name as an existing filter, the new filter replaces the old one, in memory and on the disk.

A filter must have at least one rule defined within it. Each rule must begin with one of two keywords: Permit or Deny. There is no limit to how many rules can be defined in a filter; the software continues to accept new rules as long as there is memory or disk space available for them.

Defining a Filter Using the ADD Filter Command

To define a filter use:

```
ADD Filter <filter name> [<rule 1> <rule 2> <more rules> ...]
```

A filter begins with the left parenthesis, and terminates with the right parenthesis. In between the parenthesis, any number of rules are allowed. Empty rules are ignored.

Each rule must be terminated by a new-line character; that is, no more than one rule can occupy a single line of input. Rule syntax checking is performed

time when rules are entered. Any syntax error is identified immediately and triggers the following actions:

- Displays error descriptions about the kind of syntax problem
- Discards the current line of input
- Displays help information
- Prompts for continued input

A rule that begins with the pound (#) sign indicates that it is a comment, and the software ignores the whole line.

Creating Filters Using An Off-line Editor

You can use any PC or workstation text editor to create and edit your filter files. The workstation is also a good place to back-up the filters file. Because filters can be complex, do the following:

- Examine and carefully edit the filters on your workstation
- Add comments to them using the pound (#) sign
- Use TFTP to transfer the final results under the FILTERS directory on the bridge/router.

After the file transfers are complete, you must either reboot the bridge/router or issue the RESTart command to restart the firewall with the newer set of filters.

Use the TEst command to test filters with test packets generated by the bridge/router. The system then reports whether the packet was permitted or denied.

The RESTart command examines the filter file, detects any syntax errors, and provides the line number, the offending keywords, and other applicable help information. If there is a syntax error in the filter file, none of the defined filters will take effect. As a result, you are responsible for making appropriate corrections to the filter file off-line and reentering the RESTart command.

A filter file must contain only filter rules conforming to the syntax specified in Firewall filter parameters in *Reference for NETBuilder Family Software*. Blank lines and comment lines starting with the pound sign (#) are ignored.

Displaying Filters

To display all the filters that are currently defined, enter:

```
SHow -Firewall Filter
```

The display shows the names, sizes, and creation dates for all of the filters that are currently stored on disk.

To display the contents of a particular filter, use:

```
SHow -Firewall Filter <filter name>
```

Deleting Filters

Filters must be individually deleted from the system. Deleting a filter not only removes it from local storage, but also removes it from memory. The effect is immediate, and any interface currently associated with this filter loses its effects.

To delete a filter, use:

```
DELete -Firewall Filter <filter name>
```

Assigning Filters to Interfaces

Each interface (port) can have two filters associated with it: one filter that applies to traffic received on that port, and one that applies to traffic to be transmitted on that port.

The command syntax for InFilter and OutFilter is:

```
SETD !<port> -Firewall InFilter = <filter name>
SETD !<port> -Firewall OutFilter = <filter name>
```

For more information on the InFilter and OutFilter parameters, refer to Chapter 24 in *Reference for NETBuilder Family Software*.

All incoming packets, including broadcast, unicast, and multicast IP packets, received on the <port> are subject to filtering operations as specified in the InFilter parameter. Packets going to the router, as well as packets to be forwarded by the router, are all subject to filtering, including all the incoming routing protocol packets such as OSPF, RIP, or others.

All outgoing packets, including broadcast, unicast, and multicast IP packets, to be transmitted over <port> are subject to the filtering operation as specified in the OutFilter parameter. Packets originating from the router (for example, routing protocol packets), as well as forwarded packets (from another source), are all subject to the same filtering operations.

To remove a filter from the interface, use:

```
SETD !<port> -Firewall InFilter = "" (empty name)
```

If an assigned <filter name> is not found in local storage, no filter operation is activated.

Activating and Deactivating Filters

To enable or disable all firewall filtering on an interface, use:

```
SETD !<port> -FireWall CONTrol = Filter | NoFilter
```

Each interface can be enabled independently of other interfaces. Enabling filtering on an interface enables ALL of the applicable firewall filters including:

- Service-independent filters, such as SourceSpoofing and IPTunnel.
- Generic filters, such as those defined in the InFilter and OutFilter parameters.
- Default actions as defined in the DefAction parameter and predefined filters.

Only packets being forwarded by IP forwarding routine, as well as those destined to or originated from local system, are subject to the filtering action; packets being bridged are not subject to this service. Bridging filters are the appropriate place to define rules for filtering bridged packets.

Firewall Filters versus IP Filters

There are several differences between firewall filters and traditional IP filters used on the NETBuilder II and SuperStack II bridge/routers:

- Traditional IP filters are output filters only. They cannot be configured to perform input filtering, an important first level of defense. Not only is input filtering necessary in blocking certain attacks (such as IP source address spoofing), it also protects the router itself.
- Separate filters cannot be specified on a per-interface basis using traditional IP filters.
- Traditional IP filters contain non user-friendly hex numbers, offsets, and masks. This syntax is complex and confusing for users to design and maintain.
- Traditional IP filters cannot effectively deal with packets containing IP options.
- Traditional IP filters reorder the filtering rules to their own preference. Users may be confused in some situations, and the filter design may be defeated in other situations.

Because of the special role traditional IP filters play, they remain unchanged and operate in parallel with the firewall. Some specific services are only available from the traditional IP filter such as X.25 profile ID, packet priority, protocol reservation, and dial-on-demand discard. For those services, uses must continue to use existing IP filters.

Filters — Firewall Execution Order

When Ip and Firewall filters are enabled, the sequence of execution for through traffic is as follows:

- receive packet -> input firewall filter -> "forwarding decision" -> traditional IP filter -> output firewall filter -> transmit packet

For traffic coming into and terminating at the bridge/router, the sequence is:

- receive packet -> input firewall filter -> "internal process" (no IP filter is applied)

For traffic originating from within the bridge/router, the sequence is:

- "internal process" -> output firewall filter -> transmit packet

Traditional IP filters only apply to "through" traffic. Traffic *to* the router is unfilterable.

Setting Up System Logs

IP Firewall can be configured to log system messages to a log server, to the local console, or both. Log messages contain crucial information such as the date and time, interface, incoming and outgoing, packet header summary, and reason. Each message contains two types of codes: facility and priority. The facility code tells syslog what subsystem the message is from and the priority code tells syslog how important the message is (ranging from Log(0), emergency, which is the highest priority to Log(7), debug, which is the lowest priority). Logging can be done on permitted or denied packets.

You can set up your firewall to log system messages in one of three ways:

- If you want log messages sent to your local console port, enter:

  SETDefault -FireWall Log = Console

- If you want log messages sent to your syslog server, enter:

  SETDefault -FireWall Log = Syslog

  Most UNIX workstations come with syslog support. Consult your workstation manual for more details. If you choose this logging method, you also need to configure the IP address of your syslog server using:

  ```
  SETDefault -AuditLog LogServerAddr = <IP address>
  ```

  For more information on the LogServerAddr parameter, refer to Chapter 10 in the *Reference for NETBuilder Family Software*.

- If you want to send log messages to your local console port and to your syslog server, enter:

  SETDefault -FireWall Log = (Syslog,Console)

Specifying Log Content

You can enable specific information to be logged. For example, to see all of the denied source-spoofing packets, enter:

SETDefault -FireWall Log = SrcSpoofing

If you want to log all incoming FTP connections, add a Log option to your -Firewall FTPIn command. For example:

ADD !2 -FireWall FTPIn Permit Log

You can log the summary of the packet or detailed contents (the first 64 bytes) of the packet using:

```
SETDefault -FireWall Log = SUmmary | DEtail
```

For more information on selectively logging messages, refer to the Log parameter in Chapter 24 of *Reference for NETBuilder Family Software*.

Firewall Terms

The following terms are used in this chapter to describe the firewall feature:

firewall	A router and/or workstation with multiple network interfaces that controls and limits specific protocols, types of traffic within each protocol, types of services, and direction of the flow of information.
secure logging	A method that takes an audit trail of system activity is received from a bastion host and places it in a secure location.
IP source address spoofing	Spoofing uses "forged" source addresses to make an outside packet appear to have come from the inside network so that the firewall allows it to have access to the private network. Spoofing works on the principle that, by default, routers perform route lookups only on the destination IP address in each packet, paying no attention to the incoming interface of the source IP address.

8 IP SECURITY OPTIONS

This chapter describes how to configure your bridge/router to implement IP security options to protect datagrams at specified classification levels under the protection rules of specific authorities. These security features, which comply with RFC 1108, are necessary for any network implementing IP security options; for example, Department of Defense networks.

Your system can use Internet Protocol (IP) security options in an internetwork to:

- Transmit the common security labels from source to destination.
- Ensure that the route taken by the datagram is protected to the level required by all protection authorities indicated in the datagram.

This chapter also describes the use of source IP spoofing as a common type of security violation and provides the Internet Computer Emergency Response Team (CERT) recommendations for preventing this type of network attack.

For conceptual information, refer to "IP Security Terms" on page 8-13.

Configuring IP Security Parameters for End Systems

This section describes how to configure IP security parameters for end system configurations (IP routing is not being used). Parameters related to IP security that may need to be configured depend on the type of networking devices involved and the amount of security required.

The procedure is an example of how to use the IP security option commands and is not a standard configuration procedure. Depending on the type of network security you require, your configuration procedure may differ from the one provided.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths according to Chapter 1.

If you are using your system as an end system (the system receives packets and sends packets through the same interface; packets are not routed through the box to another interface), a configuration example is provided in the next section. If you are using your system as an IP router, a configuration example is provided in "Configuring IP Security Options for IP Routers" on page 8-2.

Before beginning the configuration, make sure your system has the following initial settings for the IP security option parameters:

- SecLEVel is set to UNCLass UNCLass for all ports.
- SecAuthIn is set to GENSER for all ports.

- SecCONTrol is set to NoEXTended for all ports.
- SecLabelSys is set to UNCLass GENSER for all ports.
- SecLabelValues is set to RFC1108.

The following procedure describes how to configure your system to transmit and receive datagrams with a TopSECret classification level, how to accept datagrams with any combination of GENSER and SIOP-ESI protection authorities, and how to attach a TopSECret GENSER label to datagrams originated by the system. The IP address was assigned to the system using the SETDefault !0 -IP NETaddr command.

Procedure To configure the system, follow these steps:

1. To configure all system ports to transmit and receive TopSECret datagrams enter:

   ```
   SETDefault !0 -IP SecLEVel = TopSECret
   ```

2. Specify the protection authorities that can be present in datagrams received on all ports.

   a. Change the default setting by entering:

   ```
   DELete !0 -IP SecAuthIn GENSER
   ```

   b. Set the Security Authorization SIOP-ESI protection by entering:

   ```
   ADD !0 -IP SecAuthIn GENSER SIOP ANY
   ```

3. Configure the classification level and protection authority label for datagrams originated by the system by entering:

   ```
   SETDefault !0 -IP SecLabelSys = TopSECret GENSER
   ```

4. Enable the system to perform security processing of packets received from a file server by entering:

   ```
   SETDefault -IP SecFileServer = Yes
   ```

   The default for the SecFileServer parameter is "No." For the system to communicate with the file server when IP security options are enabled, you must set the SecFileServer parameter to "Yes."

   For more information, refer to the SecFileServer parameter in Chapter 29 in the *NETBuilder Family Bridge/Router Reference Guide*.

5. Enable security options on the system by entering:

   ```
   SETDefault -IP CONTrol = SECurity
   ```

   For information on how to check your configuration, refer to "Verifying IP Security Options" on page 8-8.

Configuring IP Security Options for IP Routers

This section describes how to configure IP security parameters for IP router configurations. Parameters related to IP security that may need to be configured depend on the type of networking devices involved and the amount of security required.

The procedures are an example of how to use the IP security option commands and are not standard configuration procedures. Depending on the type of network security you require, your configuration procedure may differ from the one provided.

Prerequisites Before beginning the configuration, make sure your system has the following initial settings for the IP security option parameters:

- SecLEVel is set to UNCLass UNCLass for all ports.
- SecAuthIn and SecAuthOut are set to GENSER for all ports.
- SecLabelDefault is set to NONE for all ports.
- SecCONTrol is set to NoEXTended, NoBasicFirst, NoLabelAdd, and NoLabelStrip for all ports.
- SecLabelSys is set to UNCLass GENSER for all ports.
- SecLabelValues is set to RFC 1108.

Procedures Figure 8-1 is an example of a typical internetwork in which IP security options are configured. The configuration allows the following communications:

- PCs with the supercomputer
- Workstations with other workstations as well as the supercomputer
- Mainframes with the supercomputer

In Figure 8-1, the devices are performing the following operations:

- Mainframes generate the label "TSEC SIOP SCI."
- Workstations on port 2 and 4 generate the label "SEC GENSER" and require the basic option to be first in the IP header.
- PCs can neither generate nor receive labels. A default label of "CONF GENSER" is generated for them. This label is stripped before a datagram is sent on port 3.
- Supercomputer assigns labels based on host addressing. It needs to generate datagrams with labels "SEC GENSER" when communicating with workstations, "CONF GENSER" for PCs, and "TSEC SIOP SCI" for mainframes.

A description of each port interface is provided in addition to examples of how to set the IP security option parameters.

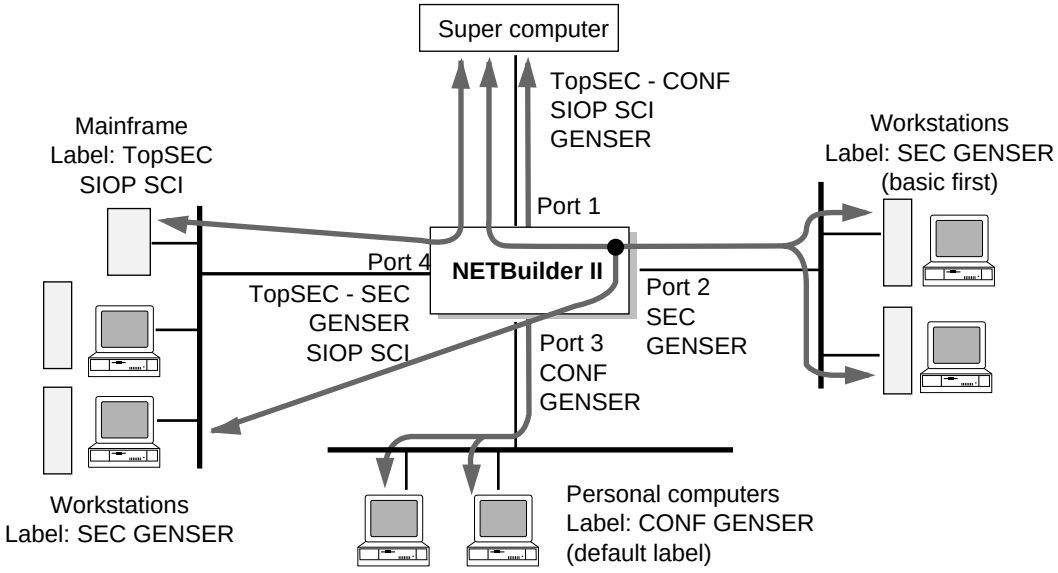


Figure 8-1 IP Security Options Configuration Example

Port 1 Configuration

This procedure shows how to configure port 1 of the system based on Figure 8-1. Port 1 connects to a supercomputer, which assigns labels based on host addressing. You want to configure the system to allow the supercomputer to communicate with the workstations on ports 2 and 4, the PCs, and the mainframes. You need to configure port 1 to transmit to and receive from the supercomputer datagrams with a TopSECret, SECret, or CONFidential classification level and with protection authority flags SIOP-ESI and SCI both set, or just GENSER set.

To configure port 1, follow these steps:

1. Specify the range of security levels of the datagrams that can be transmitted and received on port 1 by entering:

   ```
   SETDefault !1 -IP SecLEVel = CONFidential TopSECret
   ```

   The system can receive or transmit datagrams from port 1 with classification levels of CONFidential, SECret, or TopSECret.

2. Specify the protection authorities that can be present in datagrams received on port 1 by entering:

   ```
   ADD !1 -IP SecAuthIn SIOP SCI
   ```

   The system can receive datagrams from the network on port 1 with SIOP-ESI and SCI set, or just GENSER set. GENSER appears in the SecAuthIn table by default.

3. Specify the protection authorities that can be present in datagrams transmitted on port 1 by entering:

   ```
   ADD !1 -IP SecAuthOut SIOP SCI
   ```

   The system can transmit datagrams to the network on port 1 with SIOP-ESI and SCI set, or just GENSER set.

4. Configure a single classification level and protection authority label for datagrams originated by the system and transmitted on port 1 by entering:

   ```
   SETDefault !1 -IP SecLabelSys = CONFidential GENSER
   ```

   Any datagram generated by the system, including Internet Control Message Protocol (ICMP) messages, have this label when transmitted on port 1.

5. Configure the system so that a label is attached to datagrams before transmission over port 1 by entering:

   ```
   SETDefault !1 -IP SecCONTrol = LabelAdd
   ```

   This parameter is configured because PCs cannot generate labels. The system must be configured to attach a label to a datagram destined for the supercomputer. The label that is attached to the datagram before transmission to the supercomputer is based on the value of the SecLabelDefault parameter. This parameter is set on port 3 of the system.

Port 2 Configuration

This procedure shows how to configure port 2 of the system based on Figure 8-1. Port 2 connects to workstations, which require the basic security option to be the first option in the IP header. You want to configure the system to allow the workstations to communicate with the supercomputer and the workstations on port 4. You need to configure port 2 to transmit to and receive

from the workstations datagrams with a SECret classification level and with the GENSER protection authority flag. You also need to configure the port so that the basic security option is the first option in the IP header of datagrams transmitted on this port.

To configure port 2, follow these steps:

1. Specify the security level of datagrams that can be transmitted and received on port 2 by entering:

   ```
   SETDefault !2 -IP SecLEVel = SECret
   ```

   The system can receive or transmit datagrams from port 2 with classification level of SECret.

2. Configure port 2 so that the basic security option is the first option in the IP header of datagrams transmitted on port 2 by entering:

   ```
   SETDefault !2 -IP SecCONTrol = BasicFirst
   ```

   For datagrams transmitted on port 2, the workstations require that the basic security option is the first option in the IP header.

3. Configure a single classification level and protection authority label for datagrams originated by the system and transmitted on port 2 by entering:

   ```
   SETDefault !2 -IP SecLabelSys = SECret GENSER
   ```

   Any datagram generated by the system, including ICMP messages, have this label when transmitted over port 2.

Port 3 Configuration

This procedure shows how to configure port 3 of the system based on Figure 8-1. Port 3 connects to PCs. You want to configure the system to allow the PCs to communicate only with the supercomputer. You need to configure port 3 to transmit and receive datagrams with a CONFidential classification level and with GENSER protection authority flag. Because the PCs can neither generate nor accept a security label, the system must attach a default label (CONFidential GENSER) to datagrams received on port 3 and destined for the supercomputer, and strip the label from datagrams destined for the PCs.

To configure port 3, follow these steps:

1. Specify the security level of datagrams that can be transmitted and received on port 3 by entering:

   ```
   SETDefault !3 -IP SecLEVel = CONFidential
   ```

   The system can receive or transmit datagrams by this port with classification level of CONFidential.

2. Configure port 3 to strip the security label from datagrams transmitted to the PCs by entering:

   ```
   SETDefault !3 -IP SecCONTrol = LabelStrip
   ```

   Because PCs cannot receive labels in datagrams, the system must strip the label before transmission on port 3.

3. Configure port 3 to attach a default label to datagrams received from PCs by entering:

   ```
   SETDefault !3 -IP SecLabelDefault = CONFidential GENSER
   ```

PCs cannot generate or transmit labels; therefore, the system must attach a default label of CONFidential and GENSER. The datagram can then be properly routed to port 1 and the supercomputer.

4 Configure port 3 so that datagrams originated by the system and transmitted on port 3 do not have labels by entering:

```
SETDefault !3 -IP SecLabelSys = NONE
```

Because PCs cannot receive labels, the SecLabelSys parameter needs to be set to NONE.

Port 4 Configuration

This procedure shows how to configure port 4 of the system based on Figure 8-1. Port 4 connects to mainframes and workstations. You want to configure the system to allow the mainframes to communicate only with the supercomputer, and the workstations to communicate both with the supercomputer and the workstations on port 2. You need to configure port 4 to transmit and receive datagrams with a TopSECret or SECret classification level, and with SIOP-ESI and SCI protection authorities set or GENSER set.

To configure port 4, follow these steps:

1 Specify the security level of datagrams that can be transmitted and received on port 4 by entering:

```
SETDefault !4 -IP SecLEVel = SECret TopSECret
```

The system can receive or transmit datagrams on this port with classification levels of SECret and TopSECret.

2 Specify the protection authorities that can be present in datagrams received on port 4 by entering:

```
ADD !4 -IP SecAuthIn SIOP SCI
```

The system can receive datagrams from the network on port 4 with SIOP-ESI and SCI set, or just GENSER. GENSER appears in the SecAuthIn table by default.

3 Specify the protection authorities that can be present in datagrams transmitted on port 4 by entering:

```
ADD !4 -IP SecAuthOut SIOP SCI
```

To transmit datagrams to the mainframes and workstations connected to port 4, datagrams must have the SIOP-ESI and SCI protection authority flags set in the security label of the datagram. Using this protection authority, the system can receive datagrams from the supercomputer and route them to the mainframes and workstations on port 4. With the GENSER authority (the default), the system can transmit datagrams between the supercomputer and workstations on ports 2 and 4.

4 Configure a single classification level and protection authority label for datagrams originated by the system and transmitted on port 4 by entering:

```
SETDefault !4 -IP SecLabelSys = SECret GENSER
```

Any datagram generated by the system, including ICMP messages, have this label when transmitted on port 4.

Enabling IP Security Processing

To enable the IP router security options, follow these steps:

1. Enable security options on the system by entering:

   ```
   SETDefault -IP CONTrol = SECurity
   ```

   After enabling security options, change the default of the SecFileServer parameter to Yes to ensure proper communication with the file server. For more information, refer to the SecFileServer parameter in Chapter 29 in *Reference for NETBuilder Family Software*.

2. Display the configuration settings by entering:

   ```
   SHow -IP CONFiguration
   ```

   For information on how to check your configuration, refer to "Verifying IP Security Options."

Configuring Extended Security Option Labels

For environments requiring extra security measures, you can add extended security labels to IP packets leaving specific ports. With this option, you can add a string to outgoing IP packets so that only specific hosts can accept the packets. The extended security label options can be used with the normal IP security options described earlier in this chapter, or they can be used independently.

The extended security label option is not required for the majority of configurations. If you use this option, be careful to configure it correctly to obtain the desired effect.

To support this configuration, follow these steps:

1. If you have basic IP security control enabled, make sure the SecCONTrol parameter is set to EXTended and NoBasicFirst for port 2 by entering:

   ```
   SETDefault !2 -IP SecCONTrol = (EXTended, NoBasicFirst)
   ```

   If you do not specify these values, the extended labels will be discarded.

2. Specify the extended security label to be added to packets using:

   ```
   SETDefault -IP SecLabelXtra = "<string>"
   ```

   Specify the string as values of individual bytes given as a decimal number, with each byte being separated by a slash (/). The total number of bytes must be a multiple of four, and the string must end with a slash. No syntax checking is performed, therefore, the string must be specified correctly.

3. Configure port 2 to add the label specified by the SecLabelXtra parameter to all packets sent over this port by entering:

   ```
   SETDefault !2 -IP SecCONTrol = LabelXtraAdd
   ```

4. Configure port 1 to strip the extended label for all packets sent over this port by entering:

   ```
   SETDefault !1 -IP SecCONTrol = LabelExtStrip
   ```

   For more information about the parameters described in this procedure, refer to Chapter 29 in *Reference for NETBuilder Family Software*.

Verifying IP Security Options

To check your configuration before using the IP security settings on your system, follow these steps:

1. Verify the IP security configuration by entering:

   ```
   SHow -IP CONFiguration
   ```

   The system displays all IP configuration settings. If any of the IP security settings are incorrect, you can reconfigure them.

2. Check the IP security configuration by entering:

   ```
   SecCheck
   ```

   Because no port number is specified, the system checks all port configurations. If you specify a port number with this command, the system checks the configuration for that port.

   This command is a diagnostic tool and does not check for all possible configurations. If the system finds misconfigurations, warning messages are displayed.

ICMP Error Messages

The following ICMP error messages may be generated as a result of security processing:

- ICMP Parameter Problem Missing Option, Type 12 Code 1, Pointer =130

  No security options in packet, but security options required on the port.

- ICMP Parameter Problem Missing Option, Type 12 Code 1, Pointer = malformed option

  Malformed security option, for example, an invalid label.

- ICMP Destination Unreachable Communication Administratively Prohibited, Type 3

  Code 9 for the network, Code 10 for the host. Security label out of range.

These ICMP error messages may be generated because of an incorrect configuration or may indicate an unauthorized break into a secure network. The network manager can decipher ICMP message codes by using a network analyzer.

To display a count of ICMP error messages, enter:

```
SHow -SYS STATistics -IP
```

For a sample display and explanation of the entries, refer to Appendix H.

Preventing Security Attacks on IP Routers

Source IP spoofing is a common type of security violation. The Internet CERT has summarized the danger of how IP spoofing is used in network attacks. This section describes how to configure 3Com bridge/router software to provide security against this type of attack.

How IP Spoofing Works

To generate an attack, the intruder creates packets with spoofed source IP addresses. In this type of attack, the intruder transmits packets from outside the "protected" domain that claim to be from a trusted machine inside the "protected" domain (for example, the packet contains the source IP address of

a trusted machine). If the router is not configured to filter incoming packets whose source address is in the local domain, it forwards the traffic and the targeted system may become compromised. A router generally forwards this traffic because it only examines the destination IP address when it makes its forwarding decision, not the source IP address. Figure 8-2 illustrates the operation of a spoofed source IP address attack.

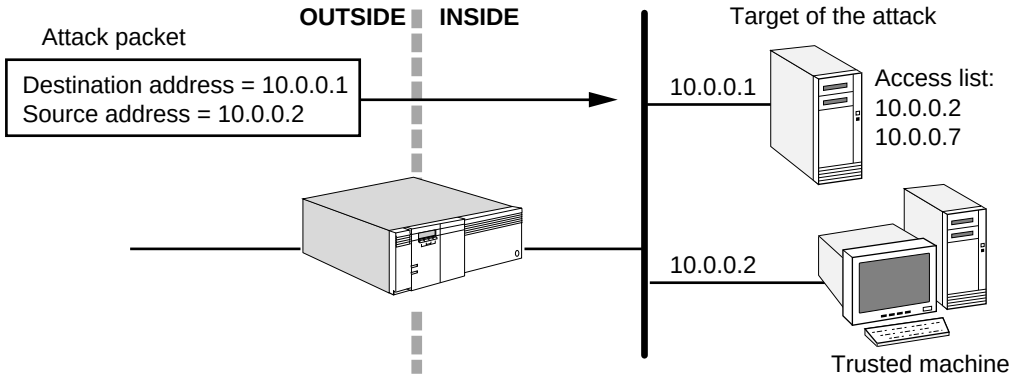


Figure 8-2 Spoofed Source IP Address Attack

Attacks are aimed at applications that use authentication based on source IP addresses. If successful, an attack leads to unauthorized user and possibly root access on the targeted system. It is important to note that the described attack is possible even if no reply packets can reach the attacker. Also, disabling source routing at the router does not provide protection from this type of attack.

Examples of configurations that are potentially vulnerable to attack include:

- Routers to external networks that support multiple internal interfaces
- Routers with two interfaces that support subnetting on the internal network
- Proxy firewalls where the proxy applications use the source IP address for authentication

Hijacking Tool

After intruders have achieved root access on a system, they use a tool to dynamically modify the UNIX kernel. This modification allows them to hijack existing terminal and logon connections from any user on the system. In taking over existing connections, intruders can bypass one-time passwords and other strong authentication schemes by tapping the connection after the authentication is complete. For example, a legitimate user may connect to a remote site through a logon or terminal session. An intruder can hijack the connection after the user has completed authentication to the remote location. The site would now be compromised. Currently, the hijacking tool is used primarily on SunOS 4.1.x systems. However, system features that make this attack possible are not unique to SunOS.

Preventing Attacks

To prevent this type of attack, the CERT Coordination Center recommends that network security personnel follow these steps:

1. Install a filtering router that restricts the input to the external interface (known as an input filter) by not allowing a packet through if it has a source address from the internal network.

2 Filter outgoing packets that have a source address different from the internal network to prevent an attack originating from the local site. Figure 8-3 illustrates the CERT recommendations.

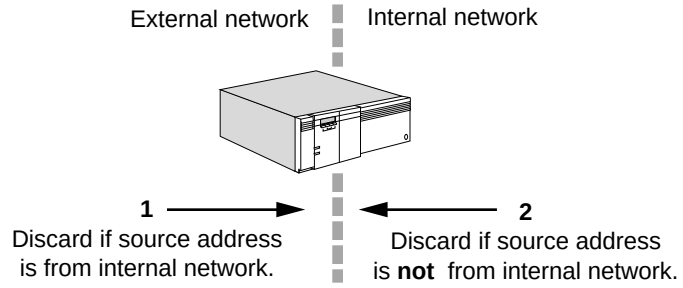


Figure 8-3 CERT Recommended Filters

CERT recommends an alternative solution if a router does not support filtering on the in-bound side. The spoofed IP packets may be filtered by installing a second router between the original external interface (a) and the outside connection (b).This router can then be configured to block all packets that have a source address in the internal network on the outgoing interface (c) connected to the original router. Figure 8-4 illustrates the alternative CERT recommendation.

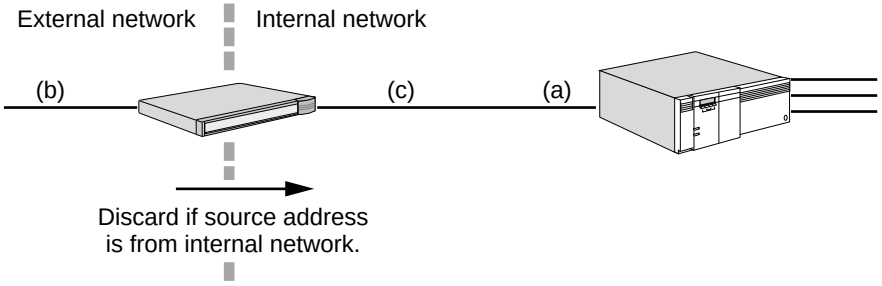


Figure 8-4 Alternative CERT Configuration

Secure Configuration Solutions

The following examples illustrate how bridge/router software can be configured to support the CERT Advisory recommendations. Each of these examples assumes that the value of the -IP FilterDefAction parameter is configured to Forward. However, none of these examples prevent a source IP spoofing attack originating from the local site.

Noncontiguous IP Networks

The example in Figure 8-5 illustrates a two-router solution where the internal network is configured with noncontiguous IP network numbers. The filters are installed on the border router, which can only have two interfaces. In a two-port router, an output filter on one port is equivalent to an input filter on the other port.

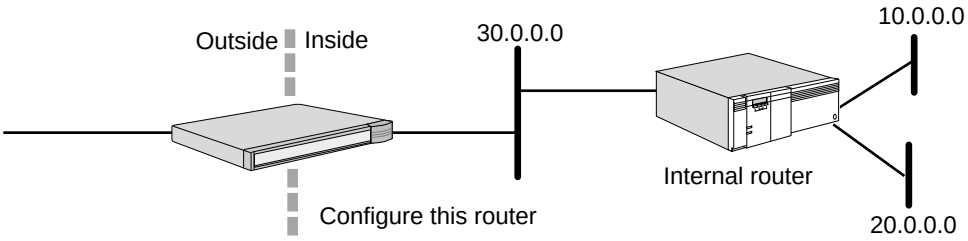


Figure 8-5 Noncontiguous IP Networks

Add the following filters to the border router to prevent an external attack:

```
ADD -IP FilterAddrs 10.0.0.0/0.255.255.255 > 10.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 20.0.0.0/0.255.255.255 > 20.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 30.0.0.0/0.255.255.255 > 30.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 10.0.0.0/0.255.255.255 <> 20.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 10.0.0.0/0.255.255.255 <> 30.0.0.0/0.255.255.255 Discard
ADD -IP FilterAddrs 20.0.0.0/0.255.255.255 <> 30.0.0.0/0.255.255.255 Discard
```

This configuration prevents the external attack and allows the internal router to route traffic between networks 10.0.0.0, 20.0.0.0, and 30.0.0.0. This configuration also works for the cascade topology shown in Figure 8-6.

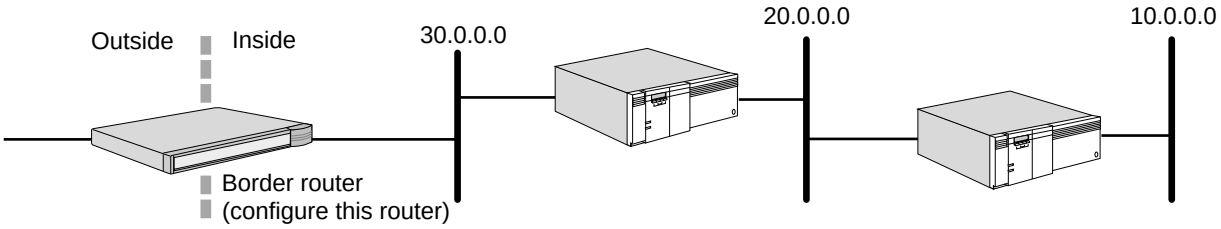


Figure 8-6 Noncontiguous IP Networks (Alternative Topology)

Subnets on the Internal Network

The example in Figure 8-7 illustrates a two-router solution when the internal network is configured with multiple subnets of the Class B network address,130.5.0.0.

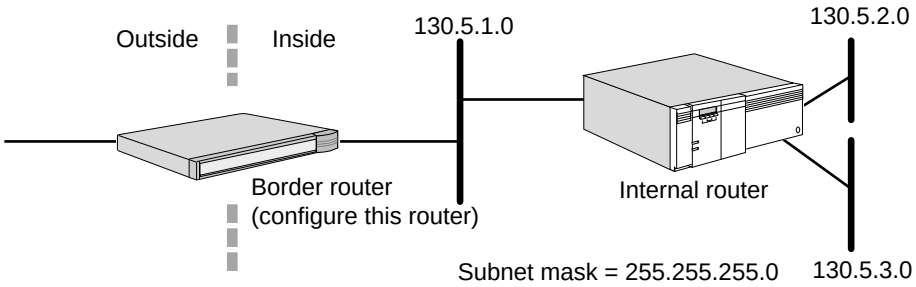


Figure 8-7 Subnets on the Internal Network

Add the following filter to the border router to prevent an external attack:

```
ADD -IP FilterAddrs 130.5.0.0/0.0.255.255 > 130.5.0.0/0.0.255.255 Discard
```

This configuration prevents the external attack and allows the internal router to route traffic between all subnetworks of 130.5.0.0. In this example, a single filter can protect multiple subnets.

Multiple Contiguous IP Networks

The example in Figure 8-8 illustrates a two-router solution where the internal network is configured with contiguous IP network numbers. Assume the service provider has provided the subscriber with the Classless Interdomain Routing (CIDR) Protocol block 200.5.0.0/255.255.0.0.

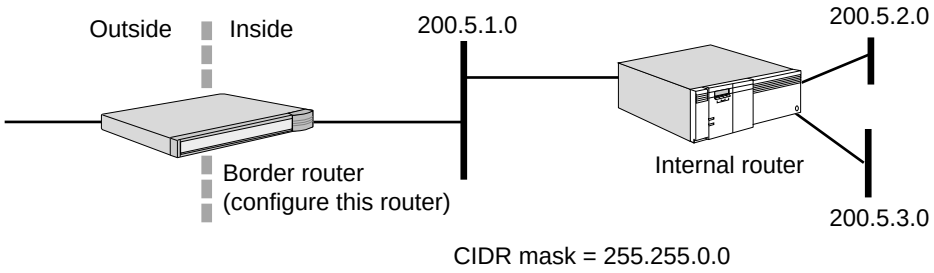


Figure 8-8 Multiple Contiguous IP Networks

Add the following filter to the border router to prevent an external attack:

```
ADD -IP FilterAddrs 200.5.0.0/0.0.255.255 > 200.5.0.0/0.0.255.255 Discard
```

This configuration prevents the external attack and allows the internal router to route traffic between supernets of 200.5.0.0/255.255.0.0. In this example, a single filter can protect multiple contiguous IP networks numbers assigned as a CIDR block.

Alternative Two-Router Configurations

Various 3Com bridge/routers can be configured for security. The external router can be a model 227 or 228 SuperStack II NETBuilder bridge/router while the inside router can be another 3Com router. In some cases, routers from two different vendors may be optimal because a bug or back door that allows entry by a hacker in one vendor's code may not exist in the other vendor's code.

In many cases, the network topology can have the following characteristics:

- An external link to the Internet, which is a simple serial link to the network provider's router.
- The inside network consists of a few noncontiguous networks or subnets of a single network number.

Figure 8-9 illustrates this common configuration. The external router is configured with the required filters. The external router is also configured with a default route pointing to the Internet. The service provider installs static routes in their router that point to the customer's network. For this configuration, it is not necessary to run a routing protocol over the external link. If the network connectivity is more complex and you are connected using a multipoint technology such as X.25 or Frame Relay, you can run the Border Gateway Protocol version 4 (BGP-4) on a model 227 or 228 SuperStack II NETBuilder bridge/router to provide the required connectivity.

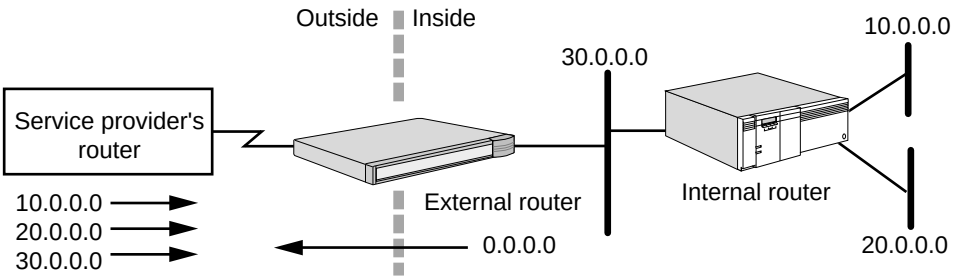


Figure 8-9 Two-Router Configuration

Firewall Configurations

Many firewall configurations require the use of two routers. A typical Internet firewall using two routers is illustrated in Figure 8-10.

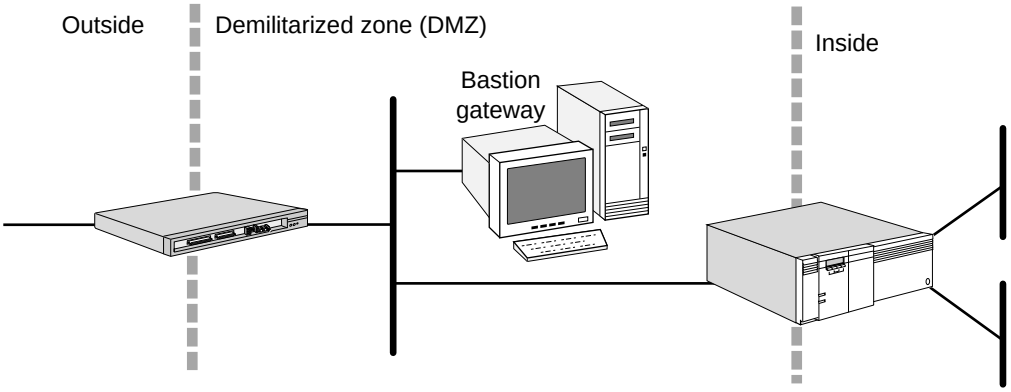


Figure 8-10 Internet Firewall

In this example, the routers create a packet filtering firewall while the bastion gateway functions as an application gateway firewall. In addition to using routers, creating a secure Internet firewall requires packet filtering and applications gateways. For information about filtering, refer to Chapter 4.

IP Security Terms

The following terms used in this chapter explain IP security concepts:

basic security options	Identifies the U.S. classification level at which the datagram is to be protected and the authorities whose protection rules apply to each datagram.
classification level	Specifies the U.S. classification level (top secret, secret, confidential, and unclassified) at which the datagram must be protected.
Classless Interdomain Routing (CIDR)	A method of using IP addresses without regard to traditional address classes to help solve the problem of the lack of class B network numbers.
extended security options	Permits additional security labeling information beyond what is presented in the basic security option to meet the needs of additional registered authorities.
label	Refers to the classification level and protection authority characteristics of a datagram.

protection authority	Identifies the agency that specifies the protection rules for transmission and processing of information contained in the datagram. Examples of protection authorities include the following: GENSER: the point of contact for this authority is the Designated Approving Authority per Department of Defense (DOD) 5200.29. SIOP-ESI: The point of contact for this authority is the Department of Defense, Organization of the Joint Chiefs of Staff. SCI: The point of contact for this authority is the Director of Central Intelligence. NSA: The point of contact for this authority is the National Security Agency. DOE: The point of contact for this authority is the Department of Energy.
source IP spoofing	A common type of security violation in which an intruder accesses a protected domain by using the IP address of a trusted machine.

9 CONFIGURING IP MULTICAST ROUTING

This chapter describes the procedures for configuring your system to perform Internet Protocol (IP) multicast routing. It describes how the multicast router works and gives guidelines for operating, managing, and troubleshooting it.

For conceptual information, refer to "How the IP Multicast Router Works" on page 9-23.

Configuring a Basic Multicast Router

The procedure in this section describes the minimum number of steps required to configure your system for IP multicasting. Depending on your network requirements, you can use the default values of the parameters in the various services, or you can further configure the router according to later sections in this chapter.

To configure the IP multicast router, you must set parameters in the MIP Service. You must also set parameters in the DVMRP Service if your network uses the Distance Vector Multicast Routing Protocol (DVMRP), or in the MOSPF Service if your network uses the Multicast Open Shortest Path First (MOSPF) routing protocol.

Configuring for Local Area Networks and Point-to-Point Links

Use this procedure to configure basic IP multicast routing over LAN ports and Point-to-Point Protocol (PPP) links. You can enable both DVMRP and MOSPF if you want to perform route exchanges between the two domains.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Configure your ports and paths as described in Chapter 1.
- If you are planning to use MOSPF as the multicast routing protocol, you must also set up OSPF for IP unicast routing as described in Chapter 6.
- Become familiar with the protocols supported by the router. This chapter describes the protocols only when the explanation is necessary for interpreting the parameters and screen displays used in the router software.

Procedure

To set up a basic configuration, see Figure 9-1 and follow these steps on the multicast router:

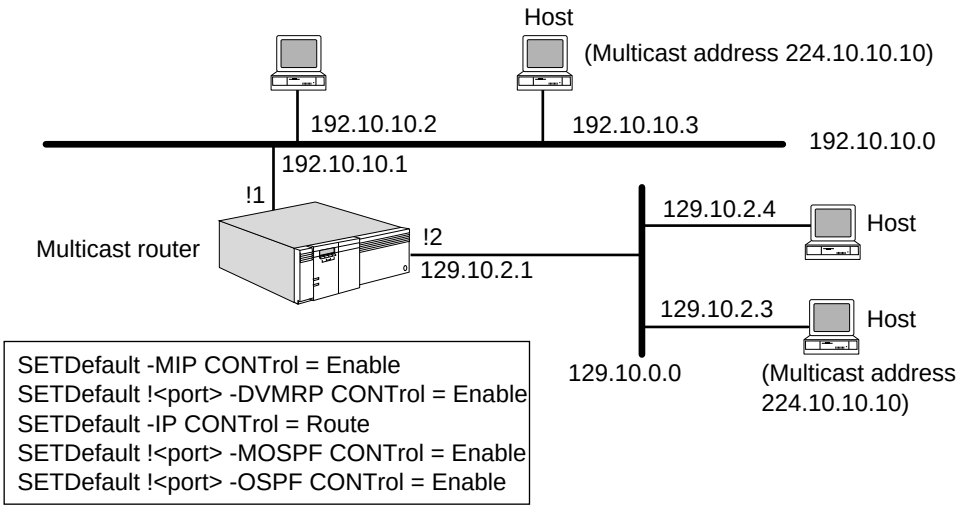


Figure 9-1 Configuring Multicast Routing

1 Assign an IP address to each router port that will perform IP multicasting using:

```
SETDefault !<port> -IP NETaddr = <IP address> [<subnet mask> [Ones
   | Zeros [MTU]]] | UnNumbered
```

Assign the IP addresses to LAN ports and to WAN ports using PPP as the serial line protocol. PPP does not require that you assign an IP address to each wide area port. If you do not want to assign an IP address to a wide area port, you must set the value of the -IP NETaddr parameter to UnNumbered. An advantage of not assigning an IP address to each wide area port is that you conserve valuable network and subnet numbers.

For example, to assign an IP address to port 1 and port 2 of the multicast router in Figure 9-1, enter:

```
SETDefault !1 -IP NETaddr = 192.10.10.1 255.255.255.0
SETDefault !2 -IP NETaddr = 129.10.2.1 255.255.0.0
```

2 Enable the MIP Service by entering:

```
SETDefault -MIP CONTrol = Enable
```

3 Determine which multicast routing protocol you want to use.

- Use DVMRP if you want to attach to the Internet's Multicast backBONE (MBONE), or if you are not using OSPF as the routing protocol. Complete step a.
- Use MOSPF if you are already running OSPF on your LAN and want multicasting support within an autonomous system. Complete step b.
- If you are connecting to the MBONE, you may also want to use DVMRP. Complete steps a and b.

a To enable DVMRP, on the each interface using multicast routing, use:

```
SETDefault !<port> -DVMRP CONTrol = Enable
```

b To enable MOSPF, on the each interface participating in multicast routing, use:

```
SETDefault !<port> -MOSPF CONTrol = Enable
```

The MOSPF Protocol depends on the OSPF Protocol for proper operation. In order to use MOSPF, you must first ensure OSPF is operating correctly. For more information how to enable OSPF, refer to Chapter 6.

You may need to set additional parameters to complete the configuration for PPP. For more information, refer to Chapter 34.

Configuring for Wide Area Networks

To configure multicast routing using DVMRP over Frame Relay or X.25, refer to "Configuring DVMRP Multicasting over Frame Relay" on page 9-10 or "Configuring DVMRP Multicasting over X.25" on page 9-11.

To configure multicast routing using DVMRP or MOSPF over SMDS, refer to "Configuring Multicasting over SMDS" on page 9-7.

Verifying the Configuration

This section explains how to verify the status of networks that are reachable from the multicast routers and to get statistics from the router.

Checking the Overall Status

To check the overall status of your configuration, follow these steps:

1 Display the settings of the parameters in the MIP, DVMRP, and MOSPF Services using:

```
SHow [!<port>] -MIP CONFiguration
SHow [!<port>] -DVMRP CONFiguration
SHow [!<port>] -MOSPF CONFiguration
```

Verify that the MIP, DVMRP, or MOSPF Services are enabled.

2 If you are using DVMRP as the multicast routing protocol, verify the entries in the routing table, forwarding table, and the neighboring router table.

a To display routing and forwarding table entries, use:

```
SHow -DVMRP RouteTable [<subnet>[/<mask>]] [Long]
SHow -DVMRP ForwardTable [<subnet>[/<mask>]] [<group>]
```

In the routing table, check each source subnet and verify that the status is Up.

In the forwarding table, for multicast datagrams that have been sent, make sure there are entries that correspond to the source (source subnet) and destination (multicast group).

For additional information about the displays, refer to "Controlling the Routing Table" on page 9-17 and "Controlling the Forwarding Table" on page 9-18.

b To display neighboring router information, enter:

```
SHow -DVMRP NeighborRouter
```

Verify that the addresses of neighboring routers appear in the list. For more information about this display, refer to "NeighborRouter" on page 20-7 in *Reference for NETBuilder Family Software*.

3 If you are using MOSPF as the multicast routing protocol, verify the entries in the forwarding table by entering:

```
SHow -MOSPF ForwardTable
```

For multicast datagrams that have been sent, make sure there are entries in the forwarding table that correspond to the source and destination.

The MOSPF forwarding table is built only when the router attempts to forward IP multicast packets. The table shows packets the router has recently processed including those successfully forwarded or discarded. The forwarding table varies from router to router because not all routers have forwarded multicast packets. Routers may periodically flush the forwarding table when topology changes are made. For additional information about the display, refer to "Displaying the Forwarding Table" on page 9-22.

4 Examine the local group membership table using:

```
SHow [!<port> | !*] -MIP LocalGroups [<Group addr>]
```

Verify that group memberships of local hosts are displayed in the table.

If you are running MOSPF, only the designated router (DR) and backup designated router (BDR) collect local group information; the table may be empty for non-DR and BDRs.

Getting Statistics

To view statistics, enter:

```
SHow -SYS STATistics -MIP
SHow -SYS STATistics -DVMRP
SHow -SYS STATistics -MOSPF
```

Statistics for DVMRP and MOSPF reflect data that is forwarded, not control messages. Some control statistics are provided in the MIP Service statistics.

You can collect statistics for a specific period by using the SampleTime and STATistics parameters. For more information on these parameters, refer to Chapter 58 in *Reference for NETBuilder Family Software*. For information on interpreting the statistics, refer to Appendix H.

Troubleshooting the DVMRP Configuration

If you are unable to make connections to other networks after setting up the router, review the following troubleshooting procedure. This procedure can help correct problems in making connections involving more than one multicast router. If the router continues to operate improperly after completing this procedure, contact your network supplier for assistance.

You can find neighboring multicast-capable routers and retrieve status using:

```
MRInfo <target IP> [!<port>] [<timeout (0–120)>]
```

For example, if you are unable to connect to network 128.60.0.0 as shown in Figure 9-2, on multicast router 4, enter:

```
MRInfo 128.50.0.1
```

The MRInfo command sends an AskNeighbors packet to request neighboring router information. The display provides the addresses of the neighbor, the neighbor's neighbors, the metric, threshold, and status. The status indicates whether the link is down or disabled. It also indicates whether a tunnel link is down, which would prevent multicast router 4 from sending packets to multicast router 1 or to the host on network 128.60.0.0 that listens to multicast address 224.10.10.10.

If no reply packet is received from 128.50.0.1, enter the MRInfo 128.40.0.1 command to see if the tunnel between multicast router 1 and multicast router 2 is up.

You can also discover the multicast tree from a specified receiver to the source using:

```
MTraceRoute <source> <destination> [G <group>] [H <reports>]
   [!<port>] [T <timeout>] [W <gateway>] [R <Resp addr>]
   [L <Resp ttl>]
```

For example in Figure 9-2, if you want to see the multicast tree from the Host on network 128.60.0.0 to multicast router 4, on multicast router 4, enter:

```
MTraceRoute 128.60.0.1 128.20.0.1
```

Each router in the tree that receives a multicast trace route packet adds its forwarding information associated with the request to the request packet and forwards the packet to the upstream router. When the request packet reaches multicast router 1, multicast router 1 sends a multicast trace route response back to multicast router 4 because the source address is on one of its subnets. The display shows the route from the source to the destination, including hop count, IP subnet, the multicast routing protocol used, the threshold, delay time, and error flags.

For more information about the MRInfo and MTraceRoute commands, refer to Chapter 1 in *Reference for NETBuilder Family Software*.

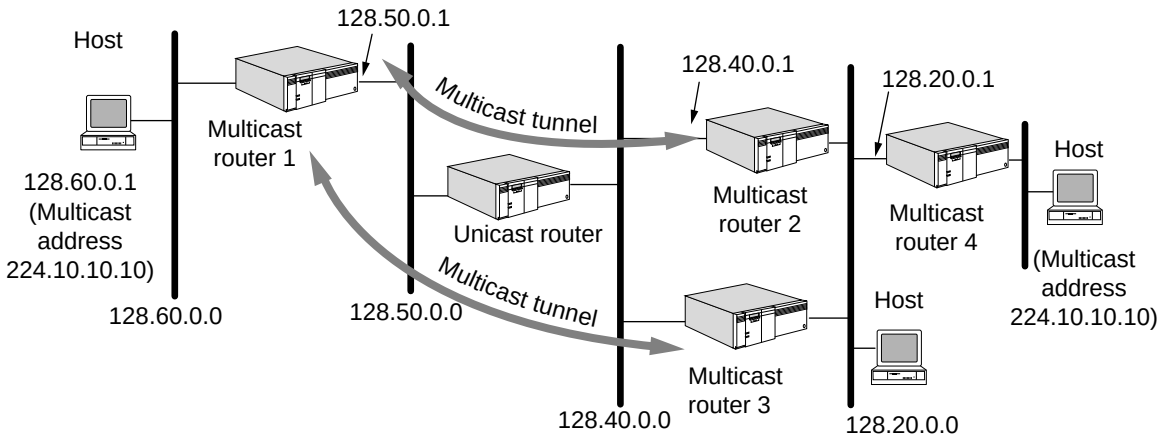


Figure 9-2 Troubleshooting Multicast Router Topologies

Customizing the Multicast Router

After you set up and check the configuration of the basic multicast router, the router begins multicast packet routing among group members. If desired, you can further customize your multicast router as follows:

- Control local group membership queries.
- Adjust the threshold on multicast datagrams.
- Configure multicast routing using DVMRP or MOSPF over SMDS.
- Configure using the DVMRP Protocol.

- Configure a multicast tunnel for DVMRP routers separated by a nonmulticast router.
- Configure scoping (filtering) to prevent traffic from being forwarded beyond a boundary router to a set of addresses.
- Configure multicasting over Frame Relay and X.25.
- Configure a metric.
- Control the bandwidth (rate limit) allocated for multicast datagram traffic.
- Configure routing policies.
- Configure forwarding policies.
- Configure route aggregation.
- Control the DVMRP routing and forwarding tables.

- Configure using the MOSPF Protocol.
  - Configure interarea multicast routing.
  - Configure interautonomous (AS) multicast routing.
  - Configure forwarding policies.
  - Display the forwarding table.

Controlling Local Group Membership Queries

You can control how often Internet Group Management Protocol (IGMP) query messages are sent by the designated router to request local group membership information. For DVMRP routers, the designated router is the router with the lowest IP address. For MOSPF routers, the designated router is the OSPF designated router.

To control local group membership queries, use:

```
SETDefault !<port> -MIP QueryInterval = <seconds>(5–5400)
```

The default setting of this parameter is 120 seconds.

Adjusting the setting of the QueryInterval parameter affects the MembershipExpirationTime, the length of time a local group membership is valid without confirmation. The MembershipExpirationTime is set to two times the value of the QueryInterval parameter plus 20 seconds.

By adjusting the QueryInterval parameter, you control how long entries remain in the local group membership table.

Adjusting the Multicast Datagram Threshold

You can adjust the threshold on the router to prevent multicast packets whose time-to-live (TTL) value is less than threshold from being forwarded to the given interface. By adjusting the default value, you can provide scope control and prevent certain multicast datagrams from being forwarded out of your network.

To adjust this threshold value, use:

```
SETDefault {!<port> | !<tunnel ID>} -MIP THreshold = <value> (1–255)
```

By default, this value is set to 1.

For example, on multicast router B, enter:

```
SETDefault !3 -MIP THreshold = 191
```

Suppose the host in China sends a packet to multicast address 224.10.13.2 with an IP TTL of 192 as shown in Figure 9-3. As the packet is forwarded by each multicast router in its path, the IP TTL value is decremented. When the packet reaches multicast router B, the IP TTL value is 190. The packet will not be forwarded to multicast router A because packets with a TTL value less than the configured threshold of 191 are prevented from reaching the United States.

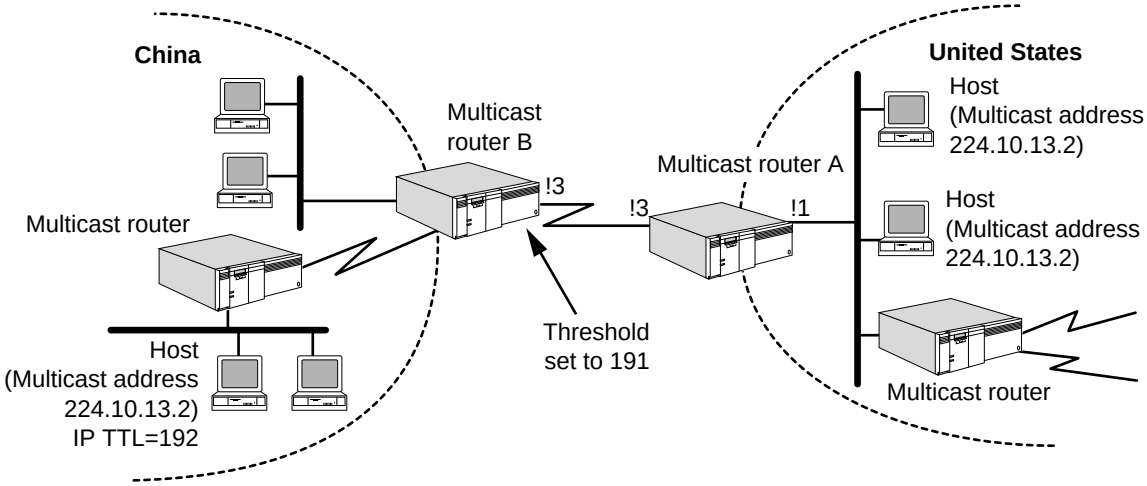


Figure 9-3 Configuring the Threshold

Configuring Multicasting over SMDS

To configure DVMRP or MOSPF multicasting over SMDS, see Figure 9-4 and follow these steps on both ends of the link:

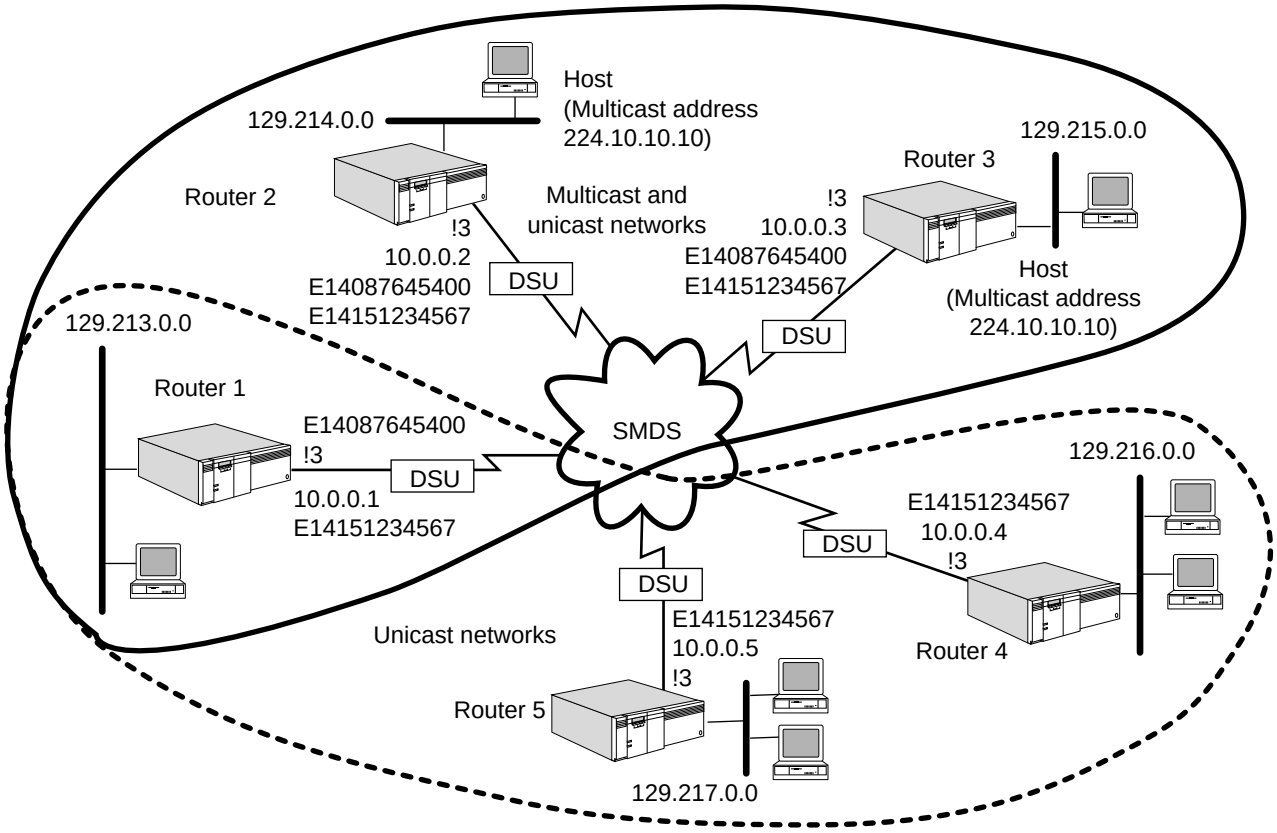


Figure 9-4 Multicasting over SMDS

1. Set up the SMDS Service as described in "Setting Up the SMDS Service" on page 44-1.
2. Assign an IP address to each router wide area port connected to the SMDS cloud that will perform IP multicasting using:

   ```
   SETDefault !<port> -IP NETaddr = <IP address> [<subnet mask> [Ones
      | Zeros [MTU]]]
   ```

   For example, on Router 1, enter:

   ```
   SETDefault !3 -IP NETaddr = 10.0.0.1 255.0.0.0
   ```

3. Specify the IP-to-SMDS group address mapping information per subnet.

   SMDS group addresses are provided when you subscribe to the SMDS network. Your SMDS service provider may omit the E in the address when reporting it to you, but you must include the E when configuring the bridge/router. The digit that follows the letter E is the country code. This example uses digit 1, the country code for the United States. Use the country code for your own country reported by your SMDS service provider.

   If you want to separate unicast and multicast packets, perform steps a and b; otherwise, perform only step a.

   a. Configure the IP Protocol to route packets using:

      ```
      ADD -IP SMDSGroupAddr <IP address> $<E0-E999999999999999>
      ```

      For example, on routers 1, 2, 3, 4, and 5 in Figure 9-4, enter:

      ```
      ADD -IP SMDSGroupAddr 10.0.0.0 $E14151234567
      ```

   b. Configure the MIP Service to separate unicast and multicast packets using:

      ```
      ADD -MIP SMDSGroupAddr <IP addr> $<E0-E999999999999999>
      ```

      For example, on routers 1, 2, and 3, enter:

      ```
      ADD -MIP SMDSGroupAddr 10.0.0.0 $E14087645400
      ```

      Routers 1, 2, and 3 form a network of unicast- and multicast-capable routers that can communicate using a different SMDS address from the one used by routers 4 and 5.

4. Enable DVMRP on the wide area port using:

   ```
   SETDefault !<port> -DVMRP CONTrol = Enable
   ```

   Enable MOSPF on the wide area port using:

   ```
   SETDefault !<port> -MOSPF CONTrol = Enable
   ```

Using the DVMRP Protocol

This section describes how to further customize your multicast router if you are using the DVMRP Protocol as the multicast routing protocol.

Configuring a DVMRP Multicast Tunnel

When two DVMRP routers are separated by a nonmulticast router, a multicast tunnel can be used to build a virtual link between the two routers. Packets to be tunneled over the virtual link are encapsulated by IP-over-IP (IP protocol number is set to 4). No group is associated with a tunnel, and the only neighbor on a tunnel is the remote-end router (nonmulticast routers are not considered to be neighbors).

To configure a multicast tunnel, refer to Figure 9-5 and follow these steps on both ends of the tunnel:

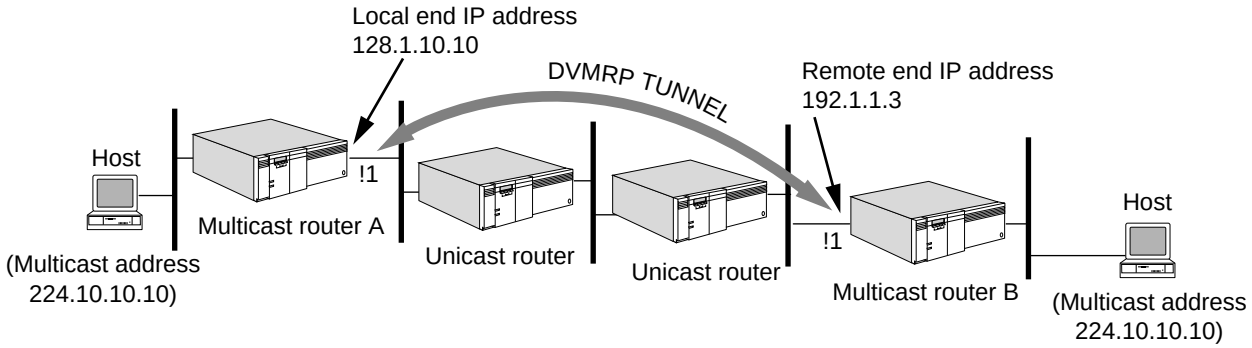


Figure 9-5 Configuring a Multicast Tunnel

1 Create a virtual point-to-point link between the pair of multicast routers using:

```
SETDefault !<tunnel ID> -DVMRP TUnnel = <local-end IP> <remote-end
   IP> [<ttl> (1–255)]
```

For example, on multicast router A, enter:

```
SETDefault !T1 -DVMRP TUnnel = 128.1.10.10 192.1.1.3 3
```

On multicast router B, enter:

```
SETDefault !T1 -DVMRP TUnnel = 192.1.1.3 128.1.10.10 3
```

Up to 32 tunnels can be configured.

The local-end IP address can be any IP address assigned to the system. The remote-end IP address must be unique; you cannot assign tunnels with different local IP addresses and the same remote IP address. The remote-end IP address cannot belong to one of the directly connected subnets if the underlying subnet has broadcast or multicast capability. By default, the TTL is set to 64.

The TTL value should be set to a value greater than or equal to the number of unicast routers in between the multicast routers plus the value of the -MIP THreshold parameter on the remote router interface.

If IP (unicast) routing is not enabled, you must configure a static route for the remote end of the tunnel.

2 Enable DVMRP routing on the specified tunnel interface using:

```
SETDefault !<tunnel ID> -DVMRP CONTrol = Enable
```

Configuring DVMRP Scoping

To configure scoping (filtering), refer to Figure 9-6 and configure a set of multicast destinations that are not reachable through the boundary router port or tunnel.

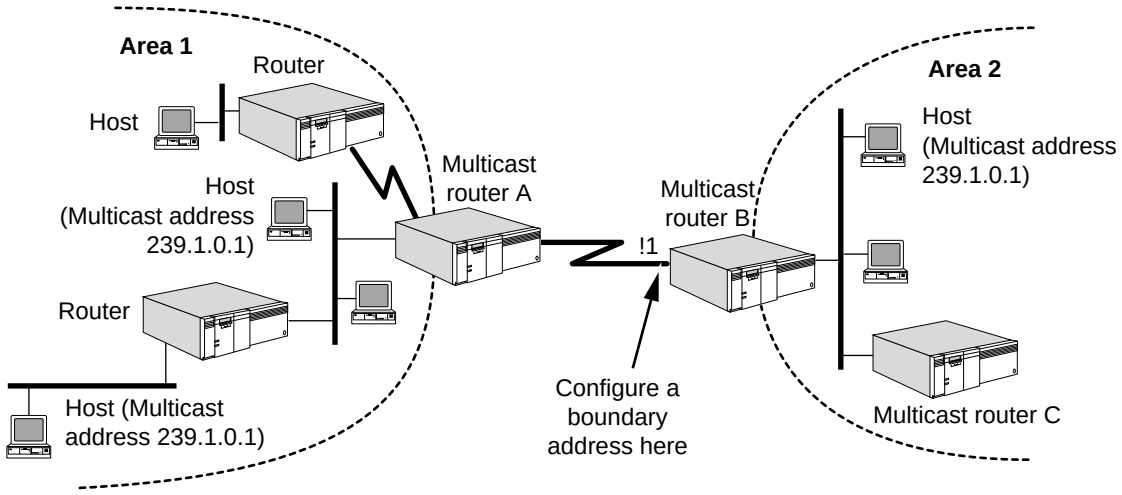


Figure 9-6 Configuring Scoping

Use:

```
ADD {!<port> | !<tunnel ID>} -DVMRP BoundaryAddr <IP addr>
   [<subnet mask>]
```

For example, to configure multicast router B with a boundary address so that packets destined to the group of multicast addresses 239.1.0.1 through 239.1.255.1 are dropped, enter:

```
ADD !1 -DVMRP BoundaryAddr 239.1.1.1 255.255.0.255
```

Packets from area 1 destined to the above address ranges do not reach area 2; packets from area 2 destined to one of the blocked address also do not reach area 1.

You can block a single address by not specifying the subnet mask (the default subnet mask is 255.255.255.255).

Configuring DVMRP Multicasting over Frame Relay

To configure multicasting over Frame Relay, refer to Figure 9-7 and follow these steps on both ends of the link:

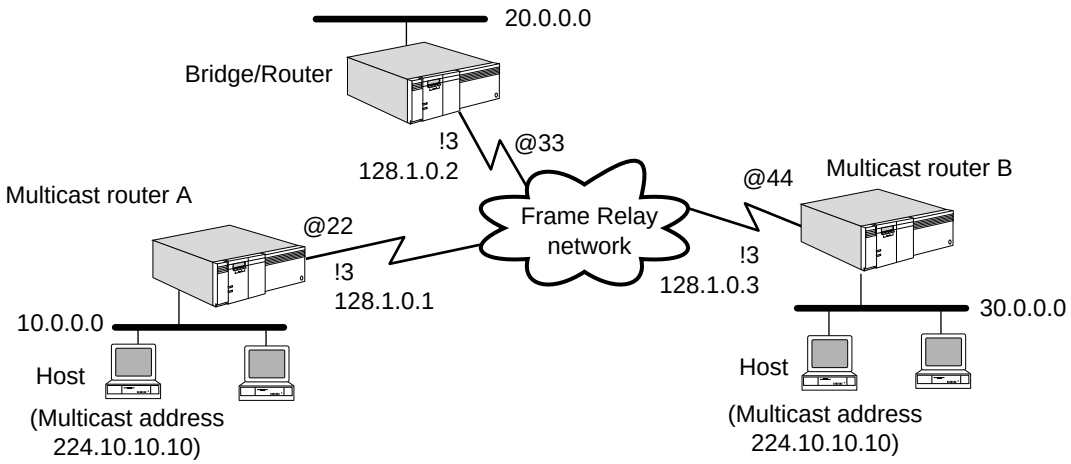


Figure 9-7 Multicasting over Frame Relay

1. Set up the Frame Relay Service as described in "Setting Up the Frame Relay Service" on page 42-1.
2. Assign an IP address to each router wide area port that will perform IP multicasting using:

   ```
   SETDefault !<port> -IP NETaddr = <IP address> [<subnet mask> [Ones
      | Zeros [MTU]]]
   ```

3. Add a neighbor address over the Frame Relay network using:

   ```
   ADD !<port> -DVMRP NEighbor <FR_DLCI>
   ```

   Specify the Frame Relay data link connection identifier (DLCI) address associated with the permanent virtual circuit.

   For example, only multicast routers A and B are participating in multicast routing. To configure the neighbor address, on multicast router A, specifying the DLCI address of multicast router B, enter:

   ```
   ADD !3 -DVMRP NEighbor @44
   ```

   On multicast router B, enter the same command and specify the DLCI address of multicast router A.

4. Enable the DVMRP routing protocol on each wide area port using:

   ```
   SETDefault !<port> -DVMRP CONTrol = Enable
   ```

5. Display neighboring router information using:

   ```
   SHow !<port> -DVMRP NeighborRouter [<IP addr>]
   ```

   If <IP addr> is specified, only neighboring router information for this IP address is displayed. For more information about elements in the display, refer to "NeighborRouter" on page 20-7 in *Reference for NETBuilder Family Software*.

Configuring DVMRP Multicasting over X.25

To configure multicasting over X.25, refer to Figure 9-8 and follow these steps on both ends of the link:

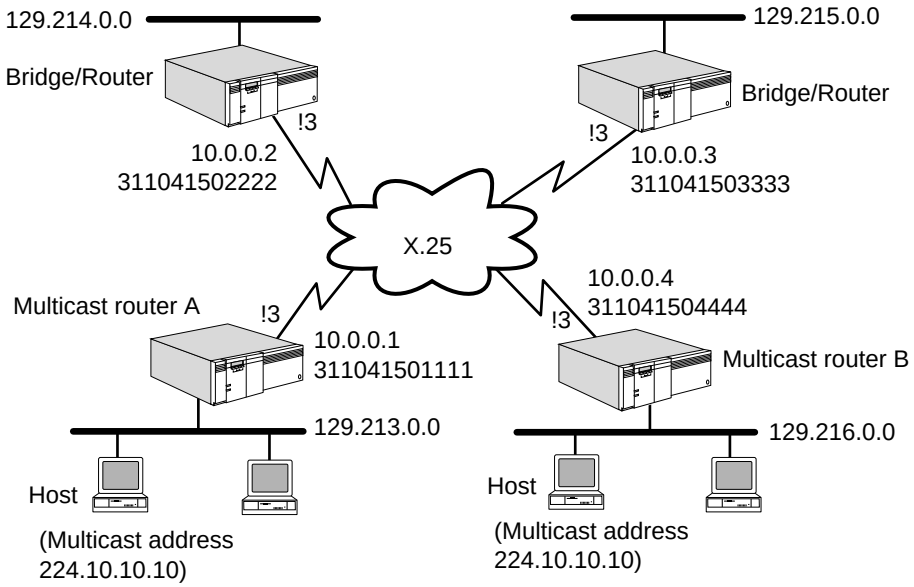


Figure 9-8 Multicasting over X.25

1. Set up the X25 Service as described in "Setting Up the X25 Service" on page 45-1.
2. Assign an IP address to each router wide area port that will perform IP multicasting using:

   ```
   SETDefault !<port> -IP NETaddr = <IP address> [<subnet mask> [Ones
      | Zeros [MTU]]]
   ```

3. Add neighbor address over the X.25 network using:

   ```
   ADD !<port> -DVMRP NEighbor <X.25 DTE>
   ```

   Specify the X.25 address associated with the remote router.

   For example, only multicast routers A and B are participating in multicast routing. To configure the neighbor address, on multicast router A, specifying the DTE address of multicast router B, enter:

   ```
   ADD !3 -DVMRP NEighbor #311041504444
   ```

   On multicast router B, enter the same command and specify the DTE address of multicast router A.

4. Enable the DVMRP routing protocol on each wide area port using:

   ```
   SETDefault !<port> -DVMRP CONTrol = Enable
   ```

5. Display neighboring router information using:

   ```
   SHow !<port> -DVMRP NeighborRouter [<IP addr>]
   ```

   If <IP addr> is specified, only neighboring router information for this IP address is displayed. For more information about elements in the display, refer to "NeighborRouter" on page 20-7 in *Reference for NETBuilder Family Software*.

Configuring a DVMRP Metric

You can configure a metric, or administrative cost, on an interface using:

```
SETDefault {!<port> | !<tunnel ID>} -DVMRP MEtric = <value> (1–31)
```

The default metric is 1.

You may want to adjust the metric if you have multiple routes to the same source and want one route selected over the other. For example, suppose that DVMRP learns about two routes to the same source. Route 1 has an administrative cost of 25; Route 2 has an administrative cost of 3. The DVMRP Protocol selects Route 2 because it is the route with the lowest metric.

Controlling the DVMRP Rate Limit for Multicast Traffic

The DVMRP rate limit is the bandwidth measured in kilobits per second. You can control the rate limit that is allocated for multicast datagram traffic using:

```
SETDefault {!<port> | !<tunnel ID>} -DVMRP RateLimit =
   <Kbits/second> (0–100000)
```

The default is 0, which means that no limit is applied to the given interface, and the interface uses its full bandwidth.

To set the rate limit on multicast router A in Figure 9-9, enter:

```
SETDefault !2 -DVMRP RateLimit = 2000
SETDefault !3 -DVMRP RateLimit = 1600
SETDefault !T1 -DVMRP RateLimit = 500
```

To control your multicast traffic, you need to configure the rate limit if you are connected to the MBONE, which anticipates traffic at a rate of 500 kbps. Refer to Figure 9-9 and the explanation that follows.

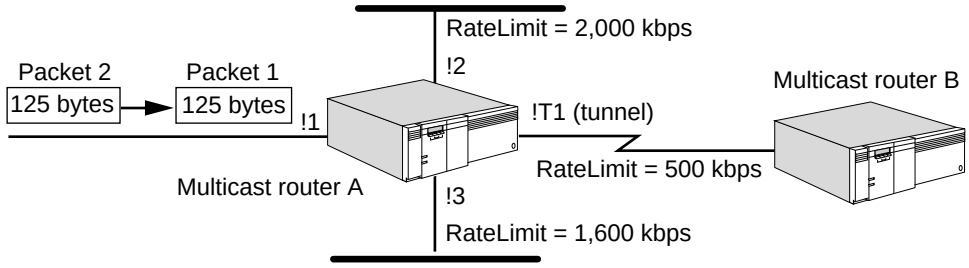


Figure 9-9 Controlling the Rate Limit

When multicast router A in Figure 9-9 receives two 125-byte packets, it queues the packets into the ports' transmit queues (because of the rate limit settings) instead of immediately forwarding them. Multicast router A controls packet forwarding as follows:

After 1 millisecond, port 2 assigns tokens at a rate limit of 2,000 kilobits per second (kbps) (2,000 bits per millisecond or 250 bytes per millisecond). The router extracts both packets from port 2's transmit queue and forwards them to port 2's attached LAN.

Port 3 assigns tokens at a rate limit of 1,600 kbps (1,600 bits per millisecond or 200 bytes per millisecond). The router can forward only the first 125-byte packet on the attached LAN after 1 millisecond. After the next millisecond, port 3 receives another 200 byte token and can transmit up to 275 bytes (200 - 125 + 200); therefore, the router forwards packet 2 (175 bytes) on port 3's attached LAN.

The tunnel interface (!T1) assigns tokens at a rate limit of 500 kbps (500 bits per millisecond or 62.5 bytes per millisecond). After 2 milliseconds, the router forwards packet 1 (125 bytes) on the tunnel interface. After 4 milliseconds, the router forwards packet 2 (175 bytes) on the tunnel interface.

Configuring DVMRP Routing Policies

Using the routing policies supported by DVMRP, you can control the reporting of routes learned from other sources for interautonomous system multicasting.

To configure your DVMRP router to forward multicast packets that have been sourced from a MOSPF domain, see Figure 9-10 and follow these steps on the DVMRP router:

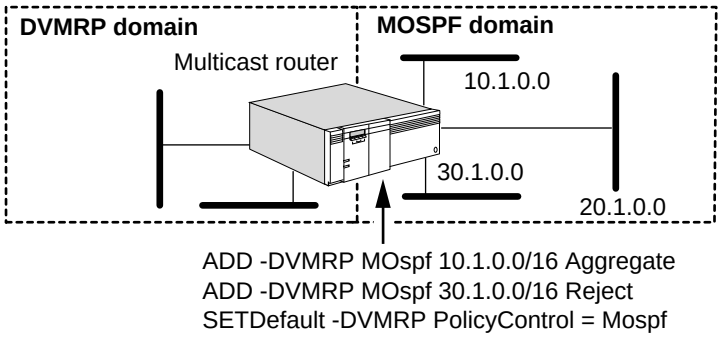


Figure 9-10 DVMRP Routing Policies

1 Enable MOSPF routing information to be advertised into the DVMRP domain using:

```
ADD -DVMRP MOspf <subnet>/<mask> [Aggregate | Individual | Reject]
   [<metric>]
```

Supply the subnet of the MOSPF route to be advertised and a mask value from 0 to 32, which is the number of leading 1s in the mask.

The <subnet>/<mask> syntax describes a range of addresses to either be accepted or rejected by DVMRP. For example:

- 10.0.0.0/8 describes all the subnets within network 10.
- 10.1.0.0/16 describes all the subnets within network 10.1.0.0.
- 0.0.0.0/0 describes all subnets.

An address can fall into multiple subnet/mask ranges. In this situation, the range with the highest mask bits is chosen. The range 0.0.0.0/0 is always the lowest priority.

The keyword Aggregate means that DVMRP advertises a single subnet/mask route, which can summarize multiple networks into a single network. The keyword Individual means that all individual source subnets are accepted and advertised as learned into the DVMRP domain. The keyword Reject means the specified source network is rejected (not advertised).

You can optionally supply a metric value from 1 to 31.

For example, to accept and aggregate routes from 10.1.0.0 advertised as a single route into the DVMRP domain, enter:

```
ADD -DVMRP MOspf 10.1.0.0/16 Aggregate
```

To accept and advertise all routes learned from 20.1.0.0 sourced from the MOSPF domain, enter:

```
ADD -DVMRP MOspf 20.1.0.0/16 Individual
```

To reject all other routes, enter:

```
ADD -DVMRP MOspf 0.0.0.0/0 Reject
```

2 Enable the DVMRP router to perform interautonomous system multicast forwarding by entering:

```
SETDefault -DVMRP PolicyControl = Mospf
```

The router imports routes sourced from MOSPF into the DVMRP routing domain.

The DVMRP Protocol does not import MOSPF external routes. To connect two DVMRP domains separated by an MOSPF domain, you must configure a tunnel to connect the two DVMRP (border) routers, and carefully configure the -DVMRP MOspf parameter.

This parameter only enables the DVMRP domain to accept MOSPF-sourced multicast packets. For the MOSPF domain to accept DVMRP-sourced multicast packets, refer to "Configuring MOSPF Routing Policies" on page 9-19.

Configuring DVMRP Forwarding Policies

Using DVMRP forwarding policies, you can filter destination groups and control data packet forwarding between DVMRP and MOSPF domains.

To configure your DVMRP router for destination group filtering, refer to Figure 9-11 and follow these steps on the multicast router:

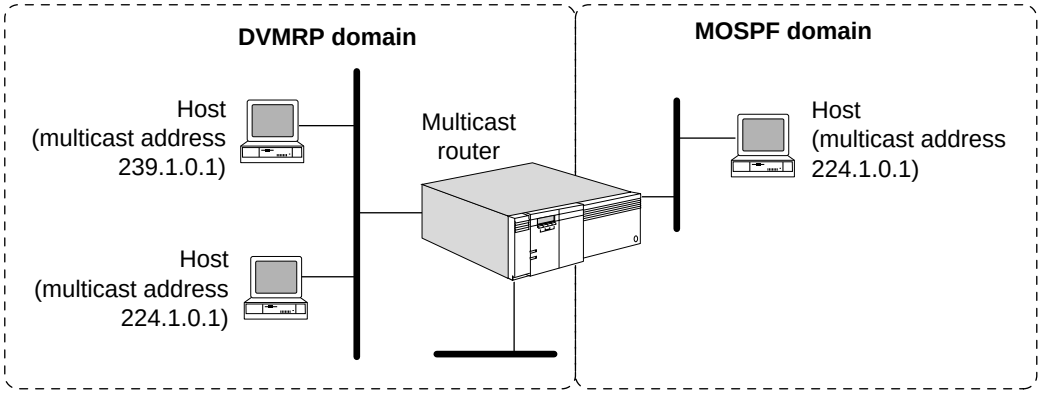


Figure 9-11 DVMRP Destination Group Filtering

1 Configure a list of destination group addresses whose data packets are accepted and forwarded, or rejected and dropped, using:

```
ADD -DVMRP DestGroup <subnet>/<mask> [Accept | Reject]
```

The <subnet>/<mask> syntax describes a range of addresses to either be accepted or rejected by DVMRP. For example:

- 239.0.0.0/8 describes all the addresses within network 239.
- 239.1.0.0/16 describes all the addresses within network 239.1.
- 239.1.10.0/24 describes all the addresses within network 239.1.10.

The Accept option causes the following actions by the multicast router:

- If the multicast router receives a packet from the MOSPF domain with a destination address that matches this destination group filter, then the multicast router accepts it and forwards it into the DVMRP domain.
- If the multicast router receives a packet from the DVMRP domain with a destination address that matches this destination group filter, then the multicast router accepts it and forwards it into the MOSPF domain.

The Reject option causes the following actions by the multicast router:

- If the multicast router receives a packet from the MOSPF domain with a destination address that matches this destination group filter, then the multicast router rejects it and drops the packet and never forwards it into the DVMRP domain.
- If the multicast router receives a packet from the DVMRP domain with a destination address that matches this destination group filter, then the multicast router rejects it and drops the packet and never forwards it into the MOSPF domain.

For example, to configure data packets to the destination group 224.1.0.1 to be rejected on the multicast router, enter:

```
ADD -DVMRP DestGroup 224.1.0.0/16 Reject
```

2 Enable the policy by entering:

```
SETDefault -DVMRP PolicyControl = DestGroup
```

Data packets to destination group addresses from 224.1.0.0 to 224.1.255.255 between DVMRP and MOSPF domains are rejected and dropped by the multicast router.

Configuring DVMRP Route Aggregation

With DVMRP route aggregation, you can combine the characteristics of several different routes so that a single route can be advertised. By combining several networks into one supernet, the number of route report messages and the size of the routing table are reduced.

To configure route aggregation, refer to Figure 9-12 and follow these steps:

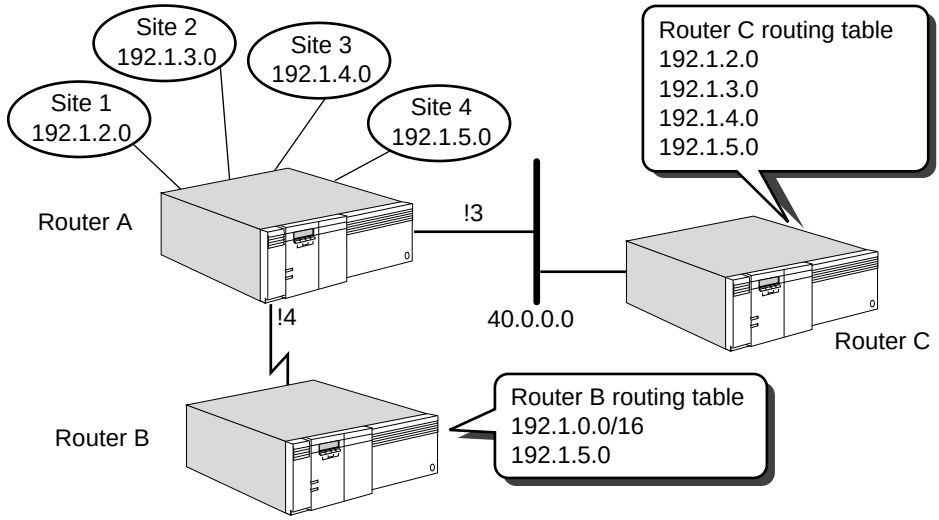


Figure 9-12 DVMRP Route Aggregation

1 Specify a list of networks that DVMRP advertises as a single supernet route using:

```
ADD -DVMRP AggregateRange <subnet>/<mask> [<metric>]
```

For example, to combine the routes to sites 1, 2, 3, and 4 into a range so that only a single route is advertised on router A, enter:

```
ADD -DVMRP AggregateRange 192.1.0.0/16
```

2 Specify a list of routes that DVMRP explicitly advertises using:

```
ADD -DVMRP AggregateExcept <subnet>/<mask>
```

For example, if you do not want site 4 included in the aggregation range, enter:

```
ADD -DVMRP AggregateExcept 192.1.5.0/24
```

3 Enable route aggregation and the DVMRP routing protocol by entering:

```
SETDefault !4 -DVMRP CONTrol = (Enable, Aggregate)
```

As shown in Figure 9-12, router A advertises a single network (192.1.0.0/16) that summarizes each of the three connected sites and also explicitly advertises the exception route (192.1.5.0) to router B. Without the use of aggregation, router A advertises each route with a separate entry as shown in the router C

routing table, which grows in size. With route aggregation, the router B routing table has an entry for 192.1.0.0 and 192.1.5.0.

Controlling the Routing Table

You can control how often the router sends route report messages, delete entries in the routing table, and display the routing table.

To control how often the router sends route report messages containing the complete routing table, use:

```
SETDefault -DVMRP UpdateTime = <seconds>(5–5400)
```

By default, DVMRP updates the routing table every 60 seconds. By changing this setting, you affect how long a route is considered valid (RouteExpirationTime) and how long a route exists without confirmation (GarbageCollectionTime). The RouteExpirationTime is equal to three times the value of this parameter, and the GarbageCollectionTime is equal to five times the value of this parameter. By increasing the value of the UpdateTime parameter, you can reduce the amount of route report traffic but you may also increase the size of the routing table.

This parameter can determine how long a neighbor is considered "up" without confirmation (NeighborExpireTime) and when to consider the associated virtual interface as a leaf link (LeafConfirmationTime). The NeighborExpireTime is set to two times the value of this parameter plus 20 seconds, and the LeafConfirmationTime is set to three times the value of this parameter plus 20 seconds.

To flush entries in the routing table learned from DVMRP, use:

```
FLush -DVMRP RouteTable
```

To display the routing table, use:

```
SHow -DVMRP RouteTable [<subnet>[/<mask>]] [Long]
```

If the <subnet> and/or <mask> syntax is specified, the routing table for the range of specified subnets is displayed. If Long is specified, the display shows a lists of ports that connect to child subtrees and leaf subnets.

For example, to display the following table, enter:

```
SHow -DVMRP RouteTable Long
```

```
SourceSubnet   SubnetMask    FromGateway    Metric   Status  TTL    InPort  OutPorts
20.0.0.0       255.0.0.0     11.11.11.11    3        Up      200    1       2, 3*, 4, 5
30.0.0.0       255.0.0.0     ----           0        Up      --     2       1, 3*, 4, 5
40.0.0.0       255.0.0.0     11.11.11.11    6        GC      100    1       2, 3*, 4
```

The display consists of the following items:

SourceSubnet	The original subnet from which the multicast datagram originated.
SubnetMask	The subnet mask of the source subnet.
FromGateway	The previous hop router that leads back to the source. If no gateway is specified, the subnet is directly connected.

Metric	The routing metric of the path back to the source subnet.
Status	The status of this route entry: Up, GC (Garbage-Collect), HD (Hold-Down), and Down.
TTL	Time-to-live indicates how much time (in seconds) is left before removing an entry from routing table.
InPort	Incoming port for the multicast datagrams from that source.
OutPorts	List of ports on which multicast datagrams originated from this source are forwarded. An asterisk (*) indicates that the outgoing port connects to a leaf of the multicast delivery tree rooted at this source.

Controlling the Forwarding Table

You can specify how long you want to keep a (source, group) pair in the forwarding table and display the contents of this table.

To control how long entries remain in the forwarding table, use:

```
SETDefault -DVMRP CacheTime = <seconds> (300–86400)
```

The default value of this parameter is 300 seconds. You can adjust the setting up to 1 day (86,400 seconds). By adjusting the CacheTime parameter, you can control the size of the forwarding table.

To display entries in the forwarding table, use:

```
SHow -DVMRP ForwardTable [<subnet>[/<mask>]] [<group>]
```

You can display the current table for each (source, group) pair. If you specify only the subnet, all group entries associated with this subnet are displayed. If you specify only the group, all source subnets associated with this group are displayed. If you specify both the subnet and group, only this particular entry is displayed.

For example, to display the following table, enter:

```
SHow -DVMRP ForwardTable
```

```
SourceSubnet   MulticastGroup  TTL    InPort   OutPorts
20.0.0.0       224.1.1.1       200    1 Pr     2p 3p 4p
               224.2.2.2       100    1        2p 3 4
               224.3.3.3       250    1        2 4b
30.0.0.0       224.1.1.1       300    1        2 3 4
               239.4.4.4       100    1 Sc
```

The display consists of the following items:

SourceSubnet	The original subnet of multicast datagrams.
MulticastGroup	The group address to which multicast datagrams are destined from the origin.
TTL	Time-to-live indicates how much time (in seconds) are left before removing a source and group entry from the table.
InPort	Indicates the incoming port for the multicast datagrams from that source. Pr A Prune message is sent to the upstream router.

Sc The multicast group address is configured as a boundary address, and no traffic for this group address is forwarded from that port.

OutPorts Indicates the ports that multicast datagrams belonging to this group are forwarded.

p The port receives all the Prune messages of the downstream neighboring routers, and no multicast datagrams are forwarded to this port.

b The multicast group address is configured as a boundary address, and no traffic for this group address is forwarded to this port.

Using the MOSPF Protocol

The following sections describe how to further customize your multicast router if you are using the MOSPF Protocol as the multicast routing protocol.

Configuring Interarea Multicasting

To perform interarea multicasting when running the MOSPF Protocol, the Area Border Router (ABR) must be configured as an interarea multicast forwarder, which is an ABR with multicast extensions enabled.

As shown in Figure 9-13, the ABR connects two areas to the backbone. The ABR must be configured as an interarea multicast forwarder so that it can summarize group membership information from attached nonbackbone areas into the backbone and to forward multicast packets between areas.

To allow multicasting between areas, enter:

```
SETDefault -MOSPF MABR = Enable
```

The router must be an OSPF ABR for the MABR parameter to take effect. By default, this parameter is enabled.

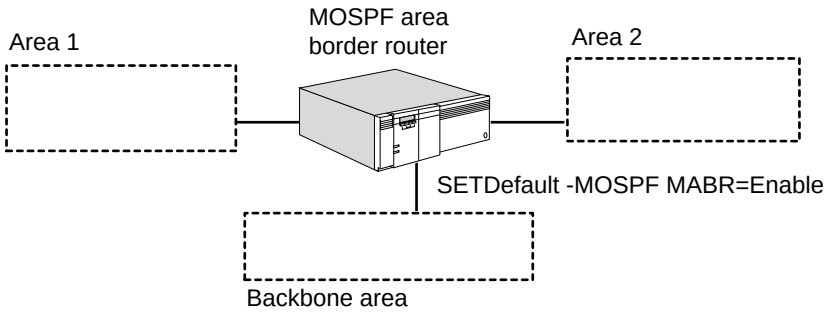


Figure 9-13 Interarea Multicasting

For more information, refer to “Interarea Multicasting” on page 9-30.

Configuring MOSPF Routing Policies

Using the routing policies supported by MOSPF, you can control the reporting of routes learned from other sources for interautonomous system multicasting. The current implementation of MOSPF routing policies only supports DVMRP as the source of multicast traffic.

To configure your MOSPF router to forward multicast packets that have been sourced from a DVMRP domain, refer to Figure 9-14 and follow these steps on the MOSPF router:

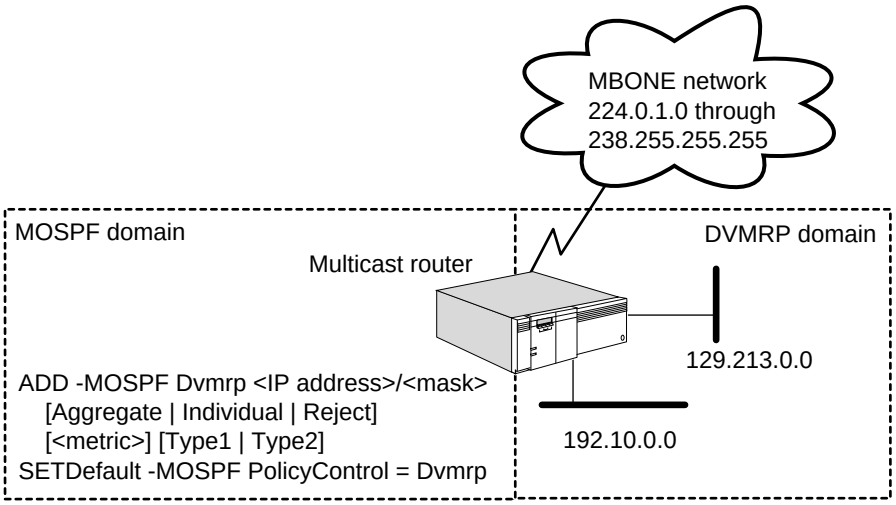


Figure 9-14 MOSPF Routing Policies

1 Enable DVMRP routing information to be advertised into the MOSPF domain using:

```
ADD -MOSPF Dvmrp <subnet>/<mask> [Aggregate | Individual | Reject]
   [<metric>] [Type1 | Type2]
```

Supply the subnet and mask of the address range of the DVMRP route to be advertised. The mask value is the number of leading 1s in the mask and ranges from 0 to 32.

The <subnet>/<mask> describes a range of addresses. For example:

- 10.0.0.0/8 describes all the subnets within network 10.
- 10.1.0.0/16 describes all the subnets within 10.1.0.0.
- 0.0.0.0/0 describes all subnets.

An address can fall into multiple subnet/mask ranges. In this situation, the range with the highest mask bits is chosen. The range 0.0.0.0/0 is always the lowest priority.

The keyword Aggregate means that MOSPF advertises a single subnet/mask route, which can summarize multiple networks into a single network. The keyword Individual means that all individual source subnets are accepted and advertised as learned into the MOSPF domain. The keyword Reject means the specified source network is rejected (not advertised).

You can optionally supply a metric value from 0 to 65,535.

You can select either Type1 or Type2. Type1 advertises the routes as a type 1 external LSA, which is always preferred over a type 2 external LSA for the same destination.

For example, to accept and aggregate routes from 192.10.10.0 advertised as a single route into the MOSPF domain, enter:

```
ADD -MOSPF Dvmrp 192.10.10.0/24 Aggregate
```

To accept and advertise all routes learned from 129.213.0.0 sourced from the DVMRP domain, enter:

```
ADD -MOSPF Dvmrp 129.213.0.0/16 Individual
```

To reject all other routes, including transmissions from the MBONE, enter:

```
ADD -MOSPF Dvmrp 0.0.0.0/0 Reject
```

2 Enable the MOSPF router to perform interautonomous system multicast forwarding by entering:

```
SETDefault -MOSPF PolicyControl = Dvmrp
```

When this command is executed, the MOSPF router declares itself as a wild-card multicast receiver to all its attached areas to attract multicast packets to all destinations. It imports specified routes sourced from DVMRP into the MOSPF routing domain as external LSAs.

This parameter only enables the MOSPF domain to accept DVMRP-sourced multicast packets. For the DVMRP domain to accept MOSPF-sourced multicast packets, refer to "Configuring DVMRP Routing Policies" on page 9-13. Failure to configure the DVMRP routing policies results in half-duplex communication.

Configuring MOSPF Forwarding Policies

Using MOSPF forwarding policies, you can filter destination groups and control data packet forwarding between MOSPF and DVMRP domains.

To configure your MOSPF router for destination group filtering, refer to Figure 9-15 and follow these steps on the multicast router:

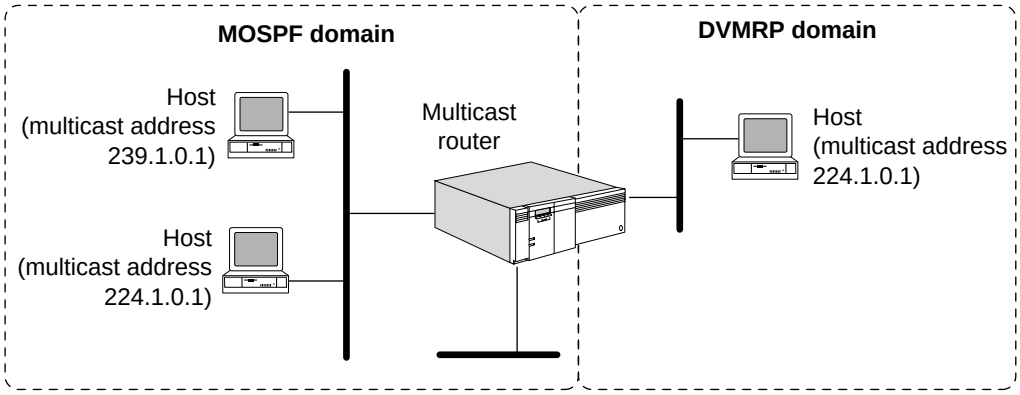


Figure 9-15 MOSPF Destination Group Filtering

1 Configure a list of destination group addresses whose data packets are accepted and forwarded, or rejected and dropped, using:

```
ADD -MOSPF DestGroup <subnet>/<mask> [Accept | Reject]
```

The <subnet>/<mask> syntax describes a range of addresses to either be accepted or rejected by MOSPF. For example:

- 239.0.0.0/8 describes all the addresses within network 239.
- 239.1.0.0/16 describes all the addresses within network 239.1.
- 239.1.10.0/24 describes all the addresses within network 239.1.10.

The Accept option causes the following actions by the multicast router:

- If the multicast router receives a packet from the DVMRP domain with a destination address that matches this destination group filter, then the multicast router accepts it and forwards it into the MOSPF domain;
- If the multicast router receives a packet from the MOSPF domain with a destination address that matches this destination group filter, then the multicast router accepts it and forwards it into the DVMRP domain.

The Reject option causes the following actions by the multicast router:

- If the multicast router receives a packet from the DVMRP domain with a destination address that matches this destination group filter, then the multicast router rejects it and drops the packet and never forwards it into the MOSPF domain.
- If the multicast router receives a packet from the MOSPF domain with a destination address that matches this destination group filter, then the multicast router rejects it and drops the packet and never forwards it into the DVMRP domain.

For example, to configure data packets to the destination group 224.1.0.1 to be rejected on the multicast router, enter:

```
ADD -MOSPF DestGroup 224.1.0.0/16 Reject
```

2 Enable the policy by entering:

```
SETDefault -MOSPF PolicyControl = DestGroup
```

Data packets to destination group addresses 224.1.0.0 to 224.1.255.255 between DVMRP and MOSPF domains are rejected and dropped by the multicast router.

Displaying the Forwarding Table

You can display the MOSPF forwarding cache using:

```
SHow -MOSPF ForwardTable [<destination>]
```

If the destination is specified, only entries toward that destination are displayed.

The display contains the following items:

DESTINATION — The destination group address. All destinations are Class D IP addresses in the range of 224.0.0.0–239.255.255.255.

Addresses from 224.0.0.0 through 224.0.0.225 are reserved for standard usage. Addresses 239.0.0.0 through 239.255.255.255 are reserved for use within organizations and are not used across the Internet.

SOURCE — The originator of the multicast IP packets. Each pair of Destination/Source identifies a particular forwarding cache.

UPSTREAM — The router or interface from which packets come. Four values are possible:

- None — No routes available. All multicast packets with the matching Destination/Source are discarded.
- External — Multicast packets are expected to be received from an interdomain multicast routing protocol such as DVMRP or from an internal client of MOSPF in the NETBuilder II bridge/router such as DLSw multicast.

	!<port>	Multicast packets are expected from a particular interface using MAC-level multicasting addresses. Packets satisfying the conditions are accepted.
	<IP address>	Multicast packets are expected from a particular neighbor, identified by the IP address. Only packets received from this neighbor are accepted; otherwise, the packets are dropped. This format is associated with a WAN interface (such as Frame Relay or X.25), or a LAN interface where the Unicast value of the -MOSPF CONTrol parameter is selected. If there are multiple WAN links (such as PPP, X.25, or Frame Relay) between two routers, more than one upstream router can be displayed, and packets from either upstream router are accepted.
DOWNSTREAM	Interfaces or routers to forward to. Three values are possible:	
	!<port>	Multicast packets are delivered to a particular interface using MAC-level multicasting addresses. This format is associated with LAN interfaces, where the Multicast value of the -MOSPF CONTrol parameter is selected, or with a PPP interface.
	<IP address>	Multicast packets are delivered to a particular neighbor identified by the IP address. This format is used with a WAN interface, or interfaces that have the Unicast value of the -MOSPF CONTrol parameter selected.
	DVMRP	Packets delivered to the DVMRP Protocol for further distribution in the DVMRP domain.
	Internal	Packets are delivered to the internal client of MOSPF in the NETBuilder II bridge/router, such as DLSw multicast.
TTL	The hop count before reaching the nearest group member.	

How the IP Multicast Router Works

Multicasting allows a host to transmit an IP datagram to a set of hosts that form a multicast group. Every member in a multicast group that uses the same multicast address can receive a copy of the IP datagram. Multicast routers are responsible for delivering (and copying) datagrams to those hosts.

Membership in a multicast group is dynamic. A host may join or leave a group at any time. A host may be a member of an arbitrary number of multicast groups; group members can span multiple subnets. Membership in a group determines whether the host receives datagrams sent to the multicast group; however, a host may send datagrams to a multicast group without being a member.

Each multicast group has a unique multicast (Class D) address. Some multicast addresses are assigned by the Internet Addressing and Naming Authority (IANA) and correspond to groups that always exist even if they have no current

members. Such addresses are said to be well-known. Typically, packets transmitted to these addresses use a TTL of 1. Other multicast addresses are available for temporary use. They correspond to transient multicast groups that are created when needed and discarded when the membership reaches zero. For more information, refer to "Multicast Addresses" on page 9-25.

Special gateways, or routers, forward multicast datagrams, but hosts do not need to explicitly know about these routers. It is the responsibility of the multicast router to receive the multicast packet from the host and correctly forward it to those members of the group.

Hosts and routers must run the IGMP Protocol for multicast connectivity. In addition, the router must run one or more of the following multicasting routing protocols:

- DVMRP
- MOSPF Version 2
- Core-Based Trees (CBT)
- Protocol Independent Multicast (PIM) (Sparse and/or Dense Mode)

The NETBuilder software includes the DVMRP and MOSPF routing protocols, which are user configurable, and the IGMP Protocol, which requires no configuration. For more information, refer to "Distance Vector Multicast Routing Protocol" on page 9-26, "Multicast Open Shortest Path First Protocol" on page 9-28, and to "Internet Group Management Protocol" on page 9-25.

MBONE Connectivity with Multicasting

The MBONE is a virtual network running on top of the Internet that is composed of a cooperative set of workstations and routers with multicast capability. The MBONE has been in existence since 1992, primarily as a research and collaboration tool using multimedia applications. It has been greatly expanded from the original Internet Engineering Task Force (IETF) video and audio multicasts, and now includes 24-hour world news audio sessions and NASA space missions, which use real-time audio and video transmissions.

With the 3Com implementation of IP multicasting, you can have the following advantages:

- Obtain audio and video transmissions using your existing infrastructure (over Ethernet, FDDI, or token ring) and on any media over which 3Com supports IP routing.
- Enable the development of entirely new classes of IP-based applications.
- Ease the migration of existing LAN-based multicast applications and distributed systems to an IP-based environment.
- Conserve bandwidth by reducing traffic and protect the host from receiving unwanted datagrams (only members of the group receive the multicast packet).
- Extend the benefits of multicast delivery beyond the confines of a single subnetwork as more multicast-capable IP routers are used.
- Access the MBONE across the Internet using tunneling.
- Experience complete compatibility with the UNIX program, mrouted 3.5 and above (less compatibility with previous releases), the UNIX program implementing DVMRP that runs on most systems on the MBONE.

Multicast Addresses

IP multicasting uses the destination address of the datagram to specify multicast delivery using Class D addresses in the range of 224.0.0.0 through 239.255.255.255.

The following Class D addresses are reserved:

- 224.0.0.0 – this address cannot be assigned to any group.
- 224.0.0.1 – this address is permanently assigned to the "all hosts" group, which includes all hosts and gateway participating in IP multicasting on a local network. No IP multicast address exists that refers to all hosts in the Internet.
- 224.0.0.2 – this address is assigned to all routers on a local network.
- 224.0.0.4 – this address is assigned to DVMRP routers on a local network.
- 224.0.0.5 – this address is assigned to all OSPF routers on a local network.
- 244.0.0.6 – this address is assigned to all OSPF designated routers and backup designated routers on a local network.
- 224.0.0.0 to 224.0.0.255 – these addresses are reserved for multicast applications that do not multicast more than one hop. Multicast packets addressed to these addresses are not forwarded outside the local network.
- 239.0.0.0 to 239.255.255.255 – these addresses are reserved for scoping purposes (a router is configured as a boundary router and multicast traffic does not cross the boundary) and for private multicast groups (traffic is not routed across the Internet).

IP multicast addresses can only be used as the destination address; they can never appear in the source address field of a datagram, nor can they appear in a source route or record route option. For more information about IP addressing, refer to Appendix D.

Internet Group Management Protocol

To participate in IP multicasting, multicast hosts and routers must have the IGMP operating. This protocol is the group membership protocol used by hosts to inform routers of the existence of members on their directly connected networks, and allows them to send and receive multicast datagrams.

Multicast routers learn about group membership when a host joining a new group sends an IGMP message to the group address declaring its membership. If the DVMRP Protocol is running, the local multicast router receives the group membership message and sends a DVMRP Graft message to its upstream router if it ever sent a DVMRP Prune message. If the MOSPF Protocol is running, the local multicast router receives the group membership message, establishes routes, and propagates the group membership information to other multicast routers throughout the internetwork.

Because membership is dynamic, local multicast routers periodically query hosts on the local network with Host Membership Query messages to determine which hosts remain members of which groups. These messages are periodically sent by the designated router (the one with the lowest IP address in DVMRP or the one with the highest router priority in MOSPF) to refresh their knowledge of membership present on a particular subnet. Hosts respond with Host Membership Report messages. If no host reports membership in a group after a query, the multicast router assumes that no host on the network remains in that

group. If the DVMRP is running, the router sends a Prune message to its upstream router for the next data packet destined to this group and assumes that no other downstream routers are interested in this group. If the MOSPF Protocol is running, the router stops advertising group membership to other multicast routers. Hosts can also send Host Leaves Group messages whenever they want to leave a multicast group.

The information learned by the IGMP is stored in a local group membership database and is used by both the DVMRP and MOSPF Protocols.

Distance Vector Multicast Routing Protocol

To propagate routing information among multicast routers, a multicast routing protocol such as DVMRP can be used. Multicast routers use the DVMRP to pass source subnet information among themselves, using the information to establish routes to deliver a copy of the multicast datagram to every subnet containing a member of the multicast group.

Like the RIP, the DVMRP passes information about known subnets and the cost to route between gateways. For each possible multicast group, the router imposes a routing tree on top of the graph of the physical interconnections. When a router receives a datagram destined for an IP multicast address, it sends a copy of the datagram over the network links that correspond to branches in the routing tree.

The 3Com implementation of the DVMRP applies the Reverse Path Multicasting (RPM) algorithm that allows for the shortest-path multicast tree to be pruned on demand. Pruning preserves bandwidth by removing multicast routers from the tree when no members for that group are on any directly connected subnets and no downstream routers are interested in that group (multicast packets do not need to be received and are discarded by this router because no group members are attached).

The DVMRP uses a number of messages to discover neighboring routers. Some of these messages include the following:

- Probe – discovers neighbors that support multicast routing.
- Route Report – contains route information.
- Prune – destined to the parent router to detach it from the delivery tree if no members for that group are on any directly connected subnets.
- Graft – sent to an upstream router when a new member joins the group after a Prune message had previously been sent.
- Graft Acknowledge – sent to the downstream router to acknowledge the previous Graft message.

Routing Table

Each DVMRP multicast router creates a routing table containing a list of routes learned from other multicast router's route report messages. Using these route report messages, the router builds a routing table and a shortest-path tree for each source.

The router also keeps track of the following links:

- Parent link

  A parent link is the expected interface to receive multicast packets from a source (the interface that leads to the previous-hop router back to the source).

- Child link

  For each (source, group) pair, the child links are the set of interfaces on which to forward multicast packets. The router uses the child link information to perform Reverse Path Broadcasting (RPB).

- Leaf link

  A leaf link is a child link that no router uses to reach a source. For a given source, if no members of a particular group on the subnet are associated with a leaf link, DVMRP truncates the leaf link from the shortest path tree using the Truncated Reverse Path Broadcasting (TRPB) algorithm.

The DVMRP router also assigns the following router functions:

- Designated router

  The router with the lowest IP address on a subnet becomes the designated router. The designated router is responsible for sending IGMP Host Membership Query datagrams on the subnet.

  When a multicast router starts, it considers itself to be the designated router until it receives a Host Membership Query or Report datagram from a neighbor router with a lower IP address.

- Dominant router

  To avoid duplicate multicast datagrams when more than one router exists on a virtual interface, one router is elected as the dominant router for a particular source. The dominant router is the router that is responsible for forwarding multicast datagrams on a subnet for a source (it has a route to the source with the lowest metric on that virtual interface).

- Subordinate router

  A subordinate router for a virtual interface is the downstream router that considers this interface to be its parent link. Information from a subordinate router helps the DVMRP router decide whether to truncate the shortest path tree. For each route entry, the subordinate router helps decide if the subnet for that virtual interface is a leaf subnet.

Forwarding Table

In conjunction with the routing table, the DVMRP creates a forwarding table. The forwarding table contains group information (source and group pairs) that is applied to the routing table's shortest-path tree. The forwarding table helps the router forward multicast datagrams to each member of the group using the routing table's shortest-path tree.

The DVMRP router can receive Prune messages from downstream routers in the shortest-path tree if the attached subnet contains no group members for the particular (source, group) pair. In this way, the router can prune the shortest-path delivery tree, allowing datagrams to only be forwarded to the subnets in which the specified group is located. The DVMRP leaf router also

prunes the shortest-path delivery tree if it no longer receives IGMP Host Membership Report messages or if all members have left a group. The forwarding table maintains an entry in its cache until the timeout period is reached. During the timeout period, if the DVMRP router learns that members have rejoined a group, it sends a Graft message to the upstream routers indicating that a member has rejoined and allows the branches of the shortest-path tree to reattach.

Multicast Open Shortest Path First Protocol

To propagate routing information among multicast routers, a multicast routing protocol such as MOSPF can be used. MOSPF is an extension of the base version 2 OSPF Protocol and is backward compatible with OSPF (routers running OSPF interoperate with MOSPF routers). The introduction of multicast extensions does not impact unicast IP traffic. MOSPF routers identify other MOSPF-capable routers for forwarding multicast IP packets. Unlike DVMRP, where separate routing protocols for unicast and multicast packets are run, OSPF and MOSPF run a single copy of the protocol. But like DVMRP, MOSPF forwards multicast traffic based on both the source and destination address, known as source and destination routing.

The MOSPF Protocol does not provide the ability to tunnel through non-MOSPF capable routers. MOSPF routers must be directly interconnected with each other. Failure to do so may lead to nondelivery of multicast packets even though unicast connectivity is maintained.

While forwarding multicast packets, MOSPF may replicate packets along the way. The replication is performed only at tree branches where replication is absolutely necessary. Although multiple copies may be forwarded, the packet is not modified (except the TTL field, where it is decrement by 1 at each hop). No IP-over-IP encapsulation is performed. The destination address is always listed as Class D multicast address. To avoid packet duplicates, equal-cost multiple path forwarding in MOSPF is not possible.

When sending multicast IP packets, MOSPF conforms to link-layer encapsulation. Over Ethernet and FDDI interfaces, the mapping between IP multicast and datalink multicast address is used. Over other kinds of LAN interfaces, link-level multicast or broadcast is used. Over WAN media, IP multicast packets are encapsulated as unicast packets.

OSPF partitions the network topology into a number of routing domains, with ASBRs interconnecting routing domains. Within a routing domain, OSPF allows multiple areas to be interconnected by ABRs. Areas may be transit, stub, or backbone. MOSPF partitions the network topology in the same way as OSPF; the same topology for both OSPF and MOSPF can be used. For more information on the OSPF topology, refer to “Understanding IP Network Topology” on page 6-38.

Learning Group Membership

MOSPF uses the IGMP Protocol to monitor multicast group membership on directly attached LANs. MOSPF periodically sends IGMP queries and listens to IGMP replies. The membership information learned is then used to build group-membership link state advertisements (LSAs).

On a LAN, only the designated MOSPF router (usually the one with the highest router priority) sends queries at the interval specified by the -MIP QueryInterval parameter to the "all hosts" address (224.0.0.1) and listens to IGMP replies. The MOSPF designated router (DR) processes IGMP replies and performs the IGMP maintenance work on the network. The DR is responsible for flooding group membership information throughout the routing domain by issuing group membership LSAs. When a new group is learned, MOSPF sends a new group membership LSA. When a group is aged out, MOSPF flushes the corresponding group membership LSA. When the MOSPF router resigns as the DR, it flushes all locally generated group membership LSAs.

In a mixed environment in which MOSPF and OSPF routers reside on the same LAN, an MOSPF router must become the DR to monitor group membership, generate group-membership LSAs, and forward multicast packets onto the LAN. Therefore, OSPF routers should be assigned a router priority of 0 to prevent them from becoming the DR, allowing an MOSPF router to become the DR.

Shortest Path First Tree

MOSPF uses the group membership LSA with the OSPF database, which provides complete topology information about the area and routing domain. The group-membership LSAs describe the location and address of all multicast groups in an area and routing domain. The group membership database is built by the IGMP Protocol and enables delivery of multicast packets.

MOSPF routers use the group membership LSA information to compute the shortest path first (SPF) tree, which enables delivery of multicast packets to remote destinations. The SPF tree is rooted at the packet's source address toward all destination group members and describes the intermediate hops from the source to all possible destinations belonging to the same group. Different sources are likely to have different trees. The SPF tree is pruned only toward the intended destination; all paths and routers that do not lead to group members are pruned from the tree. A separate tree is built for each source and destination pair.

The SPF tree is computed on demand (when a packet is received). A cache entry is created with the source and destination pair; the upstream node and downstream interface information is recorded. The SPF tree is then discarded, freeing all resources along with it. The newly created cache entry is used for forwarding decisions, and the entry is stored in the forwarding database. Future received packets with the same source and destination pair can locate its forwarding decision from the database without resorting to another SPF computation.

Forwarding Cache

Each MOSPF router in the path of a multicast packet makes its forwarding decision based on the contents of its forwarding cache. The forwarding cache is built from the local group database and the SPF tree. Each cache entry contains information about received multicast packets from the neighboring node (upstream router or LAN) and where multicast packets should be forwarded (downstream interfaces or MOSPF neighbors). Each downstream interface has a time-to-live (TTL) value associated with it. The TTL value indicates the number of hops a datagram can travel to reach the nearest multicast destination or be discarded. The hop count prevents packets from being uselessly forwarded and conserves bandwidth. The hop count is further restricted by the -MIP THreshold parameter.

The cached information is not aged or periodically refreshed; the information is kept as long as enough system resource are available, or until the next topology change. However, the forwarding cache may need to be flushed under the following circumstances:

- OSPF topology changes
- Group membership LSA changes with identical multicast destination
- Local group database changes with identical multicast destination

Interarea Multicasting

When multicast routing occurs between areas (interarea multicasting), source and destination addresses may not reside in the same area, the ABR must have multiple copies of the OSPF link databases (one for each area), and the MOSPF router must build separate SPF trees for each area.

Recall that ABRs are responsible for interconnecting areas (transit or stub) to the backbone and other areas. The backbone area is considered a transit area, with area number 0 reserved for it. All ABRs must be connected to the backbone area, either directly or through virtual links. The ABRs are responsible for summarizing reachability information from the backbone to other areas, and from other areas to the backbone. These summaries take the form of summary LSAs.

When running the MOSPF Protocol, a portion of the OSPF ABRs must be configured through the -MOSPF MABR parameter as interarea multicast forwarders, which are ABRs with multicast extensions enabled. An interarea multicast forwarder must be an ABR, but not all ABRs need to be interarea multicast forwarder.

The interarea multicast forwarder calculates all the reachable group addresses from their areas. They convey group membership information to other areas by summarizing the group membership LSAs from their attached areas into the backbone. They do not summarize group membership information from the backbone to other areas. All interarea multicast forwarders concurrently and independently perform this action.

After the router summarizes group membership LSAs into the backbone, the backbone area has complete information regarding all the reachable group memberships. The backbone area may not know the exact location of group members subnets (because that requires the detailed topology information from within the area), but it knows which area is interested in which group address. Nonbackbone areas have only group membership information for their area and do not know that some group members exist in other areas.

For multicast packets to flow between areas, all interarea multicast forwarders announce wild-card multicast receiver status (equivalent to the default route for unicast traffic) into attached areas. A wild-card multicast receiver is a router to which all multicast packets should be forwarded regardless of the multicast destination. With sufficient routing information in a backbone area, a wild-card multicast receiver is not needed. Interarea multicast forwarders do not announce wild-card multicast receiver into the backbone.

Wild-card multicast receiver status is automatic; no user configuration is required. In nonbackbone areas, all interarea multicast forwarders are wild-card multicast receivers. Backbone area do not need these receivers.

- When MOSPF routers are used between areas, they perform one SPF computation for the source and destination per attached area. Each area has its own link state database, and the SPF computation exclusively uses the LSAs within the area. The backbone area is treated the same as other areas.

Interautonomous System Multicasting

When multicast routing occurs between autonomous systems (interautonomous system multicasting), some MOSPF routers must be configured as inter-AS multicast forwarders. These inter-AS multicast forwarders have additional routing information for forwarding multicast packets outside the routing domain. These inter-AS multicast forwarders can concurrently run another inter-AS multicast protocol in the same router or be configured with static external routes. However, the current implementation does not support static routes. Inter-AS multicast routers are configured through the -MOSPF Dvmrp and PolicyControl parameters.

The MOSPF Protocol guarantees that all inter-AS multicast forwarders receive all multicast packets. When multicast packets are received from outside the MOSPF domain, MOSPF assumes those packets reach the inter-AS multicast forwarder through a Reverse Path Forwarding algorithm. The DVMRP also uses a Reverse Path Forwarding algorithm.

All inter-AS multicast forwarders declare themselves as wild-card multicast receivers in the backbone area. After reaching the backbone area, all multicast packets are required to reach all inter-AS multicast forwarders regardless of destination.

Multicast Routing Terms

The following terms are used in this chapter to explain multicast routing:

child link	The set of interfaces on which to forward multicast packets.
Core-Based Trees (CBT)	With this protocol, all the group members share a multicast delivery tree. CBT builds a multicast tree based on a core router instead of the source. It builds one multicast tree per group instead of one per (source, group) pair.
designated router	The router that is responsible for sending IGMP Host Membership Query datagrams. For DVMRP routers, the designated router is the router with the lowest IP address on the subnet. For MOSPF routers, the designated router is the router with the highest priority.
Distance Vector Multicast Routing Protocol (DVMRP)	This distance-vector protocol builds a shortest-path source-based delivery tree between each source (sender) and multicast group (receivers). It builds one multicast delivery tree per (source, group) pair. DVMRP has been implemented as a UNIX program on mrouted routers for several years.

dominant router	One of several routers on a link that is elected for a particular source. It is responsible for forwarding multicast datagrams on a subnet for a source.
downstream router	The router to which a multicast packet is forwarded.
forwarding table	A table containing group information (source and group pairs). This group information is applied to the routing table's shortest-path tree and helps the router forward multicast datagrams to each member of the group.
leaf link	A child link that no router uses to reach a source.
multicast	A technique that allows copies of a single packet to be passed to a selected subset of all possible destinations.
Multicast Open Shortest Path First (MOSPF)	This protocol is an extension to OSPF that builds a shortest-path source-based delivery tree on demand.
parent link	The expected interface that receives packets from a source; also the interface that leads to the previous-hop router back to the source.
Protocol Independent Multicast (PIM)	This protocol is not an extension of any unicast routing protocol; it relies on the unicast routing protocols and is suited for large heterogeneous internetworks. It contains two modes: Dense and Sparse. Dense mode uses a similar algorithm to the one used by DVMRP.
Reverse Path Broadcasting (RPB)	A refinement of the RPF algorithm that eliminates duplicate broadcast packets.
Reverse Path Forwarding (RPF)	An algorithm that forwards multicast datagrams by computing the shortest (reverse) path tree from the source network to all possible recipients. The router forwards the packet further only if it considers the sending router as the next-hop address of the source multicast.
Reverse Path Multicasting (RPM)	A refinement of the TRPB algorithm that provides on-demand pruning of the shortest-path multicast tree.
routing table	A table containing a list of routes that are learned from other multicast router's route report messages.
Truncated Reverse Path Broadcasting (TRPB)	A refinement of the RPB algorithm that only forwards packets to where they are wanted by pruning the shortest-path tree. This algorithm prunes branches of nonmember leaf networks.
upstream router	The router from which a multicast packet is received.

10 CONFIGURING APPN INTERMEDIATE SESSION ROUTING

This chapter describes how to configure your 3Com bridge/router to function as a network node in an Advanced Peer-to-Peer Networking (APPN) network.

APPN is an architecture designed to provide peer-to-peer routing services for Systems Network Architecture (SNA) environments. APPN is designed to work with Advanced Program-to-Program Communications (APPC) functions, and APPN uses LU 6.2 sessions to exchange network information between nodes. The 3Com implementation of APPN allows the bridge/router to function as a network node in an APPN network as well as serve as a Dependent LU Requester (DLUr) for relaying sessions with dependent logical units (LUs) on physical unit (PU) type 2.0 and 2.1 nodes.

Two types of APPN routing are available on the NETBuilder II bridge/router:

- Intermediate Session Routing (ISR)
- High Performance Routing (HPR)

This chapter describes how to configure your bridge/router for Intermediate Session Routing only. For information on how to configure your bridge/router as a network node for High Performance Routing, refer to Chapter 11.

CAUTION: *If you are upgrading from a previous APPN version, note that in version 9.0, HPR is enabled by default. If you want your existing network to perform ISR only, then you must disable HPR on your APPN ports and adjacent link stations.*

For conceptual information about APPN, refer to "How APPN ISR Routing Works" on page 10-43.

Setting Up a Basic APPN Router

The procedures in this section explain how to configure your bridge/router as a network node and configure node information to initiate APPN routing. The minimum tasks required to configure the APPN network node are separated into the following procedures:

- Setting up the bridge/router as a network node (also referred to as the *local node* when working directly at that node)
- Defining links from your system to adjacent network nodes
- Configuring dependent LU support if you have PU 2.0 nodes and PU 2.1 nodes with dependent LUs in your network
- Enabling the network node

Figure 10-1 provides a flowchart of the basic steps to configure the bridge/router so that it will operate as an APPN network node.

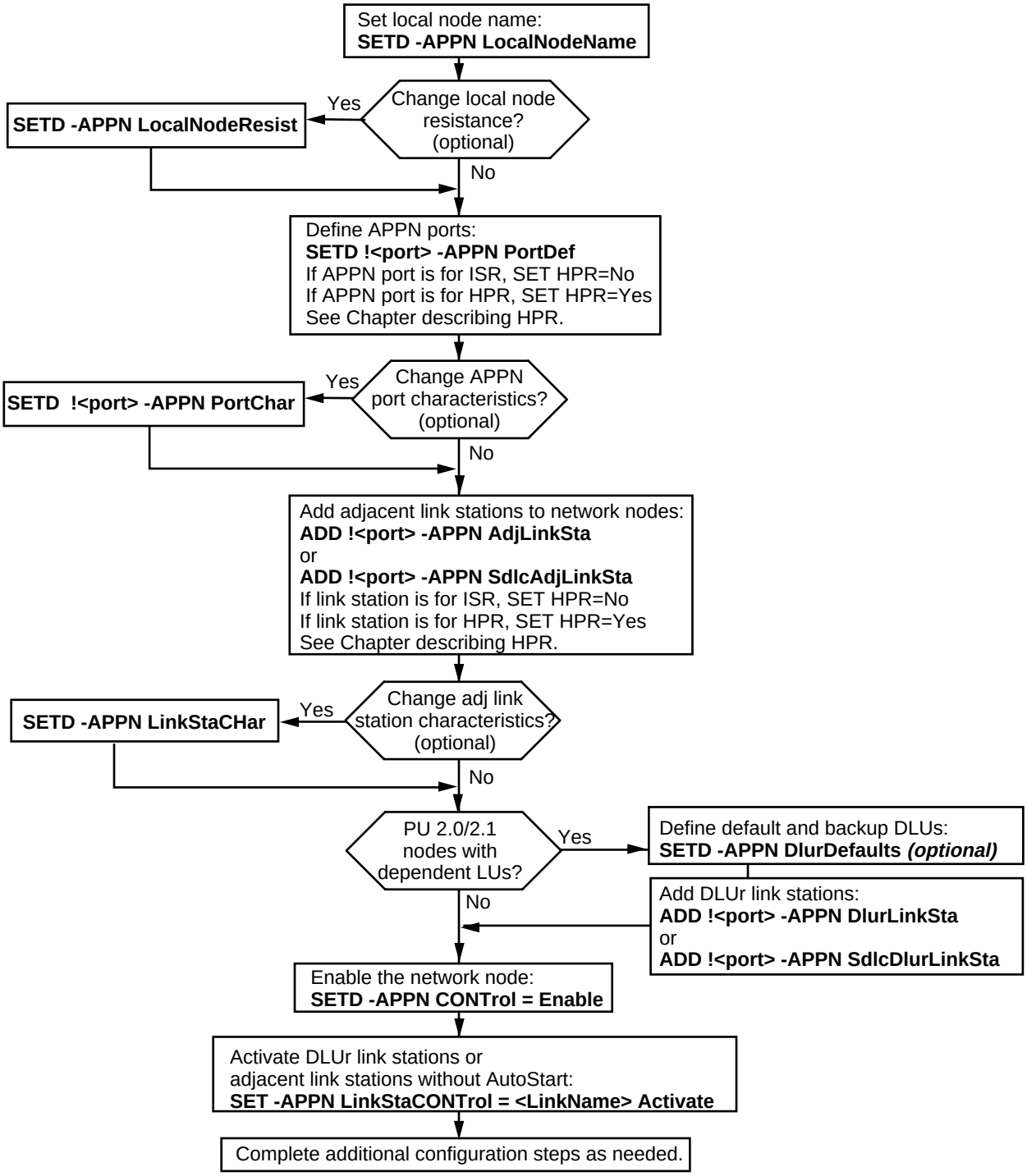


Figure 10-1 Basic APPN Configuration Steps

Setting Up Your System as a Network Node

The first task in setting up the APPN environment is to configure the local bridge/router (referred in this section as "local node") to serve as a network node. The NETBuilder II system can be configured as a network node only; because the bridge/router does not provide any application programs on the SNA network, it cannot act as an end node or LEN end node. Viewed from the SNA network, the bridge/router network node has only one LU for handling CP-CP sessions.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the procedures described in Chapter 1.
- If necessary, use LAN Address Administration (LAA) to reassign MAC addresses for paths that will be sending and receiving APPN traffic.

  You must perform this configuration *before* starting APPN. For more information on configuring LAA, refer to Chapter 28.
- If you are planning to support both APPN and DECnet on the same bridge/router, you must configure DECnet *before* configuring APPN. Configuring DECnet can change MAC addresses, which will affect any existing APPN configuration. For more information on configuring DECnet, refer to Chapter 15.
- If necessary, configure the Logical Link Control type 2 (LLC2) data link interface or the data link switching (DLSw) interface for the ports you will use for APPN traffic. For more information on configuring the LLC2 data link interface, refer to Chapter 21. For more information on configuring DLSw, refer to Chapter 24.

APPN is affected by parameter settings in other services. For more information, refer to "Configuring LLC2 with Other Services" on page 21-3.

- If you will be sending APPN traffic over SDLC lines, configure the bridge/router for SDLC operation first. For more information on SDLC configuration, refer to Chapter 22.
- If you will be sending APPN traffic over Frame Relay, configure the Frame Relay interface before configuring the APPN network node. For more information on configuring Frame Relay, refer to Chapter 42.
- If you are not familiar with APPN routing concepts, refer to "How APPN ISR Routing Works" on page 10-43.
- Refer to the IBM documents describing APPN architecture listed in "IBM APPN References" on page 10-54.

Procedure

To set up the bridge/router as a network node, follow these steps:

1 Assign a name to the local node using:

```
SETDefault -APPN LocalNodeName = <netid.cpname> [node_id]
```

This command creates the *fully qualified* control point (CP) name by combining the network ID with the CP name you create to identify the node. The fully qualified CP name identifies the network node throughout the APPN network. (When the CP name is used without the network ID, it is called a *not fully qualified* CP name.) For more information on CP name formats, refer to "Fully Qualified and Not Fully Qualified CP Name Formats" on page 10-50.

For example, to assign the local node name consisting of the network ID US3COMHQ plus the CP name NB2SF011, enter:

```
SETDefault -APPN LocalNodeName = US3COMHQ.NB2SF011
```

CAUTION: *Every fully qualified CP name on the APPN network must be unique.*

Optionally, you can add a node ID following the network ID. This node ID is used in XID negotiations. For more information, refer to the description of the LocalNodeName parameter in Chapter 5 in *Reference for NETBuilder Family Software*.

2 If desired, change the resistance value of the local node using:

```
SETDefault -APPN LocalNodeResist = <node_resistance> (0–255)
```

The resistance value advertises the desirability of routing through the node. Using different values, you can fine-tune your network to set different resistance rates on different nodes so that more traffic is routed over specific nodes.

The value of the LocalNodeResist parameter ranges from 0 to 255. A value of 0 indicates that routing is highly desirable through this node, while a value of 255 indicates routing is not desirable through the node. The default value is 128, or the median. Changing the value is optional.

3 Define each local port on the system that will send and receive APPN traffic using:

```
SETDefault !<port> -APPN PortDef = <DLC type>
   (LLC2|FR|PPP|DLSW|SDLC|UNdef) <max_btu_size>(99-8192)
   [ActLimit=<limit>(1-512)] [TGprof=<name>] [HPR=(Yes|No)]
   [ErrorRecovery=(Yes|No)] [DatMode=(Half|Full)]
   [ROle=(Neg|Pri|Sec)]]
```

Use this command to define the type of traffic being sent over the port (DLC type), as well as the maximum basic transmission unit (BTU) size the port will allow. To define the DLC type, enter LLC2 for token ring, Ethernet, FDDI and PPP links. Enter FR for Frame Relay, or DLSw for using Data Link Switching over an IP network. If you specify the DLC type as DLSw, the port number specified must be !0. Do not specify !0 if using a DLC type other than DLSw. Enter SDLC if you will be sending traffic to and from SDLC devices. Enter UNdef to remove a previously-defined port definition DLC type.

If a port has already been defined for a particular DLC type, the port definition must be removed by setting the DLC type to UNdef before it can be changed to another DLC type.

To determine the maximum BTU size to use, first determine the appropriate request/response unit (RU) size, then add an additional nine bytes (three bytes for the request header (RH) plus six bytes for the transmission header (TH)). The RU size plus the additional nine bytes comprise the BTU size. For more information on the values for the PortDef parameter, refer to Chapter 5 of *Reference for NETBuilder Family Software*.

For more information about setting the maximum BTU size, refer to "Setting the Maximum BTU Size" on page 10-51.

Optionally, you can set the activation limit (total number of LLC2 sessions for the port), and if desired, a transmission group (TG) profile for the port. For more information on TG profiles you can use, refer to the description of the AdjLinkSta parameter in Chapter 5 of *Reference for NETBuilder Family Software*.

CAUTION: *The PortDef parameter has an option to provide support for High Performance Routing. The default value for the HPR option is Yes, meaning that HPR is automatically enabled. If you want the port to perform Intermediate Session Routing (ISR) only, you must disable the HPR option by typing HPR=No as part of the command. If you want the port to perform HPR, do not change the HPR value, but note that the functionality and routing methods of HPR may*

be different from ISR. If you have links between two network nodes with HPR enabled, this configuration will create an HPR subnet in your ISR network. For more information about HPR, refer to Chapter 11.

For example, to configure port 7 as an APPN ISR port to handle Frame Relay traffic with a maximum BTU size of 1033, an activation limit of 128, and to use the TG profile SER256, enter:

```
SETDefault !7 -APPN PortDef = FR 1033 ActLimit=128 TGprof=SER256
   HPR=No
```

If you specify synchronous data link control (SDLC) as your DLC type, you can specify the DatMode value to either half duplex or full duplex, and you can specify whether the SDLC port will be the primary or secondary device in session negotiation, or whether the role will be negotiable. If you set your DLC type to SDLC, when configuring SDLC devices as adjacent link stations or as DLUr link stations you must use the SdlcAdjLinkSta or SdlcDlurLinkSta parameters, respectively.

For example, to configure port 6 as an APPN ISR port to handle SDLC traffic you can set the following attributes: maximum BTU size of 1033, activation limit of 254, TG profile of Ser19.6, and full duplex data transmission mode. To configure these attributes and set the local node as the primary device in session negotiation, enter:

```
SETDefault !6 -APPN PortDef = SDLC 1033 ActLimit=254
   TGProf=Ser19.2 HPR=No DatMode=Full ROle=Pri
```

Repeat this step for each port on the system used to send and receive APPN sessions.

4 If desired, define the characteristics of each APPN port configured in the previous step using:

```
SETDefault !<port> -APPN PortCHar = [EffectCap=<string>]
   [ConnectCost=<0-255>] [ByteCost=<0-255>] [Security=<string>]
   [PropDelay=<string>] [Usd1=<0-255>] [Usd2=<0-255>]
   [Usd3=<0-255>]
```

Using this parameter, you can specify optional settings for the port's effective capacity, connection cost, byte cost, propagation delay, and three user-configurable settings. For more information on the PortCHar parameter, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

5 Repeat this procedure for each bridge/router functioning as a network node in your APPN network.

After you have set up the bridge/router as a network node, you must then define the links to other network nodes in the APPN network. Proceed to the next section.

Defining Links to Other Network Nodes

After you have performed the basic configuration of the local node, the next step is to define the adjacent link stations to other network nodes. An *adjacent link station* is the local information regarding a link to an adjacent node. The adjacent link station is the link definition, or the representation of the link as seen by the network node.

Two network nodes that connect and exchange data are called *partner nodes*. To configure an adjacent network node as a partner node, you must configure

an adjacent link station to the other node; in this situation, the other network node does not need to configure an adjacent link station to your local node. Only one of the partner nodes needs to configure the other as an adjacent link station.

You can add links to other network nodes dynamically after the network node is enabled. For more information on dynamic configuration, refer to "Dynamic Configuration Options" on page 10-14.

Figure 10-2 is an example of a network with three different network nodes, each with its own local network, on a larger FDDI ring. In this topology, network nodes A and B are partner nodes to each other, network nodes A and C are partner nodes, and network nodes B and C are partner nodes.

For each of these partner node pairs, only one network node needs to configure its partner as an adjacent link station if both nodes are NETBuilder II bridge/routers. If one of the partner nodes is not a NETBuilder II bridge/router, the links may need to be configured in both directions, depending on the device.

For example, if network node A configures node B as an adjacent link station, then network node B does not also need to configure node A as an adjacent link station. If both partner nodes are 3Com bridge/routers, this situation applies. You can configure links in both directions, but it is not required.

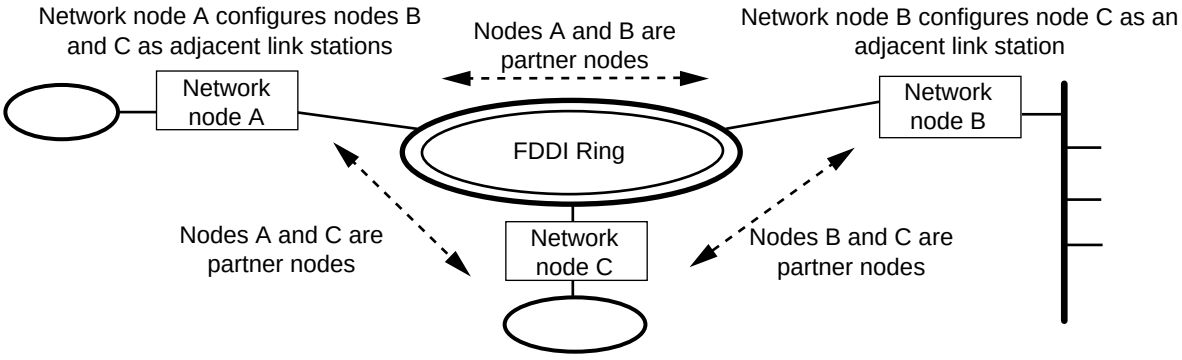


Figure 10-2 Network Nodes as Adjacent Link Stations (Example)

Procedure

To define adjacent link stations to partner network nodes, follow these steps:

1. If you set the port DLC type (configured with step 3 of the previous procedure, on page 10-4) to LLC2, FR, PPP, or DLSw, go to step a. If you set the port DLC type to SDLC in the previous procedure, go to step b on page 10-8.

   a. If you previously set the port DLC type to LLC2, FR, PPP, or DLSW, define the adjacent link station using:

```
ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn)
<max_btu_size>(99-8912) [[Cmac|Ncmac] dest media addr]
[Sap=<num>] [CPName=[netid.]cpname] [Nodeid=<ID>]
[LinkName=<name>] [TGprof=<name>] [AutoStart=(Yes|No)]
[CPSess=(Yes|No)] [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

Make sure you specify the node type as NN. In addition, specify the maximum BTU byte size and the media address of the destination node (or DLCI if running Frame Relay over a virtual port). Optionally, you can set the following for the destination node: the CP name and the node name, the node ID, the link name, the TG profile, whether the link will support AutoStart, and whether control point-to-control point (CP-CP) sessions will be activated with the adjacent node. For more information on the AdjLinkSta parameter, refer to Chapter 5 in *Reference for NETBuilder Family Software.*

CAUTION: *The AdjLinkSta parameter has an option to provide support for High Performance Routing. The default value for the HPR option is Yes, meaning that HPR is automatically enabled. If you want the link station to support Intermediate Session Routing (ISR) only, you must disable the HPR option by typing HPR=No as part of the command, and you also must disable HPR on the port by specifying HPR=No as part of the SETDefault !<port>-APPN PortDef command. If you only disable HPR on the adjacent link station but not the port, then HPR will not be totally disabled for APPN connections. If you want the link station to support HPR, do not change the HPR value, but note that the functionality and routing methods of HPR may be different from ISR. For more information about HPR, refer to Chapter 11.*

If you do not define a link name, then the local network node will assign a unique link name to the link. (You will need the link name to complete step 2. If you do not assign a link name, you can obtain the link names assigned by the system using the SHow -APPN LinkStaCONTrol command.)

For example, to add a link to an ISR network node named "FINANCE" to port 3 with a maximum BTU size of 1033 (specifying the appropriate MAC address and SAP) and a fully-qualified CP name "HQ.Finance" (with a link named FINANCE3), profile SER64, and to activate a CP-CP session when the node comes up, enter:

```
ADD !3 -APPN AdjLinkSta NN 1033 N100040C08ACE Sap=08
   CPName=HQ.FINANCE LinkName=FINANCE3 TGprof=SER64 CPSess=Yes
   HPR=No
```

For information on how to obtain the MAC address of the node, see the documentation for the end node device or applications.

To obtain the MAC address of another 3Com bridge/router acting as a network node, enter the SHow -SYS Configuration command on the second bridge/router. Enter the MAC address of the port number over which the link is established, making sure to enter the address in the correct format.

If you set the -SYS MacAddrFmt parameter to noncanonical, then you do not need to precede the MAC address with N or Ncmac. If you do not change the -SYS MacAddrFmt parameter, then the default will be canonical, and you will need to precede the MAC address with N for noncanonical format. If the -SYS MacAddrFmt parameter is set to Default, then the system will assume that the MAC address is in noncanonical format for token ring and FDDI ports, and canonical format for all other port types. For more information on MAC address format options for APPN, refer to "MAC Address Format Options for APPN" on page 10-51.

b If you previously set the port DLC type to SDLC, define the SDLC adjacent link station using:

```
ADD !<port> -APPN SdlcAdjLinkSta <type>(NN|EN|Learn)
<max_btu_size>(99-8912) <station addr>(Hex 1-FE)
[CPName=<[netid.]cpname] [Nodeid=<ID>] [LinkName=<name>]
[TGprof=<name>] [AutoStart=(Yes|No)] [CPSess=(Yes|No)]
[HPR=(Yes|No)] [ErrorRecovery=(Yes|No)] [SendWindow=<num>]
[ContactTimer=<num>] [NoRspTimer=<num>]
[NoRspTimRetry=<num>]
```

Make sure you specify the node type as NN. In addition, specify the maximum BTU byte size and the station address of the destination node. Optionally, you can set the CP name of the destination node and the node name, the node ID, the link name, the TG profile, whether the link will support AutoStart, and whether CP-CP sessions will be activated with the adjacent node. You can also set the SDLC SendWindow, ContactTimer, NoRspTimer, and NoRspTimRetry values. You can enter these options in any combination. The default value for AutoStart is yes, which means when you enable the network node, the link will be activated automatically. For the SDLC connection to take place, both SDLC partner nodes must be configured as SDLC adjacent link stations using the SdlcAdjLinkSta parameter.

For more information on the SdlcAdjLinkSta parameter, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

CAUTION: *The SdlcAdjLinkSta parameter has an option to provide support for High Performance Routing. The default value for the HPR option is Yes, meaning that HPR is automatically enabled. If you want the link station to support Intermediate Session Routing (ISR) only, you must disable the HPR option by typing HPR=No as part of the command, and you must also disable HPR on the port by specifying HPR=No as part of the SETDefault !<port>-APPN PortDef command. If you only disable HPR on the adjacent link station but not the port, then HPR will not be totally disabled for SDLC connections. If you want the link station to support HPR, do not change the HPR value, but note that the functionality and routing methods of HPR may be different from ISR. Note also that for HPR over SDLC to work properly, HPR must be configured on both partner network nodes. For more information about HPR, refer to Chapter 11.*

If you do not define a link name, then the local network node will assign a unique link name to the link. (You will need the link name to complete step 2. If you do not assign a link name, you can obtain the link names assigned by the system using the SHow -APPN LinkStaCONTrol command.)

For example, to add an SDLC link named "SDLC001" on port 4 to a network node named "HQ.FINANCE" you can set the following attributes: a station address of hex FE, maximum BTU size of 1033, TGprofile SER64, activation of a CP-CP session when the node comes up, no support for HPR, SendWindow size of 4, ContactTimer setting of 2 seconds, NoRspTimer setting of 2000 milliseconds, and a NoRspTimRetry setting of 6. To add this link and configure the attributes, enter:

```
ADD !4 -APPN SdlcAdjLinkSta NN 1033 FE CPName=HQ.FINANCE
LinkName=SDLC001 TGprof=SER64 CPSess=Yes HPR=No SendWindow=4
ContactTimer=2 NoRspTimer=2000 NoRspTimRetry=6
```

The ContactTimer, NoRspTimer and NoRspTimRetry values are valid only if the local network node is the primary station on the SDLC link. Also, The SDLC link must be configured before configuring APPN over SDLC. For more information on SDLC, refer to Chapter 22.

APPN over SDLC connections is supported on all types of HSS 3-Port modules, including V.35, RS-232, and RS-449.

When you configure SDLC adjacent link stations for APPN, if an active link becomes inactive and you change the port definition using the PortDef parameter, the link remains inactive. If you try to reactivate the link using the SET -APPN LinkStaCONTrol command, the link will reactivate within 30 seconds. To activate the link immediately, you must enable the APPN port using the SET -APPN PortControl = Enable command.

2 After you have defined the link to the adjacent network node, you define the characteristics of the link using:

```
SETDefault -APPN LinkStaCHar = <LinkStation name>
   [EffectCap=<string>] [ConnectCost=<0-255>] [ByteCost=<0-255>]
   [Security=<string>] [PropDelay=<string>] [Usd1=<0-255>]
   [Usd2=<0-255>] [Usd3=<0-255>]
```

Set attributes such as byte cost, security, connection cost, and capacity for the adjacent link station with the LinkStaCHar parameter. You can set any number of these options in any combination when entering the command. For more information on configuring this parameter, refer to Chapter 5 of *Reference for NETBuilder Family Software*.

For example, to define the characteristics of the link named "FINANCE3" for an effective capacity of 9600, a byte cost of 128, and a security value of SECurcnd, enter:

```
SETDefault -APPN LinkStaCHar = FINANCE3 EffectCap=9600 ByteCost=128
   Security=SECurcnd
```

CAUTION: *If you change any of the default characteristics for a link to a network node, the characteristic must also be changed on the partner network node. For example, if you set the security level of the TG as GUarded on the local node, then you must also configure the security level as GUarded on the partner node. Otherwise, the characteristic will be valid in one direction only, from the local node to the partner node; the characteristic on the link in the opposite direction will not match.*

3 Repeat steps 1 and 2 for each network node that will establish direct connections (or links) with the local network node.

If you did not assign link names using the AdjLinkSta parameter, the system will assign them. To obtain a list of link names assigned, enter:

```
SHow -APPN LinkStaCONTrol
```

You can configure two or more links to the same node using parallel TGs. For more information on configuring parallel TGs, refer to "Configuring Parallel Transmission Groups" on page 10-24.

If you need to configure support for dependent LUs, proceed to the next section. If you do not need to do so, proceed to "Enabling the Network Node and Activating Links" on page 10-13.

Configuring Dependent LU Support

Dependent logical unit support is required where you have PU type 2.0 or 2.1 nodes in the local network node's domain that will access a host via LU types dependent on the SSCP. LU types that are dependent on a Session Services Control Point (SSCP) are types 1, 2 ,3, or type 6.2. Configuring dependent LU support on the network node enables the network node to act as a Dependent LU Requestor (DLUr) to enable a PU type 2.0 or 2.1 node to access the host, which acts as the Dependent LU Server (DLUs). You can have many PUs with dependent LUs accessing one primary DLUs and one backup DLUs.

PU type 2.0 nodes are nodes which do not have a control point. As a result, LUs on these nodes are "dependent" on SSCP services provided by the DLUs. PU type 2.1 nodes can have both independent and dependent LUs. The dependent LUs require the SSCP services from the host, while independent LUs do not.

Figure 10-3 is an example of PU type 2.0 and 2.1 nodes accessing a host DLUs with a bridge/router acting as the DLUr. In the configuration, the DLUs is *upstream* from the network node bridge/router, while the PU 2.0 and 2.1 nodes are *downstream* from the network node.

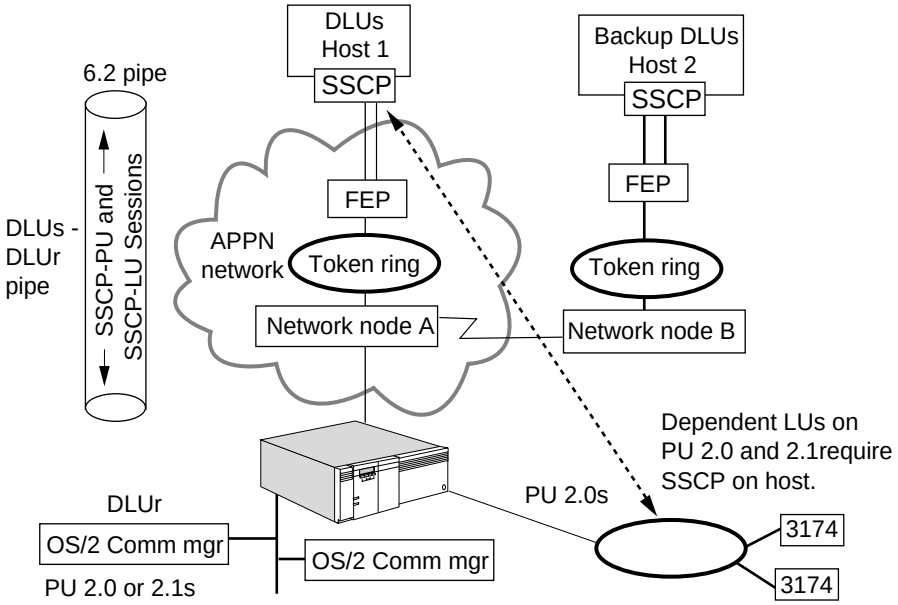


Figure 10-3 DLUr and DLUs Environment

This section is divided into two procedures:

- Defining your DLUs
- Configuring links to nodes requesting DLUr services

If the DLUs will be accessed over a WAN using Frame Relay, you also will need to configure the APPN Frame Relay interface.

Figure 10-4 is an example of a DLUr and DLUs configuration. In the configuration, the downstream physical unit (DSPU) defined in the host configuration must match the DSPU name configured on the network node using the DlurLinkSta parameter. For the PU 2.x to access the host, the MAC address of the local node must be configured as the host address on the PU. The PU thinks the host address is for the remote host providing the service, but

the network node address is used to establish the session to the network node. The network node then establishes the SSCP-LU and SSCP-PU sessions with the host.

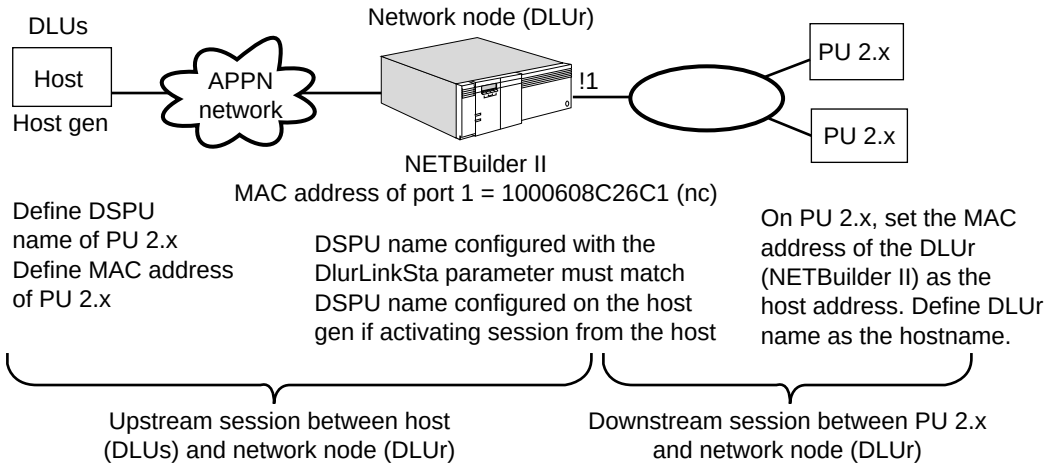


Figure 10-4 DLUs and DLUr Configuration

Defining the Default DLUs and Backup DLUs

When you define the DLUs on the network node, you are configuring the default DLUs and backup DLUs that the local node (acting as the DLU requestor) will send the SSCP traffic to. The DLUs does not need to be directly connected to the local network node, and there can be multiple network nodes in between.

When a dependent LU makes a session request to the local network node for a dependent LU server, the local node tries to find the DLUs using the following hierarchy of steps:

- The system first looks for the DLUs assigned to the DLUr link station using the DlurLinkSta parameter (refer to "Defining Downstream Links to Nodes with Dependent LUs" on page 10-12).
- If that DLUs is unavailable or no DLUs was assigned to the DLUr link station, then the system tries to use the backup DLUs assigned using the DlurLinkSta parameter.
- If the backup DLUs is unavailable or no backup DLUs was assigned to the DLUr link station, then the local node tries the default DLUs configured using the DlurDefaults parameter.
- If the default DLUs is unavailable, then the local node tries the default backup DLUs configured using the DlurDefaults parameter.

To configure the default DLUs and backup DLUs, use:

```
SETDefault -APPN DlurDefaults [Dlus=(<name>|UNdef)]
   [Backup=(<name>|UNdef)]
```

Using this command, you specify the default DLUs and the backup DLUs. You can configure one default DLUs and one default backup DLUs on the local network node.

For example, to configure a primary DLUs named "VTAM1" and a backup DLUs named "VTAM2," enter:

```
SETDefault -APPN DlurDefaults = DLUS=VTAM1 BACKUP=VTAM2
```

To change the name of a primary or backup DLUs, repeat the command and enter a different name. To remove the name of a primary or backup DLUs, enter the command but specify "UNdef." For example, to remove VTAM2 as the backup DLUs, enter:

```
SETDefault -APPN DLurDefaults = BACKUP=UNdef
```

Defining Upstream Links for Path to DLUs

You can have any number of intermediate network nodes in your APPN network between the local network node DLUr and the DLUs host. To define the upstream link for the path to the DLUs, you configure the upstream network node as a normal adjacent link station. No special configuration is required. The only requirement is that you must be able to establish 6.2 LU to LU sessions between the local network node DLUr and the DLUs host.

Defining Downstream Links to Nodes with Dependent LUs

If you have PU 2.0 nodes or PU 2.1 nodes with dependent LUs in the network node domain, then you must configure DLUr link stations to each of these nodes. Because these nodes function differently from normal APPN nodes, you cannot configure DLUr link stations and normal adjacent link stations to the same node. However, a node can have CP-CP sessions and still require DLUr. If that is the case, add these nodes using this procedure.

To add a link to PU 2.0 and 2.1 nodes that require DLUr services, follow these steps:

1. Select one of the following:
   a. If you are running normal APPN traffic to and from DLUr link stations, define each DLUr link station using:

      ```
      ADD !<port> -APPN DlurLinkSta <max_btu_size(256-8912)> <[Cmac |
      Ncmac] dest media addr> <dspu name> [Sap=<num>] [Nodeid=<ID>]
      [LinkName=<name>] [Dlus=<[netid.]name|UNdef>]
      [Backup=<[netid.]name|UNdef>] [TGprof=<name>]
      [AutoStart=(Yes|No)] [CPSess=(Yes|No)] [PU2=(Yes|No)]
      [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
      ```

      Using this command, you specify the maximum BTU size, the destination address of the DLUr link station, and the DSPU name of the PU 2.0 device. If the host will activate the session with the DLUr link station, then the DSPU name you configure here must match the name on the host configuration.

      You also specify the primary DLUs and backup DLUs that the DLUr link station will access. If a primary and/or backup DLUs is not specified, then the default primary and backup DLUs configured using the DlurDefaults parameter will be used. The default for AutoStart is No. If you want the link to automatically be activated when the network node is enabled, specify AutoStart=Yes.

b If you set the port DLC type to SDLC to run SDLC traffic to and from DLUr link stations, define each SDLC DLUr link station using:

```
ADD !<port> -APPN SdlcDlurLinkSta <max_btu_size>(265-8912)
<station addr>(Hex 1-FE) <dspu name> [Nodeid=<ID>]
[LinkName=<name>] [Dlus=[netid.]name] [Backup=[netid.]name]
[TGprof=<name>] [AutoStart=(Yes|No)] [PU2=(Yes|No)]
[HPR=(Yes|No)] [CPSess=(Yes|No)] [SendWindow=<num>]
[ContactTimer=<num>] [NoRspTimer=<num>]
[NoRspTimRetry=<num>]
```

Using this command, you specify the maximum BTU size, the destination address of the DLUr link station, and the DSPU name of the PU 2.0 device. If the host will activate the session with the DLUr link station, then the DSPU name you configure here must match the name on the host configuration.

You also specify the primary DLUs and backup DLUs that the DLUr link station will access. If a primary and/or backup DLUs is not specified, then the default primary and backup DLUs configured using the DlurDefaults parameter will be used. You can also specify SDLC attributes such as the SendWindow, ContactTimer, NoRspTimer, and NoRspTimRetry values. For more information on these values, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

APPN over SDLC connections is supported on all types of HSS 3-Port modules, including V.35, RS-232, and RS-449.

2 Repeat the previous step for each PU 2.0 or 2.1 node that will access a DLUs through the local network node.

Using VTAM Program Temporary Fixes

VTAM Program Temporary Fixes (PTFs) are required on a mainframe when APPN DLU services are used. Mainframe network management (NetView) services will not function for downstream physical units (PUs) if the PTFs are not installed. VTAM Version 4.2 requires PTF #UW20787. VTAM Version 4.3 requires PTF #UW20788.

Symptoms of this problem result from a lack of network management data for PUs that are downstream of a NETBuilder II using APPN DLU services. The NetView message "AAU251I AAUDRTIB 02 UNEXPECTED SENSE CODE X'1002' ENCOUNTERED FOR TARGET=pu_name" is printed in the log file when this problem occurs.

Enabling the Network Node and Activating Links

After you have set up the bridge/router as a network node and defined links to other network nodes you can now enable the network node and activate the links you defined in the previous sections.

To enable the network node and activate the links, follow these steps:

1 To enable the bridge/router to function as an APPN network node, enter:

```
SETDefault -APPN CONTrol = Enable
```

When you enable the APPN network node, you will receive a message similar to the following:

```
Wed Dec 31 16:11:15 1995 LOCAL NETWORK NODE US3COMHQ.GOLD IS
STARTED
```

After the network node is enabled, the bridge/router can communicate with other APPN network nodes, and can accept incoming link requests from end nodes.

You can totally disable the network node, or you can dynamically disable the network node so that when you reboot the bridge/router, the network node automatically is re-enabled. For more information on disabling the network node, refer to "Disabling the Network Node" on page 10-34.

2 If you configured adjacent link stations and you set AutoStart to No or configured DLUr link stations and did not set AutoStart to Yes, activate these links using:

```
SET -APPN LinkStaCONTrol = <LinkName> Activate
```

Repeat this step for each of the links you defined in the previous sections. After you have enabled the network node and activated your basic links, the basic network node will be operating. Other network nodes will be able to initiate sessions with the local node and receive sessions from the local node. In addition, end nodes in the local node's domain will be able to initiate session requests with the network node.

For additional configuration, refer to "Customizing the APPN Router" on page 10-18.

Dynamic Configuration Options

After the network node is enabled, you can configure different options such as adjacent link stations, transmission group (TG) characteristics, and port characteristics. Depending on the task, you can configure these options without disabling the network node or disrupting sessions on ports or TGs not affected. Table 10-1 lists some of the APPN entities that you can and cannot dynamically configure while the network node is operating.

Table 10-1 APPN Dynamic Configuration Options

Configuration Option	Parameter	Dynamic Configuration Allowed	Additional Information
Predefine LEN end node LUs	AdjLenDef	Yes	
Add or delete adjacent link stations	AdjLinkSta	Yes	Port the link station is mapped to can be enabled while configuring. Must activate link using LinkStaCONTrol parameter to take effect. To delete link station, must deactivate it first.
Adjacent link station characteristics	LinkStaCHar	Yes	Cannot make changes if link is active. You must first deactivate the link and then reactivate it after making the change.
Create a customized class of service, and change node row and TG row values	ConfigCos COSNodeRow COSTgRow	Yes	Refer to Chapter 12 for more information.
Enable connection network	ConnNetworkDef	Yes	The port the connection network is added to can be enabled when configuring.
Define a customized class of service to the system	CosDef	Yes	Refer to Chapter 12 for more information.
Add or delete directory entries	DirectoryEntry	Yes	
Set the default DLUs and backup DLUs	DlurDefaults	Yes	
Define DLUr link stations	DlurLinkSta	Yes	If the link is active, you cannot make changes. Deactivate the link before making changes.

(continued)

Table 10-1 APPN Dynamic Configuration Options (continued)

Configuration Option	Parameter	Dynamic Configuration Allowed	Additional Information
Activate and deactivate link stations	LinkStaCONTrol	Yes	
Set the local node name and resistance	LocalNodeName LocalNodeResist	No	Must be configured before enabling the network node.
Map mode names to a class of service	ModetoCosMap	Yes	Refer to Chapter 12 for more information.
Change APPN port characteristics and define the APPN port	PortCHar PortDef	Yes	If port is activated, must first deactivate the port using PortCONTrol parameter before changing characteristics or definitions. Port must then be reactivated after making changes.
Activate and deactivate APPN port	PortCONTrol	Yes	
Set queue priority	QueuePriority	Yes	Refer to Chapter 41 for more information.
Add or delete adjacent SDLC link stations	SdlcAdjLinkSta	Yes	Port the link station is mapped to can be enabled while configuring. Must activate link using LinkStaCONTrol parameter to take effect. To delete link station, must deactivate it first.
Define SDLC DLUr link stations	SdlcDlurLinkSta	Yes	If the link is active, you cannot make changes. Deactivate the link before making changes.

Configuring the APPN Router for Wide Area Networks

To configure your APPN router to perform routing over Frame Relay, refer to Chapter 42. APPN routing over SMDS and X.25 is not supported unless you are using DLSw. For information on routing over PPP connections, refer to Chapter 34. For information on wide area networking using ISDN, refer to Chapter 35. For more information on data link switching, refer to Chapter 24.

APPN routing over ATM is not supported.

Verifying the APPN Router Configuration

To verify that the APPN router you configured is recognized by the APPN network and is receiving incoming session requests, follow these steps:

1. Display information on ports configured for APPN using:

   ```
   SHow [!<port>] -APPN PortDef
   ```

2. Verify that the ports configured for APPN are active using:

   ```
   SHow [!<port>] -APPN PortCONTrol
   ```

   If a port is shown as "Not Defined" in the display, that indicates the port was not defined as an APPN port using the SETDefault !<port> -APPN PortDef command.

3. Verify the local node name assigned to the APPN router by entering:

   SHow -APPN LocalNodeName

   Note the local node name so you will recognize it in displays later in this procedure.

4. Verify the adjacent link stations the APPN router is linked to by entering one or both of the following commands:

   SHow -APPN AdjLinkSta
   SHow -APPN SdlcAdjLinkSta

   For more information about this display, refer to "Adjacent Link Station Information" on page 10-39.

5 Verify whether links to adjacent link stations and DLUr link stations are active by entering:

```
SHow -APPN LinkStaCONTrol
```

For more information about this display, refer to "Current Status of Link Stations" on page 10-41.

6 Verify information for all adjacent network nodes the APPN router can communicate with by entering:

```
SHow -APPN NNtopology
```

For more information about this display, refer to "Network Topology Information" on page 10-38.

7 Verify information for the number and status of all adjacent nodes the APPN router is communicating with by entering:

```
SHow -APPN AdjNodeStatus
```

The display shows the number of adjacent nodes, including adjacent nodes, and end nodes in the network node's domain, and characteristics for those nodes.

8 Verify that the APPN router is sending and receiving connections to other nodes and the status of those connections by entering:

```
SHow -APPN CONNection
```

For more information about this display, refer to "Active APPN Connections" on page 10-40.

9 Verify that LUs on other nodes are getting registered into the local node's directory by entering:

```
SHow -APPN DIRectory
```

For more information about this display, refer to "APPN Directory Information" on page 10-38.

10 Verify that the APPN router is handling intermediate session routing, and verify the status of any ISR sessions by entering:

```
SHow -APPN ISRsessions
```

For more information about this display, refer to "Intermediate Session Routing Information" on page 10-41.

11 To display the status of all DLU servers that the local node has 6.2 sessions with, enter:

```
SHow -APPN DluSStatus
```

12 To display a list of DLUr link stations, enter one or both of the following commands:

```
SHow -APPN DlurLinkSta
SHow -APPN SdlcDlurLinkSta
```

13 To display a list of downstream PUs, enter:

```
SHOw -APPN DluRStaus
```

14 To display a list of downstream LUs, enter:

```
SHow -APPN DownStreamLU
```

15 Verify link activity for the node by entering:

```
SHow -APPN AppnLOG
```

Troubleshooting the APPN Router

If the APPN router is not properly communicating with other nodes in the network, review the following procedure. For more information regarding APPN Service parameters, refer to Chapter 5 of *Reference for NETBuilder Family Software*.

You can troubleshoot problems on an APPN network by following one or more of these steps:

1. Show the version of the software by entering:

   SHow -SYS VERsion

2. Show the path configuration by entering:

   SHow -PAth CONFiguration

3. Show the port configuration by entering:

   SHow -POrt CONFiguration

4. Show the system configuration by entering:

   SHow -SYS CONFiguration

5. Show the APPN configuration by entering:

   SHow -APPN CONFiguration

6. Check the status of APPN ports by entering:

   SHow -APPN PortCONTrol

7. Check the status of adjacent link stations by entering:

   SHow -APPN LinkStaCONTrol

8. Check the status of active connections by entering:

   SHow -APPN CONNection ALL

9. Check the status of adjacent nodes by entering:

   SHow -APPN AdjNodeStatus

10. Check the status of ISR sessions by entering:

    SHow -APPN ISRsessions

11. Check the status of transmission groups by entering:

    SHow -APPN TG ALL

12. If you cannot reach a specific LU in the APPN network, determine if a route exists between the local node and the LU using:

    ```
    APpnPING [netid.]<partner_lu_name> [Mode=modename] [Size=N]
       [Consec=N] [Iterations=N] [Echo=Yes|No] [Userid=<string>
       [Password=<string>]]
    ```

    The APpnPING command performs an APPC Ping to the other LU in the network. For more information on using the APpnPING command, refer to Chapter 1 of *Reference for NETBuilder Family Software*.

13. Check the current status of LLC2 sessions by entering:

    SHow -LLC2 SESSions

14. Check the current statistics for LLC2 sessions by entering:

    SHow -SYS STATistics -LLC2

15. Perform an analyzer trace on the LLC2 LAN links.

16. Perform an analyzer trace on the PPP WAN links.

Customizing the APPN Router

After you have configured the local network node, the network node will operate as an APPN router, communicating with other network nodes and accepting incoming session requests from end nodes. You can customize the APPN router for greater control and security by performing the following tasks:

- Statically defining links (adjacent link stations) to end nodes
- Statically defining entries into the network node's directory

You also can customize the APPN router by configuring the following items:

- Links to connection networks
- Parallel TGs
- Data link switching (DLSw) between nodes
- APPN and Boundary Routing environments

Defining Links to End Nodes

You normally do not have to define links (adjacent link stations) to end nodes. In APPN, end nodes make a link request to a network node to access the network. When a network node provides routing and topology services for an end node, the network node is called the *network node server* for the end node. End nodes can have links to more than one network node at a time, but only one network node can be the network node server to that end node at one time.

Because end nodes make incoming link requests to the network node, the process is dynamic, meaning end nodes can link to one network node for a certain time, then break the link and link to another network node for a different session request. As a result, it may not be practical to statically define links to end nodes if you have different network nodes that can serve as network node servers. If you have many end nodes, statically defining links for each one may not be practical.

You may want to statically define links to end nodes if you have a secure environment or want greater control over the network.

The procedure to define links to end nodes in your network node domain is similar to the procedure used to define links to other network nodes. Low-entry networking (LEN) end nodes are a subset of end nodes, and you define links to LEN end nodes the same way. However, if the LEN end node has more than one LU, then you need to statically predefine these LUs; for more information, refer to "Preconfiguring LEN End Node LUs" on page 10-21.

To define links to end nodes in your network node domain, follow these steps:

1. Define the link to an end node on a port and specify the node type as EN using:

```
ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn)
   <max_btu_size>(99-8912) [[Cmac|Ncmac] dest media addr]
   [Sap=<num>] [CPName=[netid.]cpname] [Nodeid=<ID>]
   [LinkName=<name>] [TGprof=<name>] [AutoStart=(Yes|No)]
   [CPSess=(Yes|No)] [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

   or, if running SDLC traffic on the port:

```
ADD !<port> -APPN SdlcAdjLinkSta <type>(NN|EN|Learn)
   <max_btu_size>(99-8912) <station addr>(Hex 1-FE)
   [CPName=[netid.]cpname] [Nodeid=<ID>] [LinkName=<name>]
   [TGprof=<name>] [AutoStart=(Yes|No)]
   [CPSess=(Yes|No)][HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
   [SendWindow=<num>] [ContactTimer=<num>] [NoRspTimer=<num>]
   [NoRspTimRetry=<num>]
```

APPN over SDLC connections is supported on all types of HSS 3-Port modules, including V.35, RS-232, and RS-449.

In addition to the adjacent link station's node type, you specify the maximum BTU size, and the destination media address control (MAC) address for non-SDLC traffic, or the destination station address for SDLC traffic. Optionally, you can set the node's CP name, node ID, link name, TG profile, whether auto startup will be supported, and whether the link will support CP-CP sessions with the adjacent node. The default for end nodes is to support CP-CP sessions. For non-SDLC traffic, you can set the node's Service Access Point (SAP) number. For SDLC traffic, you can set SDLC attributes such as SendWindow, ContactTimer, NoRspTimer and NoRspTimRetry. If the adjacent link station will not support HPR, make sure to specify HPR=No to turn off HPR support.

For example, to add a link to an end node in an ISR network named "ENGREEN" to port 3 with a maximum BTU size of 1033 (specifying the appropriate MAC address and not fully qualified CP name), and to specify the link will support auto startup, enter:

```
ADD !3 -APPN AdjLinkSta EN 1033 N100040C08ACE Sap=08
CPName=ENGREEN AutoStart=Yes HPR=No
```

For information on how to obtain the MAC address of a node, see the documentation for the end node device or applications. Most SNA and token ring environments use noncanonical MAC address formats. To convert a MAC address to canonical format, use the MacAddressConvert command.

If you set the -SYS MacAddrFmt parameter to noncanonical, then you do not need to precede the MAC address with N or Ncmac.

2 After you have defined the link to the end node, define the link characteristics using:

```
SETDefault -APPN LinkStaCHar = <LinkStation name>
   [EffectCap=<string>] [ConnectCost=<0-255>] [ByteCost=<0-255>]
   [Security=<string>] [PropDelay=<string>] [Usd1=<0-255>]
   [Usd2=<0-255>] [Usd3=<0-255>]
```

With this command, you set attributes such as byte cost, security, connection cost, and effective capacity for the adjacent link station. For more information on configuring this parameter, refer to the description of the LinkStaCHar parameter in Chapter 5 of *Reference for NETBuilder Family Software*.

3 Repeat steps 1 and 2 for each end node (or LEN end node) that you will allow to link directly with the local network node.

Defining Links to Unknown Node Types

You may not know if a node is a network node or an end node, or know the node name or CP name. To define an adjacent link station to an unknown type of node, enter the ADD -APPN AdjLinkSta command or the ADD -APPN SdlcAdjLinkSta command and specify the node type as LEARN. If you specify LEARN, the system learns the node type as well as other information such as the node name and CP name. To add a link station to a node whose node type

is learned, you must at least know the MAC address of the node. To add an SDLC link station to a node whose type is learned, you must at least know the station address of the node.

For example, to define a link station on port 4 to an unknown node type with a maximum BTU size of 1033 and a noncanonical MAC address of %100040C08ACE, enter:

```
ADD !4 -APPN AdjLinkSta LEARN 1033 %100040C08ACE
```

Defining Entries in the Network Node's Directory

The network node maintains a directory of nodes it knows about, and any logical units on those nodes. When an incoming session request comes to the network node, the network node uses the information stored in the directory to determine the location of the destination LU. If the destination LU is not located in the network node's domain, the network node sends locate requests to adjacent network nodes.

Figure 10-5 shows a network node and what nodes would be included in the network node's directory. Network node A is the local node. The shaded area indicates nodes that would be included in network node A's directory, either dynamically learned or statically defined. End nodes A1 and A2 are in network node A's domain, and would be dynamically learned and added to the directory. LEN end node A3 is also in network node A's domain; if there are other LUs on that LEN node other than the LU for the node's CP, these additional LUs would have to be statically defined (for more information on defining LEN end node LUs, see below). Network nodes B and C are also dynamically learned in network node A's directory because both are adjacent nodes, one hop away.

Network node D would not be included in the directory because it is not an adjacent node, and is two hops away. End nodes C1 and C2 would not be included because they reside in network node C's domain; as a result, end nodes C1 and C2 would be included in network node C's directory.

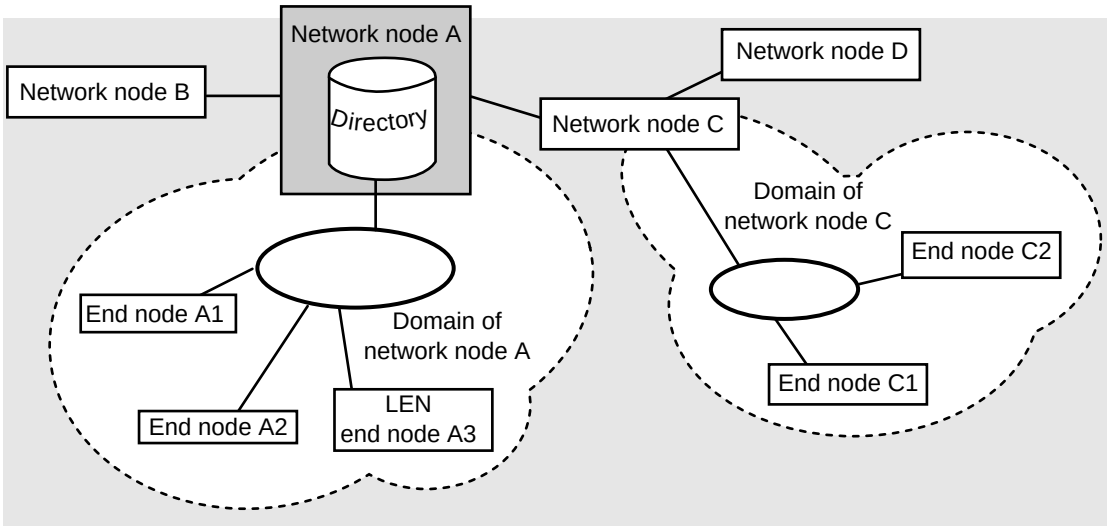


Figure 10-5 Nodes Included in the Network Node Directory (Example)

When you display the directory, the display shows the location of the logical units. In this example for network node A's directory, end nodes LUs on A1 and

A2 would be *registered* entries, meaning they were dynamically learned, and the location would be *domain*, meaning they reside in the local domain. The LU on LEN end node A3 would be a *home* entry (meaning it was statically defined), and the location would be *domain*. For more information on displaying the directory, refer to the DIRectory parameter in Chapter 5 of *Reference for NETBuilder Family Software*.

Preconfiguring LEN End Node LUs

When a LEN end node is added to an APPN network as an adjacent link station, the LEN end node sends an XID3 to the network node when the link activates. In this XID3, the LEN end node's CP name is sent in the ox0E control vector type F4. This CP name maps to the LEN end node's LU name. However, if the LEN end node has more than one LU, then you must statically preconfigure those LUs into the network node directory.

Figure 10-6 is an example of two LEN end nodes connected directly with the intermediate network node. On a LEN end node, the single LU that maps to the node's CP name that was sent in the control vector is dynamically registered through the XID3 with the network node when the link is activated. In the figure, both LU AA on PC AA and LU BB on PC BB would be dynamically registered.

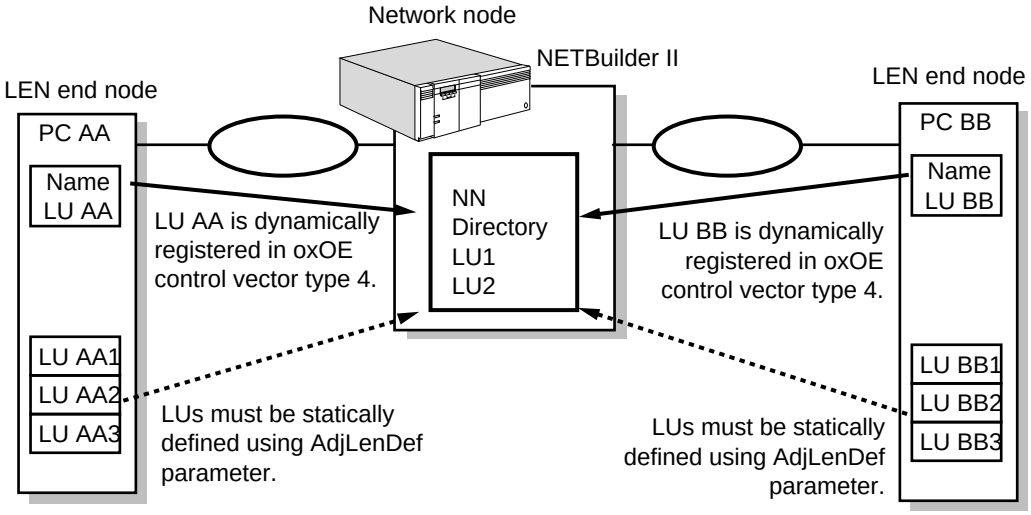


Figure 10-6 LEN End Node LU Registration

You must statically define LEN end node LUs in the following situations:

- If the LU name does not match the CP name
- If the control vector does not send the LU name
- If the LEN end node has LUs in addition to the LU that is registered through the XID3

The PCs in the figure show the last situation, in which both PCs have additional LUs. Since the XID3 only registers the LU for the network name control vector, these additional LUs must be statically defined into the network node's directory using the AdjLenDef parameter.

In APPN, when two LEN end nodes have a peer-to-peer connection, either side can activate the connection or start a session to the other node. The LEN end node that activates the connection sends a BIND to the other node. For the connection to work, the LEN end node that receives the BIND has to be preconfigured into the network node directory so that the network node can find the destination LU to send the session request.

For example, if LU AA in the figure activates a session to LU BB2, then LU BB2 must be preconfigured in the network node's directory; otherwise, the session request will not be successful. If LU AA2 wants to activate a session to LU BB2, both LUs need to be preconfigured in the network node directory. After these two LUs are preconfigured, either LU can initiate a connection. Also, once LUs are preconfigured in the network node directory, other LUs in the network can find them. Conversely, if LUs that require preconfiguration are not in the network node directory, other LUs in the network will not find them.

To statically define LEN end node LUs into the network node directory, follow these steps:

1. If you have LEN end nodes with more than one LU, or LEN end nodes in the network node domain that will receive BINDs that do not match the CP name in the XID, you must statically define these LUs using:

   ```
   ADD -APPN AdjLenDef [adjnetid.]<adjcpname> [adjlu ...]
   ```

   This command statically defines any logical units on the LEN end node in the local network node server's directory.

   For example, to add the three LUs named AA1, AA2, and AA3 on the LEN end node AA, enter:

   ```
   ADD -APPN AdjLenDef AA AA1 AA2 AA3
   ```

   When you add CP and LU names, the names are converted to all uppercase, even if you enter some lowercase letters. When entering this command, you can use the not fully qualified CP name. Use this command to define up to 4 LUs at a time; to define additional LUs, reenter the command. You can register up to 256 LUs on the network node.

2. Repeat the previous step for each LEN end node in your network node's domain with more than one LU. The entries take effect immediately.

   For information on how to display entries in the directory, refer to "APPN Directory Information" on page 10-38.

Deleting LEN End Node LUs

You can delete statically defined LEN end node LU entries from the directory using the DELete -APPN AdjLenDef command. You can specify individual LUs to be deleted. If you do not specify LU names in the command, the entire adjacent node is deleted from the directory, along with all LUs belonging to the adjacent node.

For example, to delete the LUs named AA2 and AA3 on node AA, enter:

```
DELete -APPN AdjLenDef AA AA2 AA3
```

Adding Entries

In most configurations, you do not need to statically define network nodes and regular end nodes in the directory. You do need to determine how many cached entries you will allow and if you have LEN end nodes that receive BINDs, you must statically define them for the directory.

Unlike LUs on LEN end nodes that may require static definition, LUs on end nodes and network nodes are normally learned dynamically. Although not required, you can also statically predefine the location of LUs on other nodes in the network.

To preload entries into the APPN directory cache, use:

```
ADD -APPN DirectoryEntry [netid.]<resource name>
   <type(LU|EN|NN|Wild)> [[netid.]<parent_name>
   <parent_type(EN|NN)>] [[netid.]<grandparent_name>
   <grandparent_type(NN)>]
```

Using this command, you enter the resource type into the directory. If the resource is not a network node, you must specify the parent name and parent type of the resource. The resource parent and child is used for destination node broadcast searches. When a node or LU is a child resource, the child must reply to the parent for a search to be completed.

Figure 10-7 is a simple example of how this directory hierarchy works. In this example on the network HQ, the network node NN22 is the parent resource to the end node ENGREEN, which is the child. ENGREEN is the parent resource to the LU named NSDOS, which is a child resource residing on that end node.

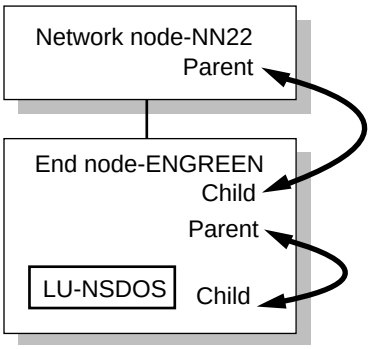


Figure 10-7 Parent and Child Directory Entries

To add a directory entry in which the LU named HQ.NSDOS is the child to the end node HQ.ENGREEN, which is a child entry to the network node HQ.NN22, enter:

```
ADD -APPN DirectoryEntry HQ.NSDOS LU HQ.ENGREEN EN HQ.NN22 NN
```

In this example, the network node HQ.NN2 is the grandparent entry to the LU HQ.NSDOS. When entering an entry for a grandchild (three levels down), you must specify the grandparent name. The grandparent type will always be a network node.

Alternatively, you can enter these directory entries separately. For example, you can enter the following three commands, the first to define the network node,

the second to define a child entry for the end node, and the third to define a child entry for the LU:

```
ADD -APPN DirectoryEntry HQ.NN22 NN
ADD -APPN DirectoryEntry HQ.ENGREEN EN HQ.NN22 NN
ADD -APPN DirectoryEntry HQ.NSDOS LU HQ.ENGREEN EN HQ.NN22 NN
```

You can add wildcard entries to the directory. Wildcards are of two types: full, where you just enter an asterisk (*), or partial, where you enter part of the name and an asterisk (for example, LU7*).

To add a partial wildcard entry for all LUs that start with "LU7" as child entries to HQ.NN22, enter:

```
ADD -APPN DirectoryEntry LU7* Wild HQ.NN22 NN
```

Deleting Entries

To delete entries from the network node directory, use:

```
DELete -APPN DirectoryEntry [netid.]<lu_name> <type(LU|EN|NN|Wild)>
```

For example, to delete the directory entry NSDOS, for the LU on ENGREEN, enter the following command, entering the LU name and specifying the type as LU:

```
DELete -APPN DirectoryEntry HQ.NSDOS LU
```

If you delete a resource, all the child entries and grandchild entries belonging to that resource will also be deleted. For example, if you delete the grandparent entry HQ.NN22, the child entry HQ.ENGREEN and the grandchild entry HQ.NSDOS will also be deleted.

Configuring Parallel Transmission Groups

A transmission group (TG) is the link between two nodes. By configuring parallel TGs, you can configure two links from the local network node to the same adjacent node. This can provide more flexibility in routing APPN traffic to and from a single device. With parallel TGs, you can configure two links between the same two nodes, but not more than two.

Parallel TGs are not recommended for links over the same LAN, because there is no practical benefit for doing so; if you have parallel TGs over the same LAN and the LAN is busy, then both TGs will be busy.

Not all network devices support parallel TGs. Make sure the device supports parallel TGs before configuring more than one link to it. If the other node is a NETBuilder II bridge/router running version 8.2 or higher, then you can configure parallel TGs to it. If you configure parallel TGs between two NETBuilder II network nodes, then you only need to configure the partner node as an adjacent link station on one side.

There are several reasons why parallel TGs can be useful on your network:

- They can provide redundant links between nodes, to enable one link to take over if the other fails.
- You can assign different security levels to different TGs between nodes, allowing greater control over the traffic.

- You can assign different classes of service to each of the two TGs, allowing you to isolate different types of traffic over each link.
- You can have greater bandwidth between two nodes

When running parallel TGs, the CP-CP sessions can only go over one link at a time. With CP-CP session error recovery, if the link goes down the CP-CP sessions can be brought back up on the other link. For more information, refer to "CP-CP Sessions on Parallel TGs" on page 10-27.

Figure 10-8 is an example of parallel TGs being used for redundant links. In the configuration, both links between the network nodes are running at the same speed, and are running the same type of traffic. Each link is over a different port.

Although the links are redundant, if one link fails the traffic is not automatically switched to the second link. Unlike connectionless protocols, which can automatically switch links if a link fails, APPN is connection-oriented. As a result, if a link fails you will lose data, but you can restart your sessions over the second link.

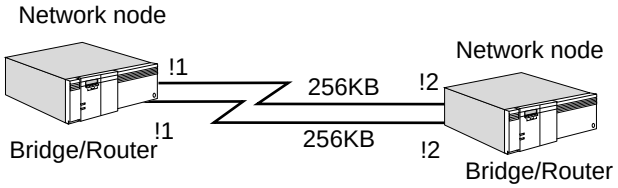


Figure 10-8 Parallel TGs for Redundant Links

Figure 10-9 is an example of parallel TGs being sent over two different LANs. This configuration allows you to have redundancy between two nodes in your LAN environment. If one LAN fails, then you can restart sessions over the second LAN. If you configured both links on the same LAN, and the LAN fails, then both nodes would be isolated.

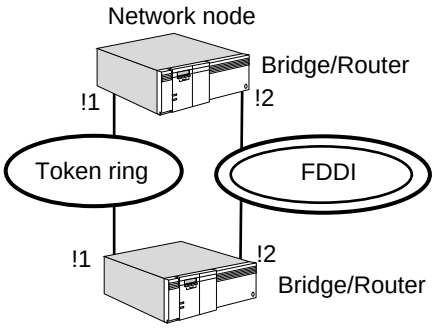


Figure 10-9 Parallel TGs over Different LANs for Resiliency

Figure 10-10 is a configuration in which parallel TGs are being used to isolate different types of traffic through different classes of service. The link on the left is set for a capacity of 256 KB and is being used for interactive traffic between terminals and the host; this type of traffic demands quicker response time so the class of service (COS) being used allows for a higher priority and the link is set for a higher speed. The link on the right is being used for lower speed batch transmissions, and as a result, is using the BATCH class of service and the link is set to a lower speed. In this example, the interactive traffic will be prioritized higher than the batch traffic.

For more information on configuring APPN class of service, refer to Chapter 12.

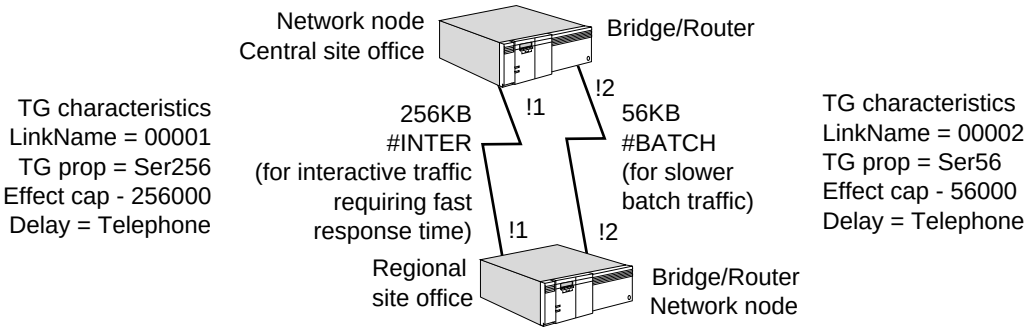


Figure 10-10 Parallel TGs for Isolating Class of Service Traffic

Configuring Parallel TGs on the Network Node

Figure 10-11 is an example of a NETBuilder II bridge/router network node with parallel TGs over two different ports to an AS/400. In this example, the TGs are on two different FDDI rings, one being used for primary traffic and the other used as a backup.

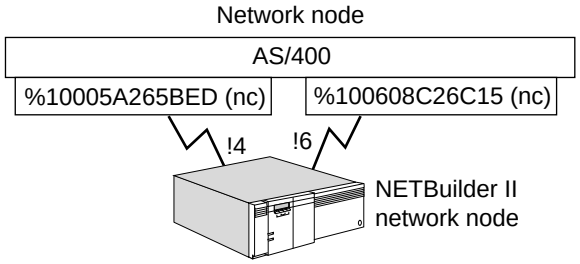


Figure 10-11 Configuring Parallel TGs

To configure the parallel TGs for ports 4 and 6 on the NETBuilder II bridge/router in the figure, follow these steps:

1. Define the ports using:

```
SETDefault !<port> -APPN PortDef = <DLC type>
(LLC2|FR|PPP|DLSW|SDLC|UNdef) <max_btu_size>(99-8192)
[ActLimit=<limit>(1-512)] [TGprof=<name>] [HPR=(Yes|No)]
[ErrorRecovery=(Yes|No)] [DatMode=(Half|Full)] [ROle=(Pri|Sec|Neg)]
```

   Define port 4 for LLC2 traffic, a maximum BTU size of 1033, and assign the TG profile "FDDI" by entering:

```
SETDefault !4 -APPN PortDef = LLC2 1033 TGprof=FDDI
```

   Define port 6 for LLC2 traffic, a maximum BTU size of 1033, and assign the TG profile "FDDI" by entering:

```
SETDefault !6 -APPN PortDef = LLC2 1033 TGprof=FDDI
```

2. Define the adjacent link stations for both ports using:

```
ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn)
   <max_btu_size>(99-8912) [[Cmac|Ncmac] dest media addr]
   [Sap=<num>] [CPName=[netid.]cpname] [Nodeid=<ID>]
   [LinkName=<name>] [TGprof=<name>] [AutoStart=(Yes|No)]
   [CPSess=(Yes|No)] [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

Add the adjacent link station to port 4 to the destination media address on the AS/400 (entering the address in noncanonical format), a SAP of 08, and to specify autostart and CP-CP session activation by entering:

```
ADD !4 -APPN AdjLinkSta NN 1033 %10005A265BED Sap=08 TGprof=FDDI
AutoStart=Yes CPSess=Yes
```

CP-CP sessions can only be active over one TG at a time.

Add the adjacent link station to port 6 for the different destination media address, a SAP of 08, specifying autostart and support for CP-CP sessions by entering:

```
ADD !6 -APPN AdjLinkSta NN 1033 %100608C26C15 Sap=08 TGprof=FDDI
AutoStart=Yes CPSess=Yes
```

You can configure any adjacent link station characteristics using the LinkStaCHar parameter.

You cannot assign specific numbers to specific TGs. The TG numbers are assigned through negotiation between the two nodes.

You can also configure parallel TGs for links to an SDLC device. You perform the same procedure, but you use the SdlcAdjLinkSta parameter.

CP-CP Sessions on Parallel TGs

When parallel TGs are configured between 3Com network nodes and both TGs support CP-CP sessions, a CP-CP session on one TG will not switch to the other TG if the user disables the port or path. This situation occurs because both sides learn about the link failure at different times. The network node with the disabled port or path learns about the link failure immediately and tries to bring CP-CP sessions up on the second TG. However, the second network node does not learn about the link failure until LLC2 times out. Because the node thinks the link is still up, the second network node does not allow CP-CP sessions to start on the second TG. After five attempts at bringing up CP-CP sessions on the second TG, the second TG will be flagged as not supporting CP-CP sessions, which prevents CP-CP sessions from coming up on that second TG.

To prevent this situation, manually stop the first TG by entering the SET -APPN LinkStaCONTrol <LinkName> Deactivate command before disabling the port and path. By doing this, both network nodes then learn that the link has gone down at the same time, and CP-CP session can be activated on the second TG.

Parallel TGs and Source Route Dual-TIC Topologies

You can configure parallel TGs in environments in which dual or multiple token ring interface cards (TICs) are configured on front-end-processors. For more information on dual-TIC topologies, refer to "Configuring DLSw for Dual-TIC Topologies" on page 24-25.

Configuring DLSw Between Network Nodes

You can configure your APPN network so that you can send SNA traffic encapsulated in TCP packets over an IP network between two APPN network nodes using DLSw.

To configure DLSw between APPN nodes, additional configuration is necessary. Figure 10-12 is an example of two bridge/routers acting as APPN network nodes using DLSw to encapsulate SNA traffic in TCP packets across an IP internetwork.

Table 10-2 lists the commands that need to be configured on each bridge/router in the figure.

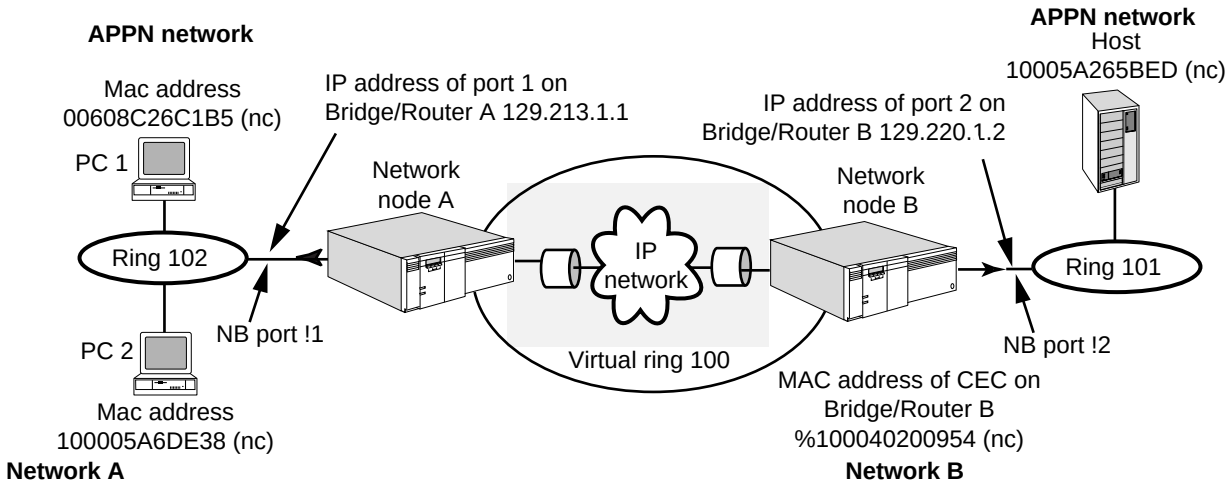


Figure 10-12 Configuring DLSw Between Two APPN Network Nodes

Table 10-2 Commands to Configure DLSw Between Two APPN Network Nodes

Commands entered on Bridge/Router A	Commands entered on Bridge/Router B
SETDefault -IP CONTrol = Enable	SETDefault -IP CONTrol = Enable
SETDefault -TCP CONTrol = KeepAlive	SETDefault -TCP CONTrol = KeepAlive
SETDefault -TCP KeepAliveLimit = 3	SETDefault -TCP KeepAliveLimit = 3
SETDefault !1 -LLC2 CONTrol = Enable	SETDefault !2 -LLC2 CONTrol = Enable
SETDefault -LLC2 TUNnelVRing = 100	SETDefault -LLC2 TUNnelVRing = 100
SETDefault -DLSW Interface = 129.213.1.1	SETDefault -DLSW Interface = 129.220.1.2
ADD !1 -DLSW PEer 129.220.1.2	ADD !1 -DLSW PEer 129.220.1.1
SETDefault -DLSW CONTrol = EnableSNA, DisableNetBios	SETDefault -DLSW CONTrol = EnableSNA, DisableNetBios
SETDefault !0 -APPN PortDef = DLSW 1033	SETDefault !0 -APPN PortDef = DLSW 1033
ADD !0 -APPN AdjLinkStation NN 1033 N%100040200954	

As shown in the figure, you configure the network nodes as DLSw peers and the DLSw tunnel interface information using the normal procedure. For specific instructions on how to configure DLSw peers, refer to Chapter 24. For information on the parameters in the DLSw Service, refer to Chapter 19 in *Reference for NETBuilder Family Software*.

You cannot perform bridging and tunneling of the same MAC address from an end station. You can perform either bridging only or tunneling only, but not both at the same time.

After configuring the two bridge/routers as DLSw peers, to configure DLSw tunneling between two APPN network nodes, follow these steps:

1 On both APPN network nodes acting as DLSw tunnel peers, configure the APPN port definition using the SETDefault !<port> -APPN PortDef syntax, specifying DLSw as the DLC type.

   When specifying the port definitions for DLSw, you must specify the port number as !0. You only need to set the port definition for !0 for ports used for DLSw, and you should not specify !0 when setting the port definition for any other DLC type.

   On bridge/router A in the figure, using port 0 and setting a maximum BTU size of 1033, enter:

   ```
   SETDefault !0 -APPN PortDef = DLSw 1033
   ```

   On bridge/router B in the figure, using port 0 and setting a maximum BTU size of 1033, enter:

   ```
   SETDefault !0 -APPN PortDef = DLSw 1033
   ```

   The maximum BTU size does not have to match on both sides of the tunnel. If the maximum BTU sizes differ, the smaller value will be used.

2 On the bridge/router that will initiate the connection, configure the tunnel peer bridge/router as an adjacent link station using:

   ```
   ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn)
      <max_btu_size>(99-8912) [[Cmac|Ncmac] dest media addr]
      [Sap=<num>] [CPName=[netid.]cpname] [Nodeid=<ID>]
      [LinkName=<name>] [TGprof=<name>] [AutoStart=(Yes|No)]
      [CPSess=(Yes|No)]
   ```

   When you enter the command, you specify that the peer is a network node, the maximum BTU size, and the MAC address of the tunnel peer. In this case, the tunnel peer will always be a network node, since the bridge/router can only serve as a network node. The MAC address you enter is the address of the tunnel peer bridge/router, not the destination SNA host (also shown in the figure).

   In the example shown in the figure, enter the following command on bridge/router A to add the link station as a network node with a maximum BTU size of 1033 and a SAP value of 08:

   ```
   ADD !0 -APPN AdjLinkSta NN 1033 N%100040200954 Sap=08
   ```

   When adding the adjacent link station for DLSw, you must specify the port number as !0 to map to port 0 configured in the previous step.

   For more information about configuring data link switching, refer to Chapter 24. For information on parameters in the DLSw Service, refer to Chapter 19 in *Reference for NETBuilder Family Software*.

Configuring APPN for Boundary Routing

Boundary Routing is the 3Com system architecture that allows a network administrator to connect a central office network to a large number of small remote office networks (leaf networks). You can configure APPN to work in Boundary Routing environments, but there are limitations as to the types of configurations that can be set up. No additional APPN configuration is required for Boundary Routing environments. For information on Boundary Routing concepts and how to configure the central office router, refer to Chapter 32.

The 3Com Boundary Routing architecture is different from the APPN concepts of boundary nodes and border nodes. The 3Com APPN implementation supports

the concept of boundary nodes but does not support the Systems Network Architecture (SNA) concept of border nodes or perform the border function. For clarification of these terms, refer to the IBM document, *APPN Architecture and Product Implementations Tutorial* listed in "IBM APPN References" on page 10-54.

Figure 10-13 is an example of a NETBuilder II bridge/router acting as an APPN network node and performing as the central site router in a Boundary Routing configuration connected to a SuperStack II NETBuilder bridge/router acting as a leaf node, which in turn is connected to a token ring network with APPN end nodes. The CP-CP session takes place between the NETBuilder II network node and the end node. The SuperStack II bridge/router acting as the leaf node does not participate in the CP-CP session, and cannot serve as an APPN node because the APPN software is not supported on the SuperStack II bridge/router platform. In this situation, no special configuration is required on the SuperStack II bridge/router.

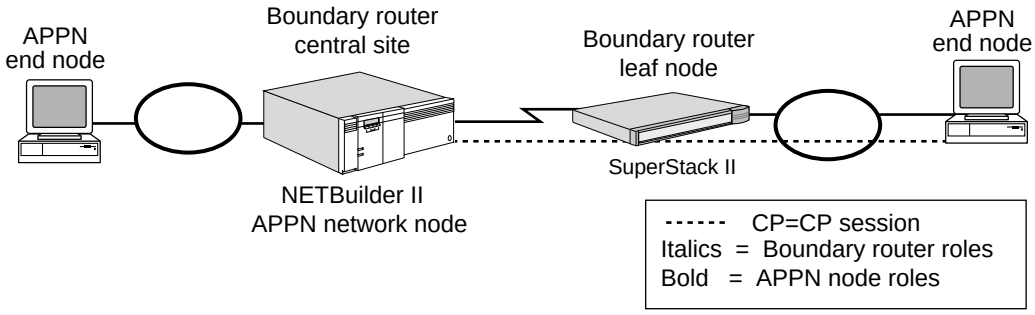


Figure 10-13 Configuring APPN with the Boundary Routing Architecture

You can also use Boundary Routing with APPN connection networks. For more information, refer to "Using Connection Networks in Boundary Routing Environments" on page 10-33.

If you are configuring APPN for Boundary Routing, the following special configuration is required on the central site bridge/router:

- You must configure the port definition to DLSw by entering:

  SETDefault !0 -APPN PortDef = DLSW

- To enable the central site bridge/router to send ring information to the leaf node, you must configure the central site WAN link as a source route link, and turn on route discovery for both IP and LLC2 using:

```
SETDefault !<port> -SR RouteDiscovery = IP, LLC2
```

Configuring APPN Connection Networks

Connection networks are a way to provide greater scalability for growing APPN networks without exponentially increasing the number of broadcast traffic and overhead that could affect network performance. By configuring connection networks, you can enable links from one node to another through a virtual routing node; although the virtual routing node is an intermediate node, the link from the source node to the destination node is a virtual link.

Defining connection networks through virtual routing nodes is a method for setting up the network topology so that you can increase the number of nodes without flooding the network with topology data unit (TDU) broadcasts.

The following sections describe these connection network topics:

- How connection networks can be used to scale large APPN networks
- How to configure links to connection networks
- How to use connection networks in boundary routing environments

Using Connection Networks to Scale Larger Networks

Every time you add a new network node to the APPN network, you increase the amount of traffic overhead since each node broadcasts TDU updates to other directly connected nodes when the network changes. As you add nodes and scale the network, the network will be subject to increasing numbers of broadcasts, including Locate broadcasts, reducing network performance.

Figure 10-14 is an example of a fully meshed APPN network in which all eight nodes are directly connected to each other. Although each node is one hop from each other, the large number of TGs means an exponential number of TDU broadcast updates flooding the network.

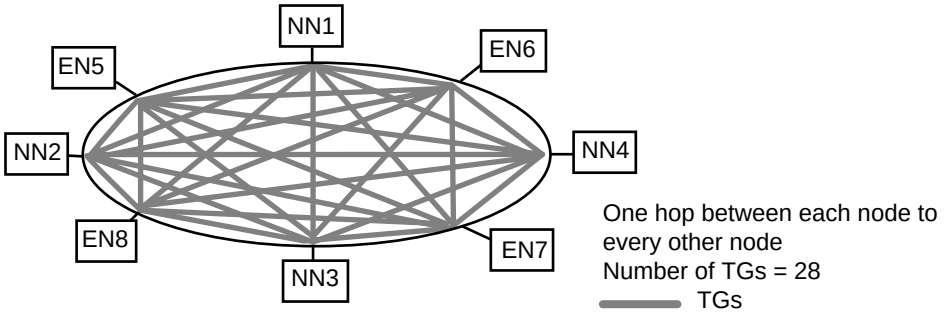


Figure 10-14 Meshed APPN Network without Virtual Routing Nodes (Direct Links)

Figure 10-15 shows the same network, but with a virtual routing node being used to provide any-to-any connectivity between each node. In this configuration, NN1 is the focal point through which all links go through. Each node only requires link definitions to the common network node (NN1), and the virtual routing node. Because of the virtual routing node, the session data is not routed through real network nodes, reducing the number of CP-CP sessions as well as the number of TDU updates and Locate broadcasts.

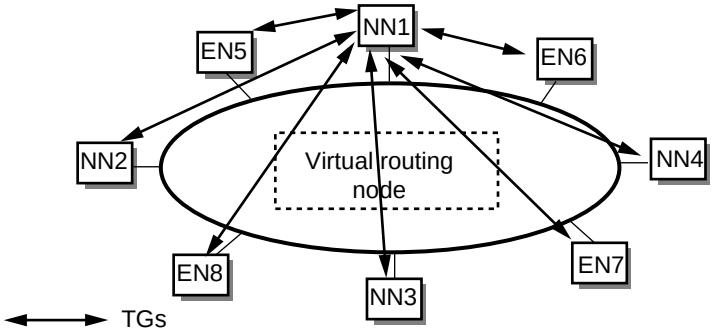


Figure 10-15 Network with Virtual Routing Node (One Point of Failure)

One problem with this configuration, however, is there is only one point of failure; if NN1 goes down, it segments your network topology so that TDU updates will not flow. You can configure more than one common network node to provide redundancy in your network. Figure 10-16 is the same network, only now NN1 and NN3 are common network nodes, each with its own network segment. In this configuration, if NN1 went down, all CP-CP sessions would go down, and network connectivity would be unknown; nodes in NN3s network segment would stay up, although they would not be able to connect with any nodes on NN1s segment. Also, by linking NN1 and NN3 through the virtual routing node, the TDU updates and Locate broadcasts would be isolated to each network segment.

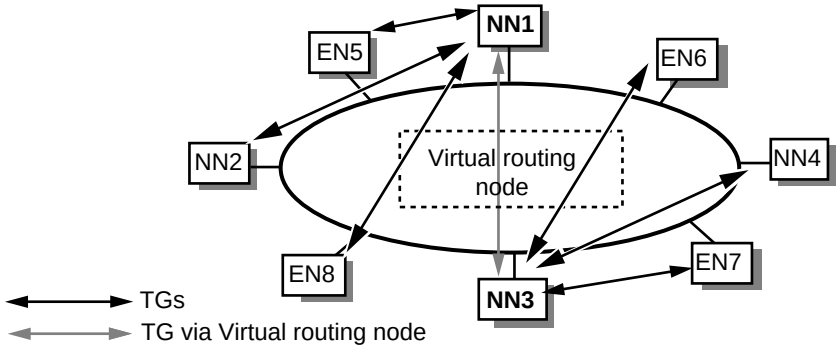


Figure 10-16 Segmented Network with Virtual Routing Node (Redundant Points of Failure)

Configuring Links to Connection Networks

To configure a link to a connection network, follow these steps:

1. Define the connection network to the port using:

```
ADD !<port> -APPN ConnNetworkDef [netid.]<cn name>
   [TG profile name]
```

Using this command, you map the connection network to the port and, if desired, assign a TG profile to the connection network. For example, to add a connection network named US3COMHQ.CN4 to port 4 and assign the TG profile FDDI to it, enter:

```
ADD !4 -APPN ConnNetworkDef US3COMHQ.CN4 FDDI
```

2. If desired, change the characteristics of the connection network using:

```
SETDefault -APPN ConnNetworkChar = <cn name> [EffectCap=<string>]
   [ConnectCost=<0-255>] [ByteCost=<0-255>] [Security=<string>]
   [PropDelay=<string>] [Usd1=<0-255>] [Usd2=<0-255>]
   [Usd3=<0-255>]
```

Using this command, you can change any or all characteristics of the connection network. For example, to change the CN4 connection network's security level to SecureCnd and byte cost to 255, enter:

```
SETDefault -APPN ConnNetworkChar = CN4 ByteCost=255
   Security=SecureCnd
```

For more information on these parameters, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

You can delete a defined connection network using:

```
DELete !<port> -APPN ConnNetworkDef [netid.]<cn name>
```

Using Connection Networks in Boundary Routing Environments

One problem with large remote APPN networks is that if you have a lot of nodes you need to configure each remote node as an adjacent link station. Also, if you are running a Boundary Routing configuration in which a NETBuilder II bridge/router is the central site router and you have many APPN nodes at the remote site, you will have increased traffic over the WAN link every time the remote nodes initiate sessions with each other.

Figure 10-17 is the problem this situation can create. In this configuration, APPN nodes are on a LAN at the remote site while the network node server is the bridge/router at the central site. Because the network node server is not on the LAN, if end node A wants to initiate a session with end node B, it must first initiate an LLC2 session with the network node at the central site to discover the location of end node B. The LLC2 sessions travel over the WAN link to and from the end nodes on the LAN. If you have many nodes at the remote site LAN sending LLC2 sessions over the WAN link to the central site, this will increase traffic over the WAN link and reduce performance.

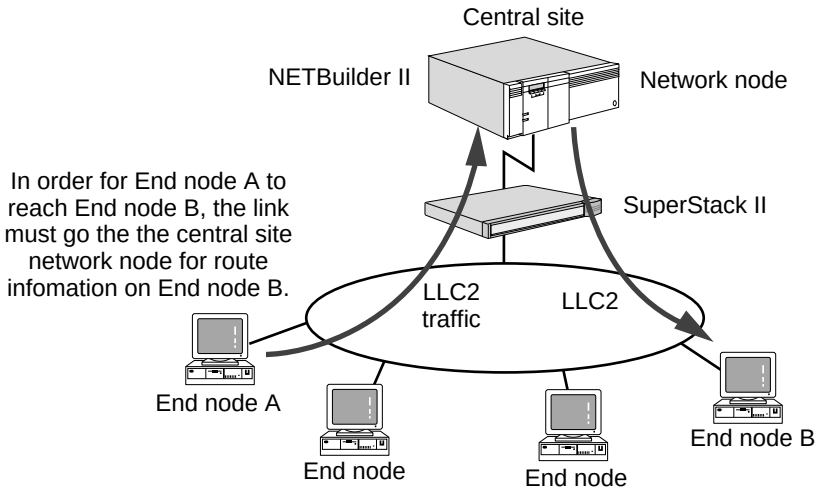


Figure 10-17 APPN and Boundary Routing without Remote Site Connection Network

Figure 10-18 shows the same configuration in which a connection network has been defined for the remote site LAN. By defining all the nodes on the LAN to the connection network, the resources on that LAN are defined on the network node only once. After the resources on the LAN are defined, end node A can discover the location of end node B and initiate sessions with it directly, bypassing the central site router. This will reduce traffic over the WAN link.

You can configure more than one remote connection network. Figure 10-18 is an example where two different remote LANs are configured as two different connection networks from the same central site router.

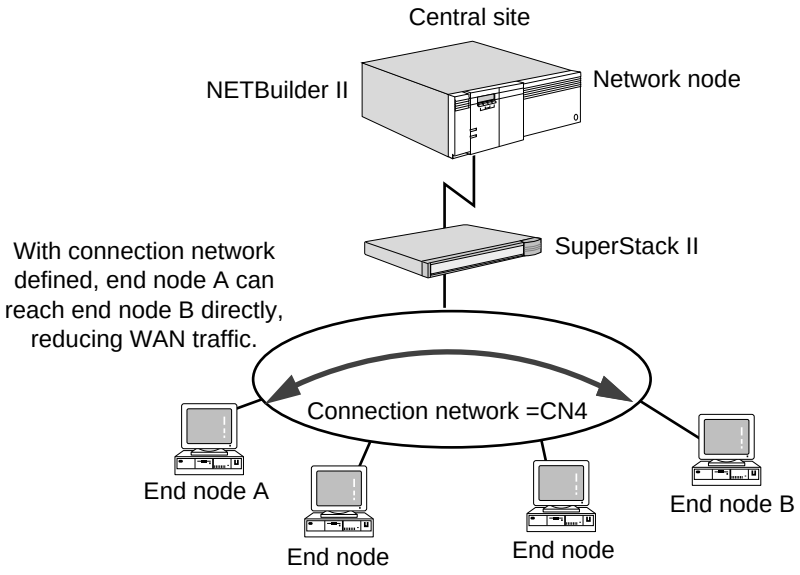


Figure 10-18 APPN and Boundary Routing with Multiple Remote Connection Networks

Operating the Network Node

After you have configured the APPN network node and it is handling sessions properly, you can perform a number of operations to control the node and links to the node. This section describes how to do the following tasks:

- Disable the network node
- Delete adjacent link stations
- Activate and deactivate APPN ports and link stations
- Display APPN information

Disabling the Network Node

You can disable the APPN network node in one of two ways:

- Totally disable the network node and take it off the network
- Dynamically disable the network node so that when the bridge/router is rebooted, the network node is automatically re-enabled

To disable the APPN network node and take it off the network, enter one of the following commands:

```
SET -APPN CONTrol = Disable
```

or

```
SETDefault -APPN CONTrol = Disable
```

If you use the SET command, and you reboot the bridge/router, the network node will automatically be enabled. If you use the SETDefault command, you will have to re-enable the network node using the SETDefault -APPN CONTrol = Enable command if you reboot the bridge/router.

When you disable the network node, you must choose either an orderly or immediate deactivation. If you specify Immediate, the links will be deactivated first, then the ports on the network node, and then the network node itself. If you specify Orderly, the node will first be advertised as "Quiesced," the session

limits will then be reset on all modes. After all ISR sessions have ended, all endpoint sessions and then all CP-CP sessions are unbound. The links are deactivated, followed by the ports on the network node, and then the network node itself. If you do not specify either, an immediate deactivation will take place.

To dynamically disable the network node with an orderly deactivation, enter:

```
SET -APPN CONTrol = Disable Orderly
```

To dynamically disable the network node with an immediate deactivation, enter:

```
SET -APPN CONTrol = Disable Immediate
```

When you disable the APPN network node, you will receive a message similar to the following:

```
Wed Dec 31 16:11:15 1993 LOCAL NETWORK NODE US3COMHQ.GOLD IS
   STOPPED
```

After the command is entered, the network node will not participate in the network and exchange traffic with other APPN nodes. When you disable the network node, any active sessions may be disrupted.

CAUTION: *3Com recommends that there be no active ISR sessions on the network node when you disable it. If you specify an orderly deactivation, the system will wait for all ISR sessions to go down before disabling the node.*

To re-enable a previously disabled network node, enter:

```
SET -APPN CONTrol = Enable
```

Deleting Links to Adjacent Nodes

As your network needs change, you can change the network topology by deleting adjacent link stations from the network node.

To delete an adjacent link station, use the DELete !<port> -APPN AdjLinkSta syntax and specify the link name of the station being removed. For example, to delete the adjacent link station on port 3 with a link name of "LINK0005," enter:

```
DELete !3 -APPN AdjLinkSta LINK0005
```

To obtain a list of link names, enter:

```
SHow -APPN LinkStaCONTrol
```

Activating and Deactivating APPN Ports and Links

You can dynamically activate and deactivate APPN ports and link stations as needed. For example, if you need to deactivate a specific port for troubleshooting purposes, you can deactivate the port. You can also deactivate a specific link station on a port, also for troubleshooting purposes. By deactivating a link station you can then reactivate the link without having to redefine the link station.

Activating and Deactivating Ports

After an APPN port has been activated, if you want to change any of the configuration attributes for that port, you must first deactivate the port. After you have made your configuration changes, you then reactivate the port.

To dynamically activate or deactivate an APPN port, use:

```
SET !<port> -APPN PortCONTrol = (<Activate [NoLinkStations] |
   Deactivate [Orderly | Immediate]>)
```

This procedure applies only to ports defined for APPN using the SETDefault !<port> -APPN PortDef command. If the port is being used to send or receive other protocol traffic, only APPN data will be affected.

When you deactivate a port, you specify either an orderly or immediate deactivation. If you specify orderly, the system waits for all ISR sessions to terminate before deactivating the port; if you specify Immediate, the system will not wait for ISR sessions to terminate. If you specify Immediate, all sessions will first be terminated, then all LLC2 sessions will be terminated; after these processes take place, the port is deactivated. If you do not specify either, then an immediate deactivation will take place.

For example, to deactivate port 3 with an orderly deactivation, enter:

```
SET !3 -APPN PortCONTrol = Deactivate Orderly
```

After you enter the command, port 3 will be deactivated from the APPN network. If you have active link stations on that port, all links will be deactivated. When you deactivate a port, all sessions or BINDs to that port will automatically be terminated.

To activate a port and activate all the link stations on that port, enter the SET !<port> -APPN PortCONTrol command and specify "Activate." For example, to activate port 3 and activate all its defined link stations, enter:

```
SET !3 -APPN PortCONTrol = Activate
```

To activate a port but not activate any defined link stations, specify "NoLinkStations" in the command. For example, to activate port 3 but not activate any of its defined link stations, enter:

```
SET !3 -APPN PortCONTrol = Activate Nolinkstations
```

For more information on the PortCONTrol parameter, refer to Chapter 5 of *Reference for NETBuilder Family Software*.

Activating and Deactivating Links

After a link has been activated, if you want to change any of the configuration attributes for that link, you must first deactivate the link. After you have made your configuration changes, you then reactivate the link.

To dynamically activate or deactivate a link, use:

```
SET -APPN LinkStaCONTrol = <LinkName> <Activate | Deactivate
[Orderly | Immediate]>
```

You must specify the local link station name in the command. To find out what the link name is, enter:

```
SHow -APPN LinkStaCONTrol
```

When you deactivate a link, you specify either an orderly or immediate deactivation. If you specify Orderly, the link is deactivated when all sessions are

stopped. If you specify Immediate, all sessions are first stopped and then the link is deactivated. If you do not specify either, an immediate deactivation will take place.

For example, to perform an orderly deactivation for a link named "Link01," enter:

```
SET -APPN LinkStaCONTrol Link01 Deactivate Orderly
```

The link is deactivated until you enter:

```
SET -APPN LinkStaCONTrol Link01 Activate
```

For more information on the LinkStaCONTrol parameter, refer to Chapter 5 of *Reference for NETBuilder Family Software*.

When you activate adjacent link stations, you may receive a message on the console indicating that the CP-CP session has been activated or deactivated. For example, if you activated an adjacent link station to the node US3COMHQ.GOLD, you will receive messages similar to the following if the command was successful:

```
CONLOSER CP-CP SESSION WITH US3COMHQ.GOLD IS UP
CONWINNER CP-CP SESSION WITH US3COMHQ.GOLD IS UP
```

The first message indicates the contention loser (conloser) of the CP-CP session is up while the second message indicates the contention winner (conwinner) of the CP-CP session is up. When you deactivate adjacent link stations, you receive similar messages but they specify "DEACTIVATE."

The messages showing information on contention winners and losers only appear if the link supports CP-CP sessions, and only if CP-CP sessions exist on the link. For example, if the link is between a network node and a LEN end node, you cannot have CP-CP sessions because they are not supported on LEN end nodes.

Pinging to APPN Network Resources

Sometimes you cannot reach a given APPN network resource. Use the APpnPING command to determine if the resource is reachable without connecting to it . With APpnPING, you perform an APPC Ping to the LU in the network that you are trying to reach. To perform a ping to an LU on the network, use:

```
APpnPING [netid.]<partner_lu_name> [Mode=modename] [Size=N]
   [Consec=N] [Iterations=N] [Echo=Yes|No] [Userid=<string>
   [Password=<string>]]
```

For example, to ping a resource named US3COMHQ.AS400LU in batch mode with 20 iterations, enter:

```
APpnPING US3COMHQ.AS400LU Mode=#BATCH Iterations=20
```

If the APPC Ping is successful, you will receive a confirmation. If the command is not successful, you will receive a message similar to the following:

```
APPING TO US3COMHQ.GOLD DOES NOT SUCCEED
```

If you specify a userid or a password, note that these options are case-sensitive.

For more information about the APpnPING command, refer to Chapter 1 in *Reference for NETBuilder Family Software*.

Displaying APPN Information

You can obtain different types of information regarding the APPN network, including end node and network node topology information. You can also display a list of LUs and their locations that the local network node knows about.

APPN Directory Information

The APPN directory database stores information regarding network resources and their location in the APPN network. To display a list of LU resources and their location known to the local network node, enter:

```
SHow -APPN DIRectory
```

A display similar to the following appears:

```
================================== SHow -APPN DIRectory ==================================
---------------------------------------Directory----------------------------------------
Resource name         Type         Parent name       Type       Entry location       Type
US3COMHQ.CUBE         NNCP                           LOCAL      HOME                 HOME
US3COMHQ.CUBE         LU           US3COMHQ.CUBE     NNCP       LOCAL                HOME
US3COMHQ.LEN1         ENCP         US3COMHQ.CUBE     NNCP       DOMAIN               HOME
US3COMHQ.LU10         LU           US3COMHQ.LEN1     ENCP       DOMAIN               HOME
US3COMHQ.NN1          NNCP                                      X_DOMAIN             HOME
US3COMHQ.EN1          ENCP         US3COMHQ.NN1      NNCP       X_DOMAIN             HOME
US3COMHQ.LU7*         WILDCARD     US3COMHQ.NN1                 X_DOMAIN             HOME
```

This display shows the following types of resources:

- All the local resources of the network node, which includes its own CP and LU, and all LEN nodes defined using the AdjLenDef parameter
- All adjacent end nodes and their registered resources
- All LUs in the network that the network node has discovered
- All resources defined using the DirectoryEntry command

For information on the meanings of the headings in this display, refer to the description of the DIRectory parameter in Chapter 5 of *Reference for NETBuilder Family Software*.

Network Topology Information

To display a list of network nodes known by your network node, enter:

```
SHow -APPN NNtopology
```

A display similar to the following appears:

```
================================== SHow -APPN NNtopology ==================================
---------------------------------------Network Node---------------------------------------
Node name             Type      RAR      Status             Function support       RSN
US3COMHQ.CN5          VRN       128      UNCONGESTED        ISR                    0
US3COMHQ.CN7          VRN       128      UNCONGESTED        ISR                    0
US3COMHQ.CUBE         NN        128      UNCONGESTED        ISR                    2
US3COMHQ.IBM4         NN        128      UNCONGESTED        ISR                    2
US3COMHQ.COM20E       NN        128      UNCONGESTED        ISR                    2
```

This table may not reflect the current network node topology, which means the bridge/router network node may not be able to access all the nodes in the

table. The table shows every network node the bridge/router network node has accessed historically, including nodes that may have since been removed from the network.

For information on the meanings of the headings in this display, refer to the description of the NNtopology parameter in Chapter 5 of *Reference for NETBuilder Family Software*.

To display information regarding local TGs, enter:

```
SHow -APPN TG
```

A display similar to the following appears (showing one TG):

```
======================== SHow -APPN TG ==========================
-----------------Network Node Transmission Group-------------------
Owning node name (type) = US3COMHQ.CN7          (VRN)
TG partner CP name (type) = US3COMHQ.CUBE       (NN)
TG number = 1
FRSN = 55
Days left before deletion = 15
RSN = 2
TG Status = OPERATIVE
Effective Capacity = 56000
Cost per connect time = 68
Cost per byte = 68
Security = 68
Propagation Delay = 68
User defined parameter 1 = 68
User defined parameter 2 = 68
User defined parameter 3 = 68
```

For information on the meanings of the headings in this display, refer to the description of the TG parameter in Chapter 5 of the *Reference for NETBuilder Family Software*.

Adjacent Link Station Information

To display a list of adjacent link stations, enter either:

```
SHow -APPN AdjLinkSta
SHow -APPN SdlcAdjLinkSta
```

This display shows basic information about adjacent link stations. In the display there are columns that may show the characters C, A, H, or E. These indicate support for the CPSess, AutoStart, HPR, and ErrorRecovery values, respectively. The hyphen (–) character means the value is not supported. For more information about the AdjLinkSta and SdlcAdjLinkSta parameters, refer to Chapter 5 of *Reference for NETBuilder Family Software*.

To obtain information regarding the characteristics assigned to each adjacent link station, enter:

```
SHow -APPN LinkStaCHar
```

Current Status of APPN Ports

To display the current status of APPN ports, use:

```
SHow [!<port>] -APPN PortCONTrol
```

If you do not specify a port number, a display similar to the following appears:

```
===================== SHow -APPN PortCONTrol =====================
-----------------Current Defined Ports and Status-----------------
Port            Port Status
!1              ACTIVE
!2              ACTIVE
!3              INACTIVE
!5              INACTIVE
```

If a port is not shown in the display, then that indicates that the port was not defined as an APPN port using the SETDefault !<port> -APPN PortDef syntax.

Active APPN Connections

To display a list of active connections, enter:

```
SHow -APPN CONNection
```

You can specify whether to display only connections to a specific node by entering that node's CP name. You can display all connections in the network topology by entering the SHow -APPN CONNection ALL command. If you do not specify either a CP name or ALL, the display will only show connections to the local network node.

The following is a sample of the display obtained using the SHow -APPN CONNection ALL command:

```
================================== SHow -APPN CONNection ==============================
-------------------------------------Connection Topology--------------------------------
Node name                  Partner name                  TG num     State       RSN
US3COMHQ.CN7(VRN)          US3COMHQ.CUBE                 1          UP          2
US3COMHQ.CN5(VRN)          US3COMHQ.IBM4                 1          UP          26
US3COMHQ.CUBE              US3COMHQ.IBM4                 1          UP          20
US3COMHQ.CN5(VRN)          US3COMHQ.COM20E               1          UP          2
US3COMHQ.CUBE              US3COMHQ.COM20E               1          UP          2
US3COMHQ.IBM4              US3COMHQ.CUBE                 1          UP          8
*US3COMHQ.COM20E           US3COMHQ.IBM4                 1          DOWN        12
```

For more information on the CONNection parameter, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

Current Status of Link Stations

To obtain information regarding the current status of adjacent link stations and DLUr link stations, enter:

```
SHow -APPN LinkStaCONTrol
```

A display similar to the following appears:

```
================================ SHow -APPN LinkStaCONTrol ============================
-------------------------Current Defined Link Stations and Status---------------------
Port           LinkName         AdjCPName              Type          #Sess       LinkStatus
!1             @I000001         US3COMHQ.COM20E        NN            4           ACTIVE
!1             LINK0000         US3COMHQ.IBM4          NN            4           ACTIVE
!1             LINK0064                                               0           INACTIVE
```

If the entry in the LocalLinkName column shows the @ character, this indicates an incoming link station that is not locally defined but was learned dynamically.

Current Status of Adjacent Nodes

To display the current status of adjacent nodes, enter:

```
SHow -APPN AdjNodeStatus
```

A display similar to the following appears:

```
================================ SHow -APPN AdjNodeStatus ============================
-----------------------------------Adjacent Node Status-------------------------------
CP name               Type     TG num     Status           VRN Address    Sap      RSN
US3COMHQ.IBM4         NN       1          OPERATIVE                                2
US3COMHQ.COM20E       NN       1          OPERATIVE                                2
```

This display shows the status of the CP-CP session between the local network node and the adjacent node. The CP name in the display is the name of the adjacent node. For more information about the data in the display, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

Intermediate Session Routing Information

Intermediate Session Routing is the intermediate routing process that takes place between the originating LU and the destination LU. Network nodes handle the Intermediate Session Routing between the originating and destination LUs. By checking the status of ISR sessions, you can check the status of sessions routing through the node.

To obtain information regarding the current status of ISR sessions flowing through the local network node, enter:

```
SHow -APPN ISRsessions
```

A display similar to the following appears:

```
================================== SHow -APPN ISRsessions ============================
-------------------------------------ISR Sessions-----------------------------------
Originator CP name  COS name    Limit  Res    Primary     Link name     Secondary   Link name
                                                LFSID                    LFSID
US3COMHQ.IBM4       SNASVCMG    NO             010201      LINK0000      000201      @I000001
US3COMHQ.IBM4       #INTER      NO             010202      LINK0000      000202      @I000001
```

For more information on the headings in the ISRsessions display, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

Figure 10-19 is the basic concept of LU-LU sessions (routed through an intermediate node) and the relationship to CP-CP sessions. A session between two LUs spans from one endpoint LU to the other and is routed through an intermediate node. The process of routing LU-LU sessions through intermediate nodes is called *intermediate session routing*. The figure shows the relationship between the primary and secondary LUs, and the CP of the primary LU and the CP of the secondary LU, as viewed from network node A.

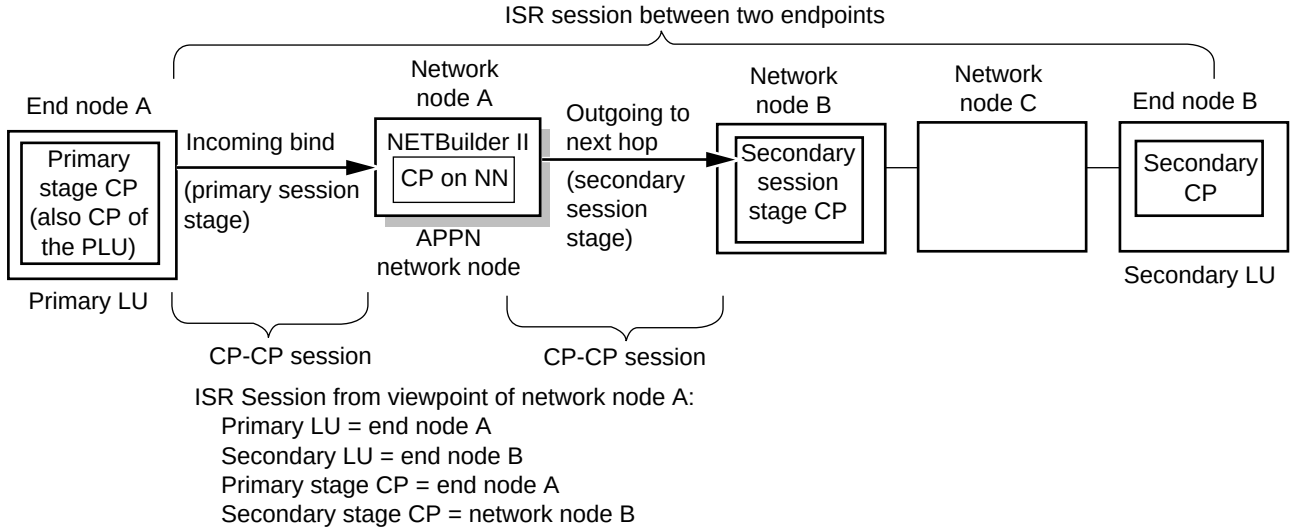


Figure 10-19 Intermediate Session Routing (Example 1)

The ISR session information differs depending on which network node you are viewing the session from. For example, Figure 10-20 is the same session example, but from the viewpoint of network node B. As shown in the figure, the primary and secondary CP information is different from the viewpoint of network node A. Also in Figure 10-19, end node A is the Primary Stage CP and also is the CP of the Primary LU. In Figure 10-20, because the network is now viewed from network node B, network node A is the Primary Stage CP, but end node A is still the CP of the Primary LU.

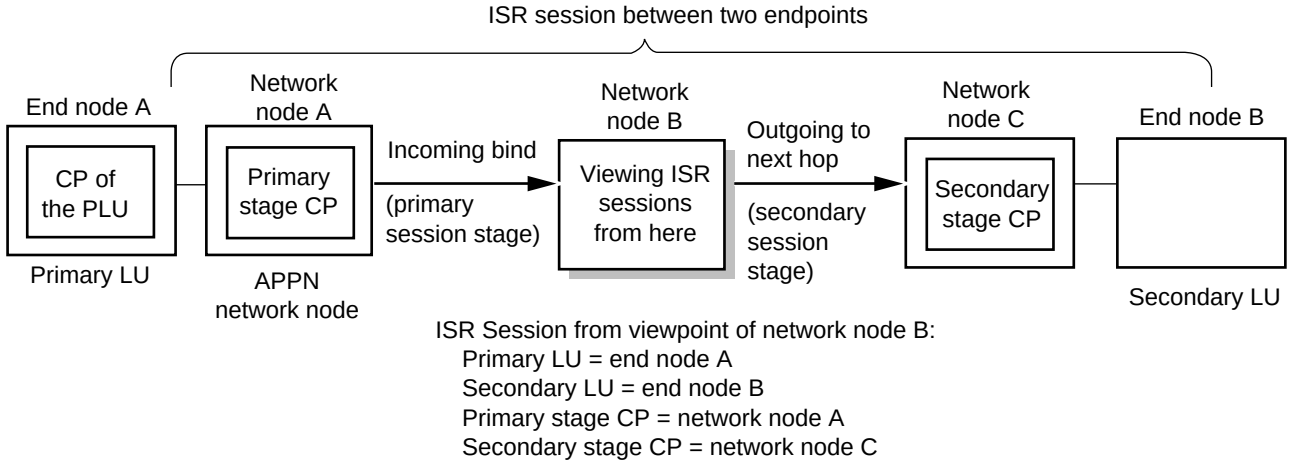


Figure 10-20 Intermediate Session Routing (Example 2)

How APPN ISR Routing Works

APPN ISR routing works differently from other protocol routing architectures. Unlike other protocols such as IP, you do not configure static routes using APPN. Instead, network nodes maintain a directory of LU resources (and more importantly, the location of the LU resources) available in their domains. When an originating LU requests a session to a destination LU, the location of that destination LU is discovered by checking the directories on the network nodes. The actual route is determined using the APPN class of service tables. For more information on how APPN class of service tables determine the best route to take in an APPN network, refer to Chapter 12.

This section describes the following major topics regarding APPN concepts and how APPN network nodes facilitate APPN routing:

- APPN node types
- Role of the network node
- How the network node learns about LU resources in its domain
- How the network node learns about LUs on other adjacent network nodes, and how this information is communicated among network nodes

APPN Node Types

This section describes briefly the different node types defined on an APPN network. For more detailed conceptual information, refer to the IBM document, *APPN Architecture and Product Implementations Tutorial* and the other documents listed in "IBM APPN References" on page 10-54.

Nodes in an APPN network are divided into the following three types:

- Network nodes
- End nodes
- Low-entry networking end nodes

Figure 10-21 is a sample of a network topology with different devices acting as different types of nodes.

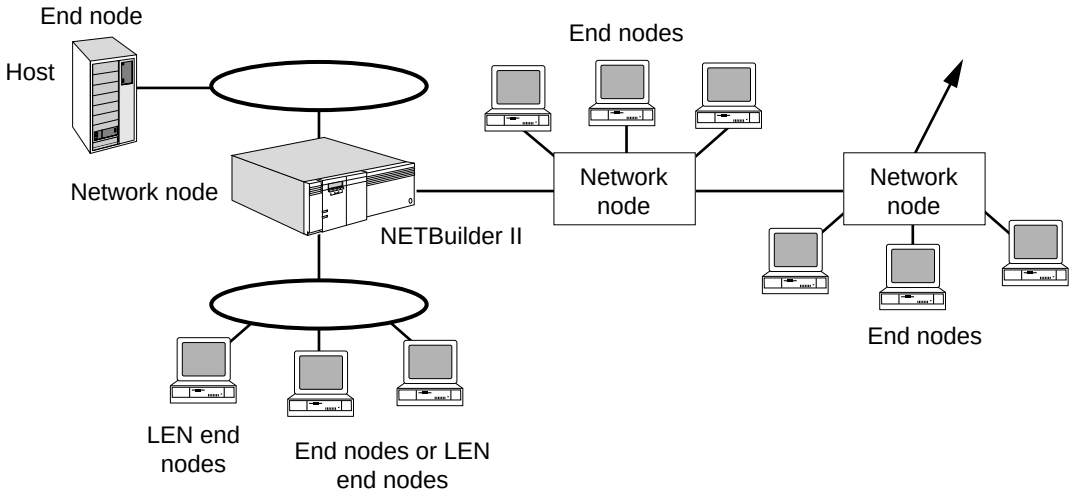


Figure 10-21 Node Types on an APPN Network

Network Nodes

Network nodes provide routing services and directory services for LUs on network nodes and end nodes. When a session request is initiated by a LU in the network node, an end node or LEN end node, the network node tries to locate the destination LU either on its own nodes or by querying other nodes. After the LU is located, the network node determines the best route to the destination LU according to the class of service for that session.

When a network node is added to an APPN network, the node learns network topology information from active adjacent network nodes. A network node exchanges network topology information with adjacent network nodes only when there is a change to the network topology.

When used in APPN networks, 3Com bridge/routers can function only as a network node, and does not support local LUs for application programs. Other devices that can serve as network nodes in an APPN environment include the following IBM platforms:

- IBM 6611
- S/36
- AS/400
- 3174 workstation controller (depending on the version; older versions may not be able to function as a network node)
- PCs running OS/2 Communications Manager
- IBM hosts running APPN protocols (VTAM with or without NCP supporting APPN)

This is not a complete list; other products may also be able to serve as network nodes.

End Nodes

End nodes provide limited directory and routing services for their local LUs. End nodes establish Control Point-to-Control Point (CP-CP) sessions with an adjacent network node so that LUs on the end node are available on the APPN network. The end node can also establish sessions to other LUs in the network.

The end node selects a network node to serve as its network node server and registers its LUs with the network node. By registering the end node's local resources with the network node, the network node can route any session requests from a remote node to the end node's LU. End nodes can have active connections to more than one network node at the same time, but only one network node can serve as the end node's network node server at one time.

Devices that can act as end nodes in an APPN environment include the following IBM platforms:

- AS/400
- PCs running OS/2 Communications Manager
- IBM hosts running VTAM

Low-Entry Networking End Nodes

Low-entry networking (LEN) end nodes are different from normal end nodes in that they cannot establish CP-CP sessions with a network node. As a result, LEN end nodes cannot register their resources with the network node; these resources must be predefined on the network node.

If the LEN end node has only one LU, then that LU is learned dynamically by the network node. However, if the LEN node has more than one LU, all LUs in addition to the first one must be statically defined in the network node's directory. For more information on defining LEN end node resources, refer to "Adding Entries" on page 10-23.

Many devices that are normally network nodes or end nodes can also be LEN end nodes, depending on how they are configured. Examples of devices that can be LEN end nodes in APPN networks include the following:

- IBM PCs running SAA Networking Services/2 or Networking Services/DOS (NS-DOS)
- IBM hosts running VTAM (depending on VTAM version and how it is configured)
- AS/400
- RS/6000 ANS Services/6000
- PCs running OS/2 Communications Manager

Non-IBM personal computers can also serve as LEN end nodes.

Differences Between Network Nodes and End Nodes

The primary difference between network nodes and end nodes is how each node type operates. Table 10-3 compares the basic differences between node types (note that LEN end nodes are a specific type of end node). For more detailed information regarding node type functionality, refer to the IBM document, *APPN Architecture and Product Implementations Tutorial,* in "IBM APPN References" on page 10-54.

Table 10-3 Functionality Differences Between Node Types

Capability	Network Nodes	End Nodes	LEN End Nodes
Ability to have CP-CP sessions	Yes	Yes	No
Dynamically learns LU locations	Yes	No	No
Maintains directory of LUs and their locations	Yes	No*	No
Performs intermediate session routing	Yes	No	No
Calculates session routes	Yes	No	No
Supports applications via LU interface	Yes†	Yes	Yes

* End node maintains a directory of its own LUs only.

† The NETBuilder II bridge/router provides only the routing function, and has no other application programs.

Network Node Role

The role of the network node is to provide the network services for the end nodes in its domain. It also provides the directory database that lists LUs in the local network node domain, so that the LUs can be discovered by other network nodes in the network.

The network node maintains a directory database of information for two types of nodes:

- End nodes (including LEN end nodes) in the network node's local topology
- Adjacent network nodes in the larger network topology

Figure 10-22 is the conceptual difference between the local topology and network topologies known by the network node.

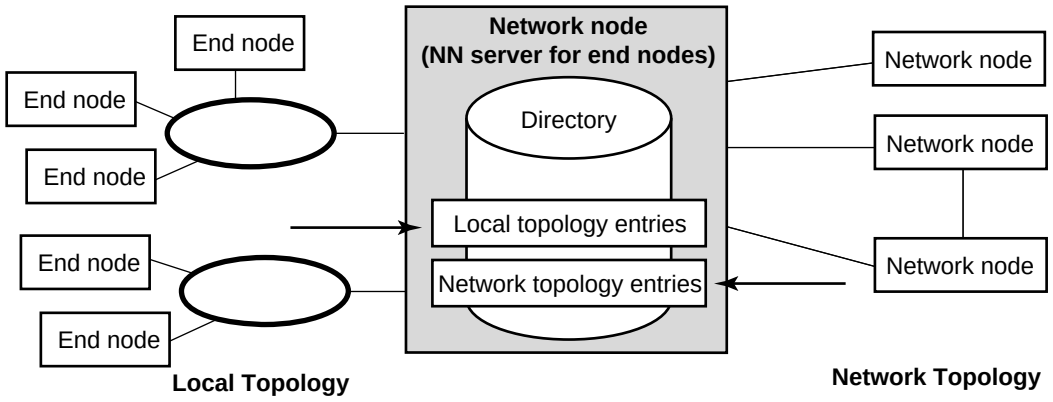


Figure 10-22 Local Topology and Network Topology for the Network Node

The network node acts as the network node server for the end nodes in its domain. The network node server provides the following services to end nodes:

- Distributed directory services

  These services locate network resources in the APPN network, and pass the information onto the end node.

- Routing services

  These services calculate the best route between the origin and destination LUs based on the required class of service. For more information on how class of service calculates routes, refer to Chapter 12.

An end node can have links to more than one network node. However, only one network node can act as the end node's network node server at one time.

The network node maintains two databases:

- Directory database

  These databases are LU resources on end nodes in the network node's local domain. These databases can be LUs on end nodes that were learned dynamically by the network node, or LUs on LEN end nodes that were statically defined in the directory.

- Topology database

  This database maintains information regarding all network nodes and the TGs between them. The network nodes and associated TGs together make up the APPN network backbone.

How the Network Node Directory Learns About Local End Node LU Resources

APPN is a point-to-point protocol, which means that links are established between two single partner nodes. The end node maintains a direct link to the network node server.

After you configure a link from the network node to an adjacent end node, the following processes can take place:

1. The end node calls on the network node to set up a CP-CP session (this does not apply to LEN end nodes).
2. The end node "registers" its LUs with the network node by sending information from the end node's local directory database to the network node.
3. After the network node receives the directory information, the directory entries are stored in the network node's local directory database.

   These entries are temporary entries in the network node's directory database, and will change depending on how resources change on the end node. For example, if a resource on the end node is added or deleted, the information is sent to the network node's local directory database to be updated.

   As long as the end node maintains a CP-CP session with the network node, the end node's resources will be registered in the network node's local database directory. After a CP-CP session is deactivated between the end node and the network node, the end node's registered entries in the network node's directory database are automatically deleted.

   Figure 10-23 illustrates how this process works.

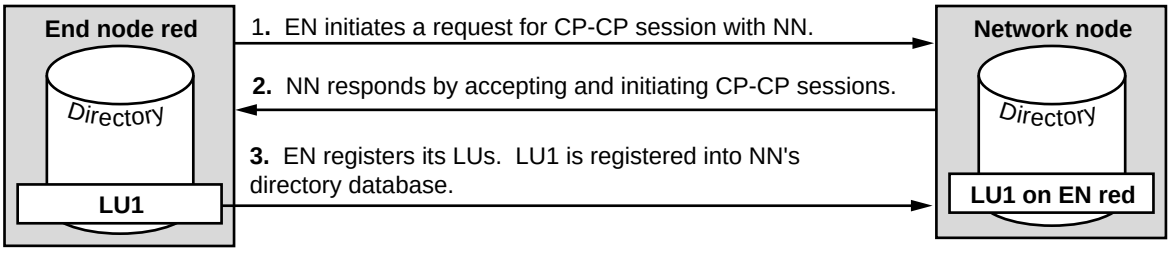


Figure 10-23 End Node Resource Registration into Network Node's Directory

How the Network Node Discovers the Location of Destination LUs

When end nodes initiate a request for a session with a destination LU, the end node requests that the network node allocate a session to the destination LU. The network node then consults the local directory database to discover if the LU is in its domain. If the destination LU is within the network node server's domain, the network node can send the session request directly to the destination LU. However, if the destination LU is on an end node or network node, the local node may send a Locate request first. When the local node receives a positive response to the locate request, it forwards the BIND request.

Figure 10-24 is an example of discovering the destination in the local topology. In this example, LU1 on end node "Red" wants to initiate a session with LU6. This example assumes that CP-CP sessions are up between end node "Red" and end node "Green." Based on the examples shown in the figure, the following steps take place:

1. LU1, the originating LU on end node "Red," initiates a Locate request to network node A, requesting the location of LU6, the destination LU.
2. The network node checks the local directory database for the location of LU6.
3. The directory database discovers a directory entry for LU6, which shows it is located on end node "Green," and that this end node is within the network node's local domain.
4. The network node sends a Locate request to "Green," verifying that LU6 is still available.
5. "Green" sends a "locate positive" response to the network node.
6. The network node forwards the response to "Red."
7. LU1 sends a BIND to LU6 to begin process for a logical unit-to-logical unit (LU-LU) session.

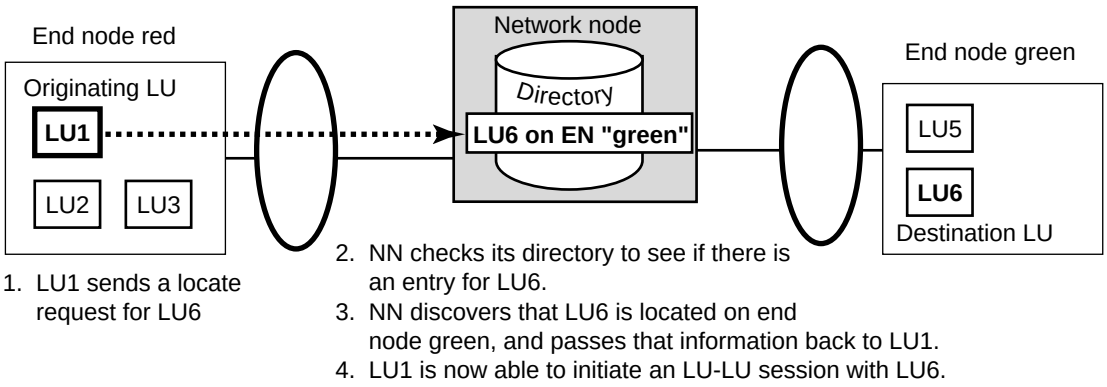


Figure 10-24 Discovering a Destination LU in the Local Directory

If the destination LU is not within the network node's local domain, the network node then sends locate requests to adjacent network nodes. These network nodes in turn check their directories to see if the destination LU is in their local domains.

Figure 10-25 is an example of discovering the destination LU in the larger network topology.

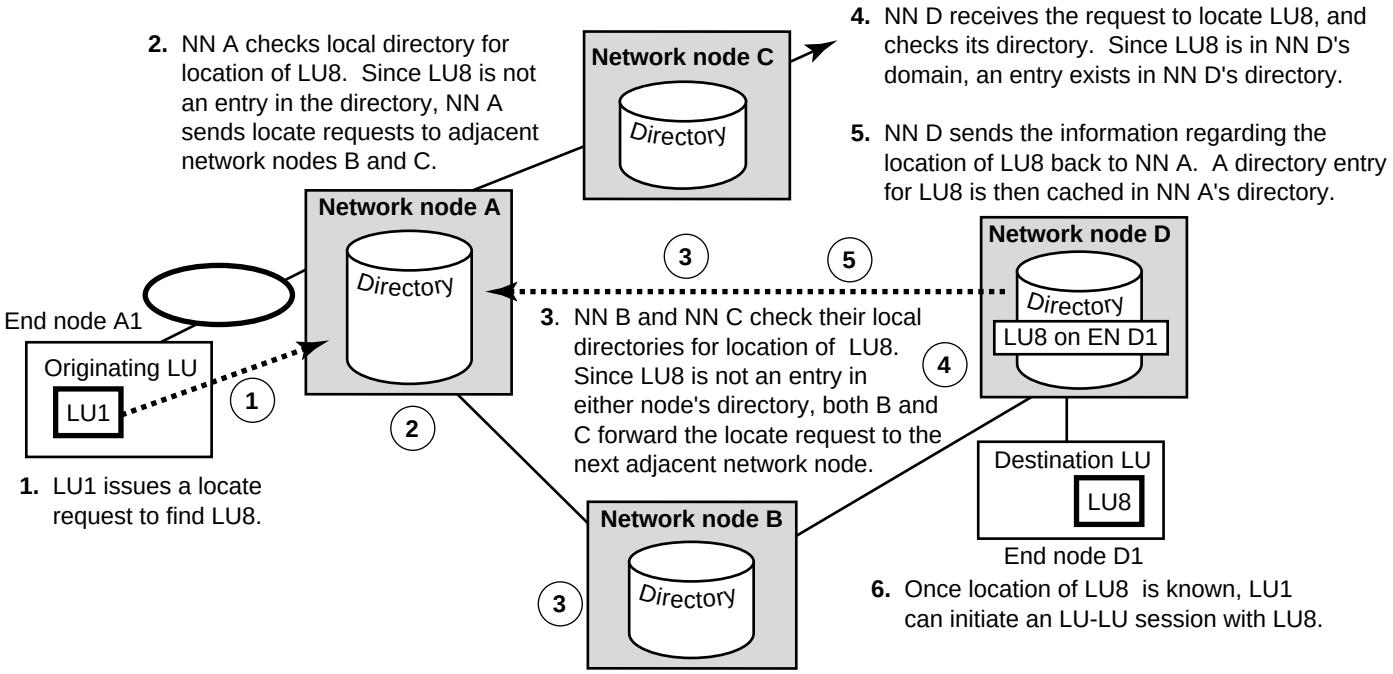


Figure 10-25 Discovering a Cross-Domain LU

In this example, since the destination LU is not within the local domain, the network node server sends Locate requests to adjacent network nodes. Those network nodes check their local databases for the destination LU, and if they do not find it, they in turn forward the Locate request to other adjacent network nodes. This process continues until the network node server for the destination LU finds the directory entry in its local directory database, and then forwards the information back to the first network node. The directory entry for the destination LU is then cached in the first network node's directory, in case that LU needs to be located again.

The process of locating LUs using directory services differs from the process of calculating the actual routes to those LUs. Routing is handled by topology and routing services, and routes are determined through the use of class of service tables. For more information on how class of service works, refer to Chapter 12.

Additional Information

This section provides the following additional information on some of the concepts and terminology for APPN:

- Fully qualified and not fully qualified CP name formats
- Canonical and noncanonical MAC address format options
- Setting the maximum BTU size
- APPN terminology

Fully Qualified and Not Fully Qualified CP Name Formats

When you configure the local network node, define adjacent link stations, and define adjacent nodes in the directory database, you enter the network name and CP name. However, there are different requirements and options for each. Figure 10-26 is the difference between fully qualified and not fully qualified CP name formats.

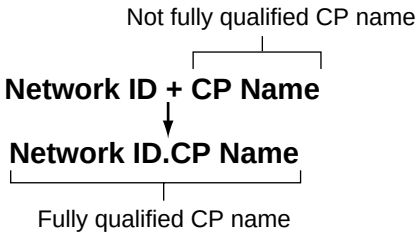


Figure 10-26 Fully Qualified and Not Fully Qualified CP Name Formats

Figure 10-27 shows the different options for entering the CP name and network ID depending on what you are configuring.

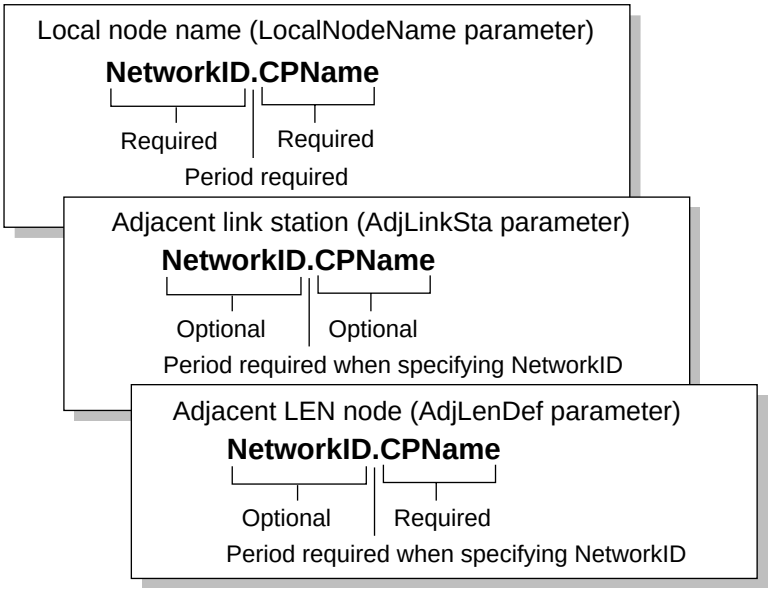


Figure 10-27 Comparison of CP Name Syntax Formats

If the adjacent network ID is not present, then the system assumes that the network ID is the same as the network ID for the local network node.

MAC Address Format Options for APPN

While most SNA environments normally use noncanonical MAC address format, the default setting for the NETBuilder II bridge/router is to use canonical format in entering and displaying MAC addresses. There are two options when setting up your system with noncanonical MAC address formats:

- Change the default MAC address format by entering

  ```
  SETDefault -SYS MacAddrFmt = Noncanonical
  ```

  If you change the default to noncanonical, you can enter MAC addresses for APPN in noncanonical format without special notation. For more information about this parameter, refer to Chapter 58 in *Reference for NETBuilder Family Software*.

- Change the default MAC address format to the Default setting by entering:

  ```
  SETDefault -SYS MacAddrFmt = Default
  ```

  If you specify Default, the system uses the appropriate MAC address for the port type. If the port type is token ring or FDDI, the system automatically displays and allows you to enter addresses in noncanonical format. All other port types would use canonical format. If you specify Default, you can still override it by preceding the MAC address with "NcMac," "Mac" or "Cmac" as described in the next paragraph.

- Precede the APPN MAC address with either "Ncmac" for noncanonical or "Mac" or "Cmac" for canonical.

  You have different options for using these prefixes with MAC addresses. Table 10-4 shows the available options. These options apply only to parameters in the APPN and SR Services.

Table 10-4 Options for Entering Canonical and Noncanonical MAC Addresses*

Canonical format where C=canonical†	Noncanonical format where N=noncanonical†
Cmac_XXXXXXXXXXX	Ncmac_XXXXXXXXXXX
Cmac_%XXXXXXXXXXX	Ncmac_%XXXXXXXXXXX
Cmac%XXXXXXXXXXX	Ncmac%XXXXXXXXXXX
C%XXXXXXXXXXX	N%XXXXXXXXXXX
C_%XXXXXXXXXXX	N_%XXXXXXXXXXX
C_XXXXXXXXXXX	N_XXXXXXXXXXX
CXXXXXXXXXXX	NXXXXXXXXXXX

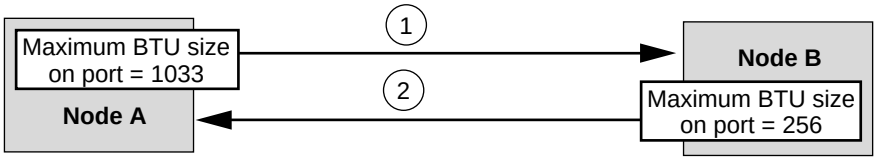
* If you do not set the -SYS MacAddrFmt parameter.
† Underscores indicate spaces.

Setting the Maximum BTU Size

When you configure adjacent link stations, one of the values you need to set is the maximum basic transmission unit (BTU) size. This value determines the maximum BTU size that will be allowed over the link.

When you configure partner nodes, each port on both sides of the link may have different maximum BTU sizes. During link station negotiation, each node communicates the maximum BTU size value it accepts. The lower of the two maximums is used to prevent the port with the lower capacity from over using its available memory capacity.

Figure 10-28 shows the process of how BTU size negotiation takes place.



① As part of session negotiation, Node A tells Node B the maximum BTU size allowed on port is 1033.
② In response, Node B informs Node A that its maximum port BTU size is 256.

As a result of negotiation, the session uses the smaller BTU size of 256 in both directions. The smaller Btu size is always used to prevent a port from receiving larger BTUs than it can handle.

Figure 10-28 BTU Size Negotiation

The maximum BTU size will differ depending on what physical medium is being used over the port. The recommended maximum BTU size allowed over APPN ports is 2,057, which equals the maximum ISR RU size of 4,096 + 9. (Certain media may allow larger frame sizes, but for the best buffer use on APPN ports, the BTU size should not be larger than 5,005.)

If the physical port medium is Ethernet, the recommended maximum BTU size is 1,500. If you are using serial lines for bridging LLC2, the recommended maximum BTU size for Source Route Transparent bridging is 1,500, while the recommended maximum for Source Route bridging is 5,005.

APPN Terms

A list of important terms that are used in this chapter is provided here to briefly explain APPN routing concepts.

adjacent link station	The local information regarding a link to an adjacent node. It is the link definition stored in the network node.
adjacent node	A node immediately adjacent to the local network node. You define adjacent nodes in the network node's directory.
connection network	A configuration in which a set of APPN nodes are grouped together with one logical name to help reduce the number of direct links required and the amount of broadcast traffic.
control point (CP)	An entity that manages T2.1 nodes and their resources. In APPN, the control point initiates links to adjacent nodes, and exchanges CP capabilities with adjacent nodes when CP-CP sessions are established.
control point-to-control point (CP-CP) session	Takes place between two adjacent nodes, to exchange routing and resource information, as well as the CP capabilities of the node. CP-CP sessions can take place between two network nodes, between a network node and an adjacent end node, and between two end nodes. (Note that LEN end nodes do not support CP-CP sessions.) Not all links can support CP-CP sessions.
dependent LU requester (DLUr)	Assists PU type 2.0 and 2.1 nodes with dependent LUs that require the services of a remote SSCP. The DLUr obtains these SSCP services from the dependent LU server (DLUs) and in turn provides the services to the dependent LUs. In the 3Com APPN implementation, the NETBuilder II bridge/router acts as the DLUr.

dependent LU server (DLUs)	A host that provides SSCP services to a dependent LU requester (DLUr).
directory	Resides on the network node and provides a list of logical units (LUs) on the local and network topologies, and the locations of those LUs. (Note that an end node also has a directory, but it only lists the end node's local LUs.)
end node	A node with LU resources that can initiate LU-LU sessions. A regular end node (as opposed to a LEN end node) can have CP-CP sessions with the network node acting as the end node's network node server. End nodes do not support intermediate session routing.
Intermediate Session Routing (ISR)	The routing that takes place through intermediate nodes between the originating LU and the destination LU. The NETBuilder II bridge/router acting as the network node provides intermediate session routing.
low-entry-networking (LEN) end node	A Type 2.1 node that does not have a control point. Without the control point, the LEN end node does not have the ability to hold CP-CP sessions with a network node. As a result, the network node cannot learn the LEN end node's LUs dynamically. The LEN end node's LUs must be statically defined in the network node's directory.
logical unit (LU)	Provides an interface for applications to communicate and gain access to an SNA network. The network node learns LUs on adjacent network nodes and end nodes dynamically, while LUs on LEN end nodes must be statically defined in the directory.
network node	The backbone of the APPN routing architecture. Network nodes provide intermediate session routing between two end stations, exchange directory and topology information with adjacent network nodes, and provide routing services for end nodes in their domain.
network node server	The network node that "serves" the end nodes in its domain by maintaining a list of LUs on the end nodes, so that incoming LU requests can find the location of a destination LU. The network node calculates routes for LUs on the end nodes.
partner node	Two adjacent nodes that have configured each other as adjacent link stations so they can have links with each other. (Partner node is not an IBM APPN term, and is used here for conceptual purposes only. The term is not to be confused with the IBM terminology for partner LUs.)
transmission group	A link between two nodes.
virtual routing node	A logical representation of a defined connection network between two nodes.

IBM APPN References

The following IBM documents provide additional information for IBM's APPN and SNA implementation:

APPN Architecture and Product Implementations Tutorial, International Business Machines Corporation, April 1994 (GG24-3669-02)

IBM Systems Network Architecture: LU6.2 Reference: Peer Protocols, International Business Machines Corporation (SC31-6808)

IBM Systems Network Architecture: APPN Architecture Reference, International Business Machines (SC30-3422)

IBM Systems Network Architecture: Management Services Reference, International Business Machines (SC30-3346)

IBM Systems Network Architecture: Formats, International Business Machines (GA27-3136)

IBM Systems Network Architecture Concepts and Products, International Business Machines (GC30-3072)

IBM Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2, International Business Machines (SC30-3269)

11 APPN HIGH PERFORMANCE ROUTING

This chapter describes how to configure your 3Com bridge/router to perform Advanced Peer-to-Peer Networking (APPN) High Performance Routing (HPR).

High Performance Routing is an advanced method of routing APPN sessions that provides greater scalability and performance than Intermediate Session Routing (ISR), the original APPN routing method. The improvements that HPR provides over ISR include:

- Dynamic rerouting if a link on a path fails, which enables the connection to stay up.
- Streamlined routing at the Systems Network Architecture (SNA) Path Control layer.
- An architecture designed to take advantage of high-speed media.
- Intermediate nodes do not have to process message segmentation, which reduces overhead.
- Intermediate nodes do not buffer messages, which reduces overhead.

You can configure ports on the NETBuilder II bridge/router to perform either ISR or HPR. By default, HPR is enabled on APPN ports and adjacent link stations. If you want to configure specific ports or link stations for ISR, you must disable HPR on those ports or link stations. For more information about configuring the bridge/router as an APPN network node for ISR, refer to Chapter 10.

For conceptual information about HPR, refer to "How HPR Works" on page 11-7.

APPN routing is supported only on NETBuilder II bridge/routers that include a DPE module.

Configuring the Network Node to Perform HPR

HPR networks operate over network nodes and end nodes like ISR networks. The NETBuilder II bridge/router can be configured as a network node only; because the bridge/router does not provide any application programs on the SNA network, it cannot act as an end node or LEN end node.

When you configure the NETBuilder II bridge/router as an HPR network node, it can function as a Rapid Transport Protocol (RTP) tower node. For an explanation of RTP tower nodes and the other types of HPR network nodes, refer to "HPR Node Types" on page 11-7.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to the procedures described in Chapter 1.

- If necessary, use LAN Address Administration (LAA) to reassign media access control (MAC) addresses for paths that will be sending and receiving APPN traffic.

  You must perform this configuration *before* starting APPN. For more information on configuring LAA, refer to Chapter 28.

- If you are planning to support both APPN and DECnet on the same bridge/router, you must configure DECnet *before* configuring APPN. Configuring DECnet can change MAC addresses, which would affect any existing APPN configuration. For more information on configuring DECnet, refer to Chapter 15.
- If necessary, configure the Logical Link Control, type 2 (LLC2) data link interface or the Data Link Switching (DLSw) interface for the ports you will use for APPN traffic. For more information on configuring the LLC2 data link interface, refer to Chapter 21. For more information on configuring DLSw, refer to Chapter 24.
- If you will be sending APPN traffic over synchronous data link control (SDLC) lines, configure the bridge/router for SDLC operation first. For more information on SDLC configuration, refer to Chapter 22.
- If you will be sending APPN traffic over Frame Relay, configure the Frame Relay interface before configuring the APPN network node. For more information on configuring Frame Relay, refer to Chapter 42.
- Configure the NETBuilder II bridge/router as an APPN network node following the procedures in Chapter 10. Using those procedures, you first set up the basic framework for your APPN configuration using ISR. To bring the APPN network node to the HPR level, follow the procedures in this chapter.

Procedure To set up the bridge/router network node to perform HPR, follow these steps:

1 Set your APPN ports to support HPR by performing one of the following steps:

a If you did not disable HPR when first setting up the ports as described in Chapter 10, you need not change anything. The port will already support HPR.

b If you are converting a port from ISR mode to HPR, set the APPN port to perform HPR using:

```
SETDefault !<port> -APPN PortDef = <DLC type>
 (LLC2|FR|PPP|DLSW|SDLC|UNdef) <max_btu_size>(99-8192)
 [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

Make sure to specify HPR=Yes, and specify whether you want the port to provide link level error recovery. You can specify the other optional values of the PortDef parameter as desired.

Setting ErrorRecovery to Yes provides link-level error recovery for the incoming connection only. If you want link-level error recovery for the outgoing connection, then you must specify Yes for the ErrorRecovery value when setting the AdjLinkSta or SdlcAdjLinkSta parameter. If you set the port data link control (DLC) type to DLSW or SDLC, then link-level error recovery is enabled by default. If you set the port DLC type to LLC2, FR, or PPP, then you must specify error recovery support if desired.

If you use error recovery, it will create additional overhead on the link.

To configure port 7 for Frame Relay at a maximum basic transmission unit (BTU) size of 1033 and to enable support for HPR and error recovery, enter:

```
SETDefault !7 -APPN PortDef = FR 1033 HPR=Yes ErrorRecovery=Yes
```

2 Define adjacent link stations for HPR by performing one of the following steps:

a If you did not disable HPR when first setting up the adjacent link station as described in Chapter 10, you need not change anything. As defined on the bridge/router, the link station will already support HPR.

b Define each adjacent link station to support HPR using:

```
ADD !<port> -APPN AdjLinkSta <type>(NN|EN|Learn)
 <max_btu_size>(99-8912) [[Cmac|Ncmac] dest media addr]
 [Sap=<num>] [CPName=[netid.]cpname] [Nodeid=<ID>]
 [LinkName=<name>] [TGprof=<name>] [AutoStart=(Yes|No)]
 [CPSess=(Yes|No)] [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)]
```

Make sure to specify the adjacent link station as an HPR node by specifying HPR=Yes. If you want link level error recovery for the outgoing connection, specify ErrorRecovery=Yes. You can specify the other optional values as desired.

c If using SDLC, define each adjacent link station to serve as an HPR node using:

```
ADD !<port> -APPN SdlcAdjLinkSta <type>(NN|EN|Learn)
 <max_btu_size>(99-8912) <station addr>(Hex 1-FE)
 [CPName=<[netid.]cpname] [Nodeid=<ID>] [LinkName=<name>]
 [TGprof=<name>] [AutoStart=(Yes|No)] [CPSess=(Yes|No)]
 [HPR=(Yes|No)] [ErrorRecovery=(Yes|No)] [SendWindow=<num>]
 [ContactTimer=<num>] [NoRspTimer=<num>] [NoRspTimRetry=<num>]
```

Make sure to specify the adjacent link station as an HPR node by specifying HPR=Yes. If you want link level error recovery for the outgoing connection, specify ErrorRecovery=Yes. You can specify the other optional values as desired.

When you configure HPR over SDLC connections, HPR must be enabled on both sides of the SDLC connection.

For more information on the full syntax of these parameters, refer to Chapter 5 of *Reference for NETBuilder Family Software*.

APPN over SDLC connections is supported on the NETBuilder II HSS-3-Port V.35 module only.

3 If you have not done so already, define the link characteristics using:

```
SETDefault -APPN LinkStaCHar = <LinkStation name>
 [EffectCap=<string>] [ConnectCost=<0-255>] [ByteCost=<0-255>]
 [Security=<string>] [PropDelay=<string>] [Usd1=<0-255>]
 [Usd2=<0-255>] [Usd3=<0-255>]
```

4 To enable the bridge/router to function as an APPN network node, enter:

```
SETDefault -APPN CONTrol = Enable
```

After HPR has been configured on the network node and the node has been enabled, the network node can participate in the HPR network. If the network node is part of an ISR environment, an HPR subnet can be created. Depending on how you set up your network, the network node can be either an HPR

endpoint (for Rapid Transport Protocol (RTP) connections) or an HPR intermediate node (for Automatic Network Routing). An HPR node does not become an RTP endpoint until it accepts a session through it.

Not all devices that support APPN support HPR. If you configure an adjacent link station to a device that does not support HPR, RTP connections cannot take place, but ISR sessions can.

Configuring HPR Subnets within ISR Networks

After you have configured two adjacent network nodes for HPR, and they have established the appropriate RTP connection, an HPR subnet is created, even if the two nodes reside within an APPN ISR network. When you create a mixed HPR and ISR environment, you only gain the benefits HPR provides on those links where both partner nodes support HPR. On links where one node is HPR-capable and the partner node is not, the link defaults to normal APPN ISR operation.

Figure 11-1 is an example in which two network nodes within an APPN ISR network have been upgraded to support HPR. In this example, only the links between the HPR-capable nodes support HPR operation, including RTP connections. Since there are alternate paths that are ISR only, you will not gain the benefits that HPR provides because the topology routing services and class of service used to calculate routes do not provide greater weight to the HPR paths over ISR paths. As a result, mixing HPR nodes and ISR nodes in this type of configuration is not recommended.

You can create customized class of service tables to prioritize HPR paths over ISR paths, but you will need to do extra configuration. For more information on customizing class of service, refer to Chapter 12.

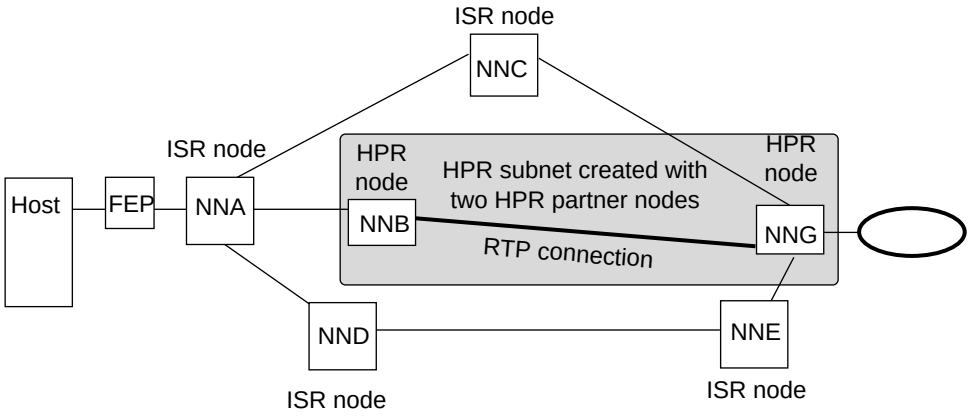


Figure 11-1 HPR Subnet within APPN ISR Network

Figure 11-2 is an example in which HPR nodes and ISR nodes are mixed over a path. In the top example, two HPR nodes form an HPR subnet, and an ISR node is between two HPR nodes. The result is that the third HPR node is isolated, and the benefits of HPR are limited to only the HPR subnet. In the bottom example, the ISR node is converted to support HPR, which chains all four HPR nodes together to form a larger HPR subnet, extending the benefits of HPR over a larger portion of the network.

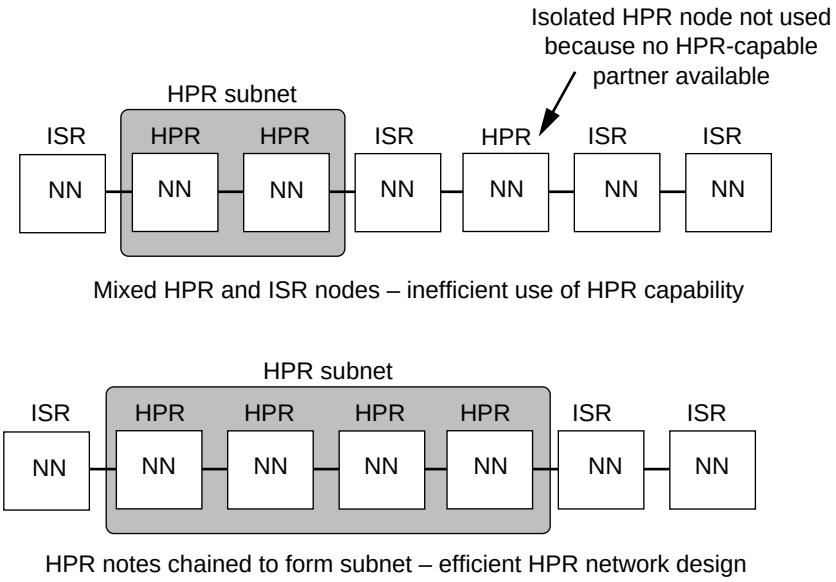


Figure 11-2 Chaining HPR Nodes to Form HPR Subnet

3Com recommends that you design your network so that HPR nodes reside either:

- On the network backbone between APPN ISR segments.
- Within a self-contained HPR subnet in which there are no alternate ISR paths.

By configuring HPR nodes within a larger HPR subnet on the network backbone, you gain the high-speed benefits and processing efficiencies that HPR provides. Figure 11-3 is an example of an HPR subnet used on a network backbone connecting multiple ISR networks.

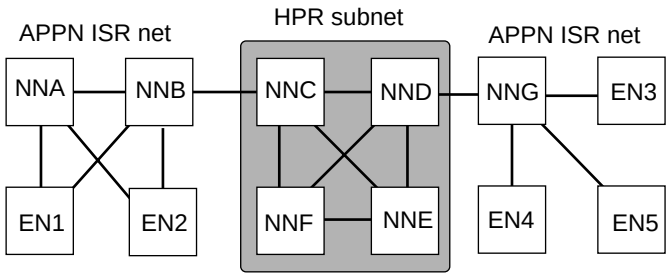


Figure 11-3 HPR Subnet on a Network Backbone

Using HPR with Boundary Routing Environments

You can configure HPR with Boundary Routing so that the NETBuilder II bridge/router acting as an HPR node is also acting as the central node in the Boundary Routing topology. Because the Superstack II NETBuilder bridge/router, acting as the leaf node at the remote site, does not support APPN in either ISR or HPR mode, the NETBuilder II bridge/router at the central site provides the HPR boundary function (to translate ISR traffic to HPR and vice-versa).

Figure 11-4 is an example in which HPR is configured with Boundary Routing at a remote site where an APPN connection network has also been configured. In the example, network node A (the central node) is an RTP tower node for HPR and is maintaining RTP connections with network node B. The central node is

also providing the boundary function to the APPN end nodes at the remote site. For more information about Boundary Routing system architecture, refer to Chapter 32.

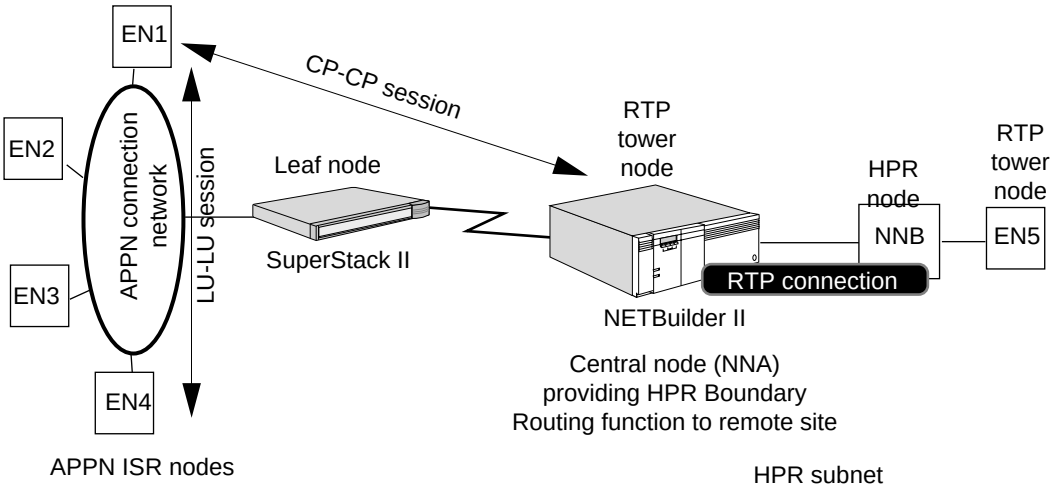


Figure 11-4 HPR and Boundary Routing

Operating the HPR Network Node

After the network node has been configured for HPR, you can perform tasks such as setting RTP connection timers and initiating nondisruptive path switching.

Setting RTP Connection Timers

To set the timers for RTP connections, use:

```
SetDefault -APPN HprTimer = [AliveTimer=<30-600>]
 [PathSwitchTimerLow=<240-960>][PathSwitchTimerMed=<120-480>]
 [PathSwitchTimerHigh=<60-240>][PathSwitchTimerNtwk=<30-120>]
```

Using this command, you set the timer settings for the RTP connection. Changing this parameter only affects new RTP connections, and has no effect on existing RTP connections. The options allow you to set the path switch timer for connections with low, medium, high, or network priority.

Displaying RTP Connections

To display a list of RTP connections, use:

```
SHow -APPN RTP [name]
```

To display statistical information regarding RTP connections, use:

```
SHow -APPN RTPStats [name]
```

For information about the contents of these displays, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

Initiating a Nondisruptive Path Switch

A nondisruptive path switch can be triggered for several reasons, such as a local or remote link failure, or an RTP connection failure detection. When one of these failures occurs, the system initiates the path switch to try to determine if an alternate path is available. Normally, this nondisruptive path switching occurs automatically.

Using the PathSwitch command, you can request that the system switch an RTP connection to an alternate path. When a path switch is initiated, the system checks all available paths in the HPR topology to determine if a more desirable path is available. If a more desirable path is available, the RTP connection switches to that path; if the current path is the most desirable, then the system remains at the current path.

To initiate a nondisruptive path switch, enter:

```
PathSwitch <RTP name>
```

You must specify the RTP connection name that you want the system to switch.

To obtain a list of RTP connection names, enter:

```
SHow -APPN RTP
```

You cannot specify the new path to switch to; the system determines which path to switch to.

You can only switch paths from one HPR path to another; you cannot switch an RTP connection to a path running APPN ISR traffic.

For more information about nondisruptive path switching, refer to "Nondisruptive Path Switching" on page 11-10.

How HPR Works

High Performance Routing is designed to work in conjunction with APPN Intermediate Session Routing (ISR) network nodes. HPR nodes perform many of the same functions as ISR nodes. For example, HPR nodes use the same method of calculating routes based on the Topology Routing Service database and class of service tables. HPR nodes also supports such APPN features as connection networks and support for parallel transmission groups (TGs).

In the HPR architecture, both partner nodes must support HPR for RTP connections to take place between the nodes. If one node supports HPR and the partner node does not, then the link will support ISR functionality only.

For more complete information regarding HPR, refer to the IBM document *APPN Architecture and Product Implementations Tutorial* (GG24-3669-92).

HPR Node Types

There are two different levels of HPR node functionality:

- Base HPR node

  Base HPR nodes support Automatic Network Routing (ANR) and can only act as intermediate nodes in an RTP connection. Base HPR nodes cannot be the endpoint of an RTP connection. The 3Com bridge/router cannot act as a base HPR node.

- RTP Tower node

  RTP tower nodes can be either RTP endpoints or RTP connection intermediate nodes performing the ANR function. When acting as RTP endpoints, RTP tower nodes perform adaptive-rate-based flow control. If the RTP tower node is connected to an APPN ISR subset, then it performs the boundary function, which joins a session in the HPR subnet with a session in the APPN ISR subnet. The 3Com bridge/router network node acts as an RTP tower node in the HPR network.

Figure 11-5 shows the relationship of the different node types in an HPR network.

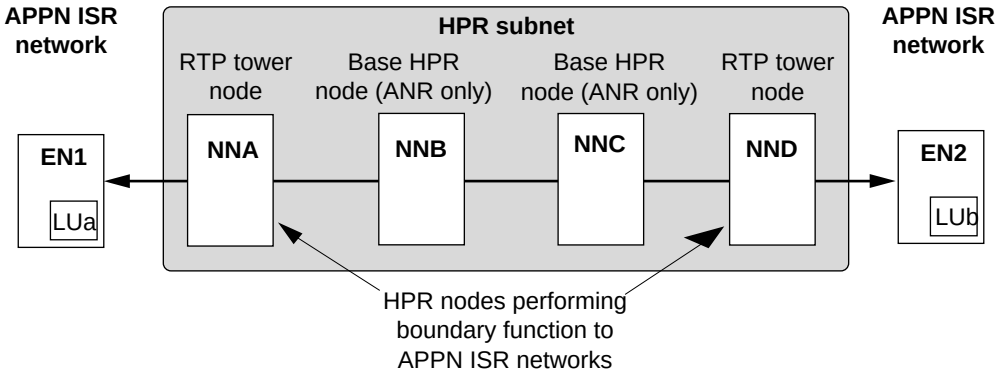


Figure 11-5 HPR Node Types

IBM Devices Supporting HPR

For the APPN HPR network node to provide HPR functionality on a link, the partner node device must also support HPR. IBM devices that support HPR include:

- VTAM V4R3/NCP V7R3
- OS/400 V3R1 (ANR only)

This list is not complete, and other devices may support HPR in the future. HPR can be supported on APPN network nodes and end nodes.

Automatic Network Routing

Automatic Network Routing is a source routing protocol used to route LU6.2 session and control traffic from node-to-node through an HPR network or subnet. ANR operates at the lower end of the SNA Path Control layer.

Unlike most SNA traffic, which is normally connection-oriented, ANR packets are connectionless, and HPR routes these network layer packets independently. These packets contain a network layer header that carries routing information. Because the routing information is processed at the network layer, this processing is more efficient than the processing for ISR packets.

The routing information is contained in the ANR routing field, which consists of a string of ANR labels. Each label describes the path from one node to the next immediate node; the ANR label string describes the path from the source HPR node to the destination HPR node of the RTP connection.

When an HPR node receives an ANR packet, it checks the first label of the ANR routing field and uses that label to determine which link to send the outgoing packet over. That label is then stripped from the ANR routing field, so the receiving node can check the next label in the ANR routing string.

Figure 11-6 shows how ANR routes network layer packets and how ANR labels are used to route the packets from node-to-node, and then are stripped when they are no longer needed. In the figure, the ANR label from network node A to network node D is A0-A0-85, and at each intermediate node the first part of the label is stripped from the packet.

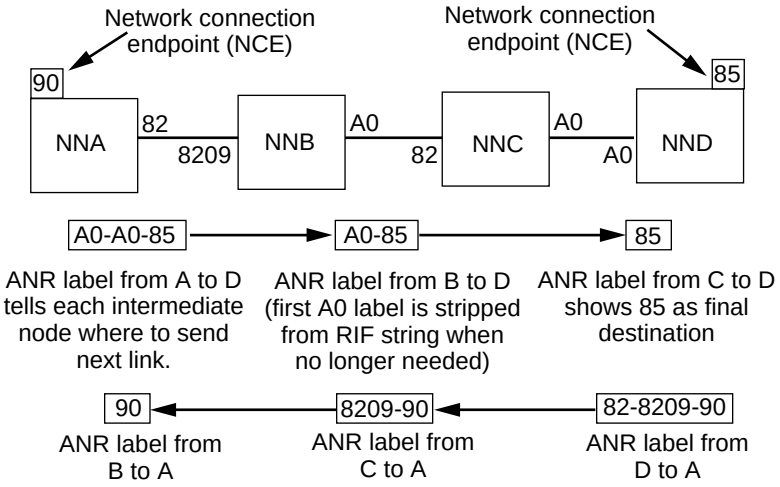


Figure 11-6 ANR Label Processing

The ANR label is from 1 to 8 bytes long and is of local significance on the node only. ANR labels only need to be unique on the local node, not on the larger network. In the figure, the label A0 is used several places, but the duplication is acceptable as long as the A0 label is unique on each node. In addition, ANR labels can be of different sizes within a node.

Rapid Transport Protocol

Rapid Transport Protocol is a reliable connection-oriented protocol that HPR uses to carry session traffic through an HPR network. It routes logical unit-to-logical unit (LU-LU) session traffic flows between the two RTP connection endpoints using the ANR routing method. RTP provides the following features:

- Full duplex transmission and delivery of messages in sequence
- Message segmentation and reassembly
- Selective retransmission, in which only the portions of data that are lost are retransmitted
- Adaptive-rate-based congestion and flow control

RTP Connections

RTP connections are logical connections between two nodes over a specific path in an HPR network. These logical connections are used to transport full-duplex session traffic end-to-end between the two nodes. RTP connections support a single class of service on each connection, enabling all traffic on the connection to use the same transmission priority. You can multiplex multiple sessions of the same class of service over one RTP connection, but all traffic on a given session must flow on the same RTP connection. If you have multiple sessions with different classes of service, then the bridge/router uses different RTP connections for each class of service.

Figure 11-7 is an example of an RTP connection across several nodes in an HPR subnet. LUa on EN1 is connected to LUb on EN2. In between the two end nodes is an HPR subnet, with the RTP connection spanning across it.

You can have multiple sessions with the same class of service on an RTP connection. All traffic using a specific class of service travels on the same RTP

connection (the 3Com HPR implementation does not support sending traffic of the same class of service over different RTP connections). The figure also shows the ANR routing packets being forwarded from node-to-node.

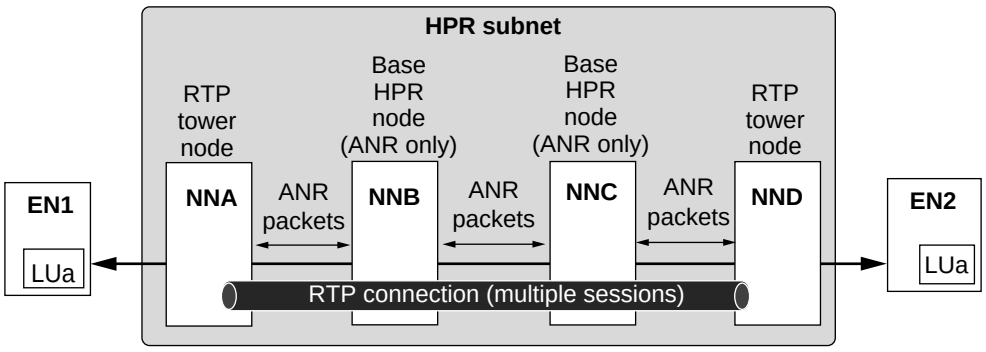


Figure 11-7 RTP Connection Across HPR Subnet

You can have multiple RTP connections between HPR nodes, with each RTP connection handling a different class of service. In Figure 11-8, there are multiple RTP connections. One RTP connection is used for batch sessions (using the BATCH class of service), and one RTP connection is used for interactive sessions (using the INTERACTIVE class of service).

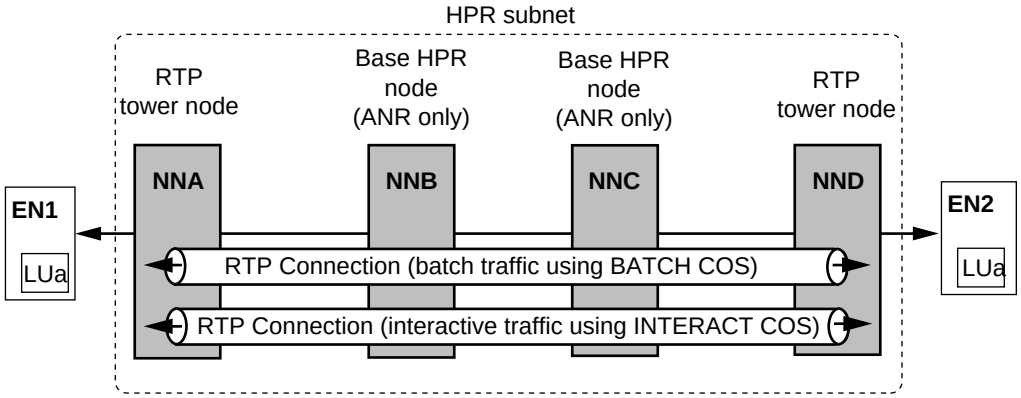


Figure 11-8 Multiple RTP Connections Using Different Classes of Service

Nondisruptive Path Switching

Through RTP, HPR provides nondisruptive path switching, which enables the node to switch an RTP connection to a new path if the current path fails or the link fails. When the system initiates a path switch, it attempts to switch the RTP connection to the most desirable path at the time. This process enables dynamic rerouting in case of link failure, and the rerouting takes place fast enough not to disrupt the active sessions. The most desirable path at the time is the HPR-only route with the lowest weight. Even if there is an alternative ISR-only path that has a lower weight than the lowest-weight HPR route, the lowest weight HPR route is chosen.

The node can trigger a path switch in one of the following situations:

- RTP connection failure detection

  When an RTP endpoint periodically sends out a status request to its partner, if a reply is not received within the specified time set by the HprTimer parameter, the RTP endpoint sends a state exchange request to determine the status. If this state exchange fails after several retries, then the RTP endpoint determines that the RTP connection failed and triggers a path switch.

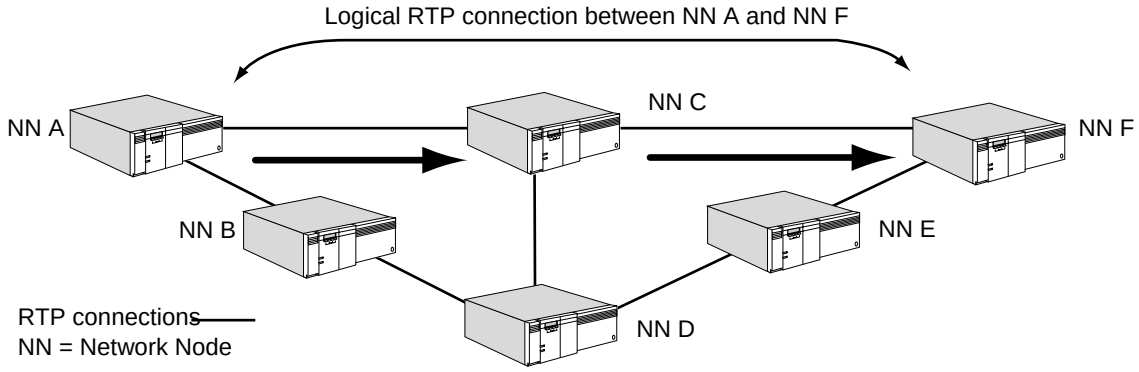
- Local link failure

  If a local link associated with an RTP connection fails, the system can initiate a path switch faster than relying on the RTP connection failure detection.

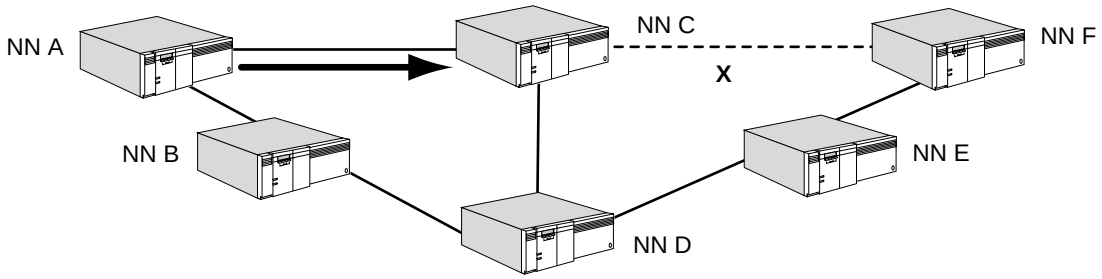
- Initiated by the user using the PathSwitch command.

  When a user initiates a path switch, the system checks to determine the most desirable path to switch the RTP connection to. If the current path is the most desirable path, the system remains at the current path.

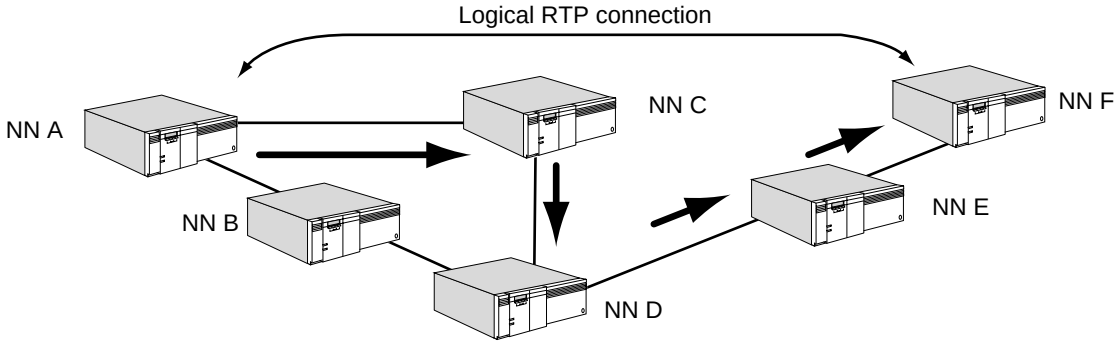
Figure 11-9 is an example where nondisruptive path switching takes place. In the figure, all network nodes shown are in an HPR network. There is an RTP connection between network node A and network node F, and network node C serves as an intermediate node on the path. When the link between node C and node F goes down and the connection times out, a nondisruptive path switch is triggered from network node A. Network node A uses Topology Routing Services (TRS) to determine the best alternate path. The least cost alternate path is the one that goes to network node C, then through network nodes D and E, and then to node F. The logical RTP connection remains up, even though one of the original links failed.



1. RTP connection between NN A and NN F is up.



2. Link between NN C and NN F goes down. NN A tries a path switch, trying to find the best alternate path.



3. NN A determines best alternate path using TRS and conducts a path switch to the new path. RTP connection stays up.

Figure 11-9 Nondisruptive Path Switching (Example)

Adaptive Rate Pacing

Adaptive-rate-based congestion and flow control is a mechanism for RTP endpoints to regulate the amount of traffic entering the HPR network. This method determines if the network is congested based on the rate of traffic entering the network, the rate of traffic leaving the network, and the buffer situation of the receiving RTP endpoint. Attempts are made to allocate equal bandwidth to all RTP connections over a link that is shared by the RTP connections. However, sessions sharing one RTP connection require individual session pacing to ensure that any one session does not occupy the whole RTP connection.

Comparison of ISR and HPR Functions

Table 11-1 lists APPN features supported on ISR nodes (base APPN) and features supported on HPR-capable nodes. This information applies to the 3Com implementation of both HPR and ISR.

Table 11-1 Comparison of ISR and HPR Function Support

APPN Feature	Support on ISR Network Node (Base APPN)	Support on HPR Network Node
DLUr support	Yes	Yes
APPN over SDLC links	Yes	Yes
Routing over WANs (not supported on ATM)	Yes	Yes
Parallel TGs	Yes	Yes
DLSw links between network nodes (for HPR, both network nodes must support HPR)	Yes	Yes
APPN in Boundary Routing topologies	Yes	Yes
APPN connection networks	Yes	Yes
Route calculation using class of service*	Yes	Yes
Rapid Transport Protocol support	No	Yes
Automatic Network Routing	No	Yes
Nondisruptive path switching	No	Yes
Adaptive-rate-based congestion control	No	Yes
Link level error recovery	Required†	Not required†

* Route calculation operates the same for both ISR and HPR nodes. By default, no priority is given to paths between HPR nodes vs. paths between ISR nodes.

† APPN ISR uses LLC2 to provide link level error recovery. HPR provides the option of not using link level error recovery, which reduces CPU processing overhead on intermediate nodes.

12 CONFIGURING APPN CLASS OF SERVICE

This chapter describes the Advanced Peer-to-Peer Networking (APPN) class of service (COS) and how it is used to calculate routes in an APPN network.

The class of service database exists in all APPN network nodes and helps determine how traffic is routed within the APPN network. This database determines how sessions are routed based on such characteristics as transmission priority, security levels, line speed, propagation delay, and resistance (the desirability of routing on the node).

Levels of class of service are used because different applications have different response time and throughput requirements. For example, interactive applications (such as a session between a terminal user and a host) normally require faster data transmission and consistent response times, while batch file transfers require high throughput and are not response-time oriented.

The same method of calculating routes based on class of service is used for both Intermediate Session Routing (ISR) and High Performance Routing (HPR) traffic. However, using the default class of service tables, no special priority is given for HPR links over ISR links.

Default SNA Class of Service Modes

IBM has created a set of Systems Network Architecture (SNA)-defined mode names and corresponding class of service names that are applicable to the vast majority of user environments. When end stations issue a session request to the network node using the IBM defaults, the network node maps the mode name to one of the default classes of service. The default COS definitions are preconfigured in your bridge/router so you do not need to perform any configuration to use them.

Table 12-1 lists the IBM default mode names and corresponding class of service names. In the table, the pound character (#) is equivalent to the hex value X'7B' as defined in the IBM architecture documents. For more information on this value, refer to the IBM documents, *Systems Network Architecture Type 2.1 Node Reference* and *Systems Network Architecture LU 6.2 Reference: Peer Protocols.* For the contents of the default SNA class of service tables, refer to "Default Class of Service Tables" on page 12-10.

Table 12-1 SNA Default Mode Names and Corresponding Class of Service Names

Mode Name	Class of Service Name	Transmission Priority
blank (no characters entered)	#CONNECT	Medium
#BATCH	#BATCH	Low
#BATCHSC	#BATCHSC	Low
#INTER	#INTER	High
#INTERSC	#INTERSC	High
CPSVCMG	CPSVCMG	Network
SNASVCMG	SNASVCMG	Network
CPSVRMGR*	SNASVCMG	Network

* This mode is used only for the CP-SVR pipe for sessions between a DLUr and DLUs.

A session request may include a Class of Service/Transmission Priority Field (COS/TPF). If a session request includes a COS/TPF and the network node knows about it (if it is one of the IBM defaults or has been defined on the network node), the network node processes the request with the COS specified in the COS/TPF. If the network node does not know the COS, then it uses the mode name to map one. If the network node does not know about the mode name, the session request will be rejected. Some implementations default to #CONNECT if the network node does not know the mode name. If the session request does not have the COS/TPF, then the network node tries to map it; if the network node cannot map it, the session request will be rejected.

To accept nonstandard modes from the end node, a class of service name must be mapped to the mode name. For information on mapping mode names to customized class of service names and creating customized class of service tables to meet specialized needs, refer to "Creating Customized Class of Service Tables" next.

Creating Customized Class of Service Tables

When you use customized class of service tables, you have more flexibility in determining how your network handles load balancing among different paths. You can also set prioritization of sessions and control response time. Customized COS definitions can also be useful for larger networks handling greater numbers of sessions. If a customized class of service mode name has been created on one of your end stations, then you also want to define the class of service on the network node.

If an end station issues a session request with a nonstandard mode, then a customized class of service must be created to handle that mode. If the nonstandard mode is not defined on the network node, the network node will use the default class of service for unknown mode names, which is #CONNECT.

To add a customized class of service to the bridge/router, follow these steps:

1. Determine the transmission needs of the class of service, and the specific sessions the class of service will be used for.
2. Create the customized class of service, specifying in order the class of service name, mode name, and transmission priority using:

```
ADD -APPN ConfigCOS <cos name> <transmit priority>
 [SNA defined COS name]
```

If you specify an IBM-defined COS name in the command, you can automatically copy the node row and transmission group row characteristics from the IBM-defined class of service to the class of service you create. For more information on the ConfigCOS parameter syntax, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

3 Configure class of service node rows, specifying the class of service name and the other attributes of the node row, using:

```
ADD -APPN COSNodeRow <cos name> <weight>(0–255) [Congestion=min
 (Yes|No),max (Yes|No)] [Resistance=min,max]
```

For more information on COSNodeRow parameter values, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

4 Configure transmission group rows, specifying the class of service name and other attributes of the transmission group using:

```
ADD -APPN COSTgRow <cos name> <weight>(0-255)
 [ConnectCost=min,max] [ByteCost=min,max] [Security=min,max]
 [PropDelay=min,max] [EffectCap=min,max] [Usd1=min,max]
 [Us2=min,max] [Usd3=min,max]
```

For more information on COSTgRow parameter values, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

5 To define the newly created class of service to the system, use:

```
SET -APPN COSDef = <cos name>
```

After this command is entered, the new class of service will be used by the system.

Mapping Class of Service Names to Mode Names

To map a class of service name to one or more mode names, use:

```
ADD -APPN ModetoCosMap <cos_name> <mode_name> [mode_name ...]
```

Use this command to map any mode names to a customized COS name you have created. An incoming session with the specific mode name will be able to map the mode name to the customized COS. You can also map mode names to default SNA classes of service.

Displaying Class of Service Information

To display a list of available classes of service, enter:

SHow -APPN COS

To display a list of all class of service node rows, including IBM default node tables, enter:

SHow -APPN COSNodeChar

To display a list of all class of service transmission group rows, including IBM default transmission group (TG) tables, enter:

SHow -APPN COSTgChar

To display a list of available modes, enter:

SHow -APPN Mode

To display a list of mode names that are mapped to class of service names, enter:

SHow -APPN ModetoCosMap

For more information about these parameters, refer to Chapter 5 in *Reference for NETBuilder Family Software*.

To display a cached tree for a class of service showing the route to a destination node, including the weight of intermediate nodes, enter:

```
SHow -APPN TreeCache
```

The display shows all classes of service in the cache. Optionally, you can specify a class of service with this command.

Deleting Class of Service Information

To delete a customized class of service table, use

```
DELete -APPN ConfigCOS <cos name>
```

To delete a class of service node row, enter the DELete -APPN COSNodeRow command and specify the class of service name and row number to be deleted. For example, to delete node row 8 in the customized class of service "SanJose," enter:

```
DELete -APPN COSNodeRow SanJose 8
```

To delete a class of service transmission group row, enter the DELete -APPN COSTgRow command, and specify the class of service name and row number to be deleted. For example, to delete transmission group row 8 in the customized class of service "SanJose," enter:

```
DELete -APPN COSTgRow SanJose 8
```

How Class of Service Calculates Routes

The APPN class of service database determines the best routing path by comparing the various factors that make some paths more desirable than others. Among the factors considered are the congestion of nodes along the path, the resistance (desirability of routing) for the nodes along the path, and the characteristics of the transmission groups (such as byte cost and connection cost) *between* the nodes along the path.

Figure 12-1 is an example of an APPN network. In this example, the class of service tables are used to calculate the best path between the network node in San Jose and the network node in New York. This example shows a simple scenario with a single transmission group between each node. However, two TGs (also known as parallel TGs) are supported between network nodes.

There are several possible paths. For this example, the paths are designated as follows:

Path A: San Jose→Seattle→Chicago→Philadelphia→New York

Path B: San Jose→Seattle→Chicago→Washington D.C.→New York

Path C: San Jose→Denver→Chicago→Philadelphia→New York

Path D: San Jose→Denver→Chicago→Washington D.C.→New York

Path E: San Jose→Los Angeles→Dallas→Atlanta→Washington D.C.→New York

All nodes in the example are APPN network nodes.

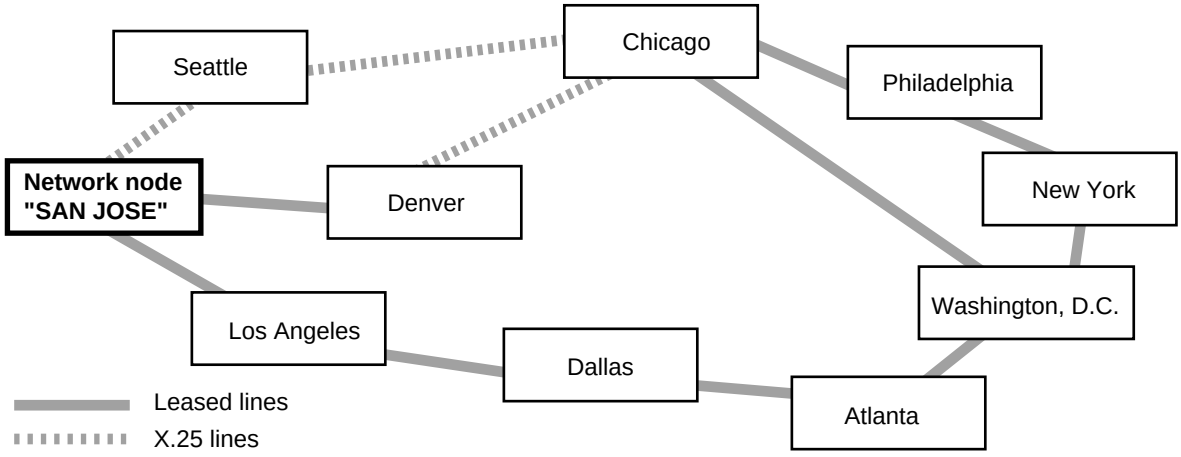


Figure 12-1 COS Example (Network Topology)

Step 1: Determining Node Weights Along a Path

The first step in how routing and topology services determine the best path is by adding the weight of the nodes along the path. The weight of the individual nodes is determined by calculating such factors as node congestion and the resistance (desirability of routing) for each node.

Figure 12-2 shows the node characteristics of the nodes in the network. The figure shows the congestion and resistance values set for each node. The weight shown for each node is calculated by adding the relative factors of congestion and resistance. The lower the resistance, the more desirable the node is to route traffic through. For example, a resistance value of 0 indicates the node is highly desirable to route traffic through, a value of 128 indicates the median, meaning the node is neither highly desirable nor highly undesirable. A resistance value of 255 indicates the node is not desirable to route through.

The resistance plus the congestion value indicates the relative weight of the node. The lower the node weight, the more desirable the node is to route traffic through.

For example, the node in Seattle is uncongested while it has a resistance of 0, indicating it is a desirable node to route traffic through. In contrast, the node in Los Angeles is congested and has a rate of 200, indicating the node is less desirable for routing traffic through.

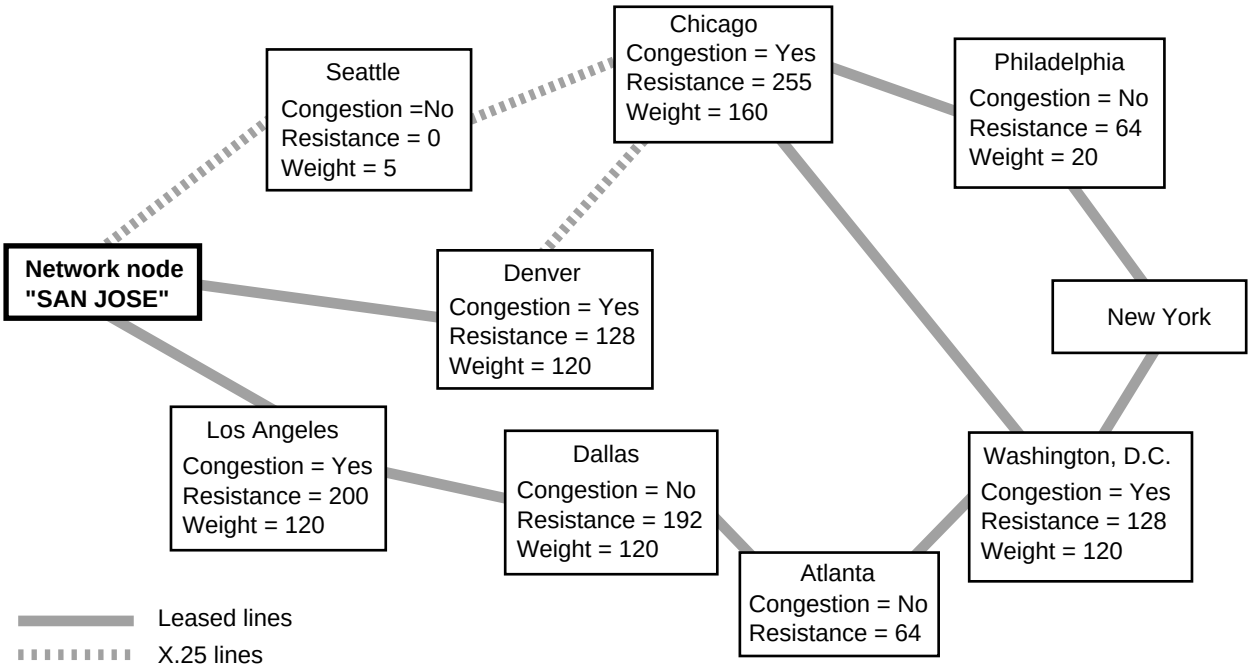


Figure 12-2 COS Example (Calculating Node Weights)

The weight for a given path is calculated by determining the requirements of each path. The requirements are then measured against the class of service node table. The weight of the first node row that meets the requirement of the node is assigned to that node. To check the default node table for the IBM-defined class of service named #CONNECT, see Table 12-8 on page 12-11.

To calculate the weight of a node, the resistance and congestion levels of that node are checked. The node table is then checked to determine the first node row in the table that would accept the requirements of that node; the weight assigned to the node is the weight of that node row. The lower the node row, the lower the weight assigned to the row; the lower the weight, the greater precedence that row has.

For example, the node in Denver has a resistance of 128 and is congested. A network node is congested if it has reached 90 percent of the maximum number of ISR sessions configured for that node. In the node row table, a node is considered either congested ("yes") or uncongested ("no").

When the node row table is used, each row is checked to find the first row that will accept the conditions. The process is as follows:

1. Node row 1 is checked. The conditions are not satisfied because the maximum resistance allowed is 31.
2. Node row 2 is checked. The conditions are not satisfied because the maximum resistance allowed is 63.
3. Node rows 3 and 4 are checked and are also rejected because the maximum resistance values allowed are still lower than Denver's resistance value of 128.
4. Node row 5 is checked, and because the maximum resistance allowed is 159, this is the first row that will accept all the conditions. Because the weight of row 5 is 60, that is the weight assigned to the Denver node.

In this example, if the Denver node were congested, the first node row that would satisfy all conditions would be row 7, which would then assign a weight of 120, changing the total weight of the path.

Using this formula, the appropriate weights of each node are calculated. Table 12-2 lists the correct weights for each node in the figure based on this class of service mode table. (If a different class of service mode is used, a different node table is used, which changes the various calculations.)

Table 12-2 Node Weights Based on Node Row Formula (Example)

Node	Weight Based on COS Node Row (for IBM-default COS #CONNECT)
Seattle	5
Denver	60
Los Angeles	120
Chicago	160
Dallas	120
Atlanta	20
Philadelphia	20
Washington D.C.	120

After the weight of each node is determined, then the weights of all nodes on a path are added together; this determines the total node weight of a given path. Based on the weight calculations in Table 12-2, the total node weight of each path is shown in Table 12-3.

Table 12-3 Total Node Weight for Each Path (Example)

Path	Total Node Weight
PATH A	185
PATH B	285
PATH C	240
PATH D	340
PATH E	380

The table indicates that of the four paths, path A has the lowest weight, which does not mean that path A is the best path. Calculating the weight of the nodes along a path is only the first step. The weights of the transmission groups for each path are then calculated. Proceed to the next section.

Step 2: Determining TG Weights Along a Path

The second factor determining the weight of a path is the weight of all the TGs along the path. The TG consists of the path between two adjacent network nodes. The number of TGs on a path is determined by the number of network nodes on the path; the more nodes on the path, the more TGs there are on the path. For example, on Path A, there are four TGs from the San Jose node to the New York node. On Path E, there are five TGs because Path E includes an additional node. Figure 12-3 shows the different transmission groups.

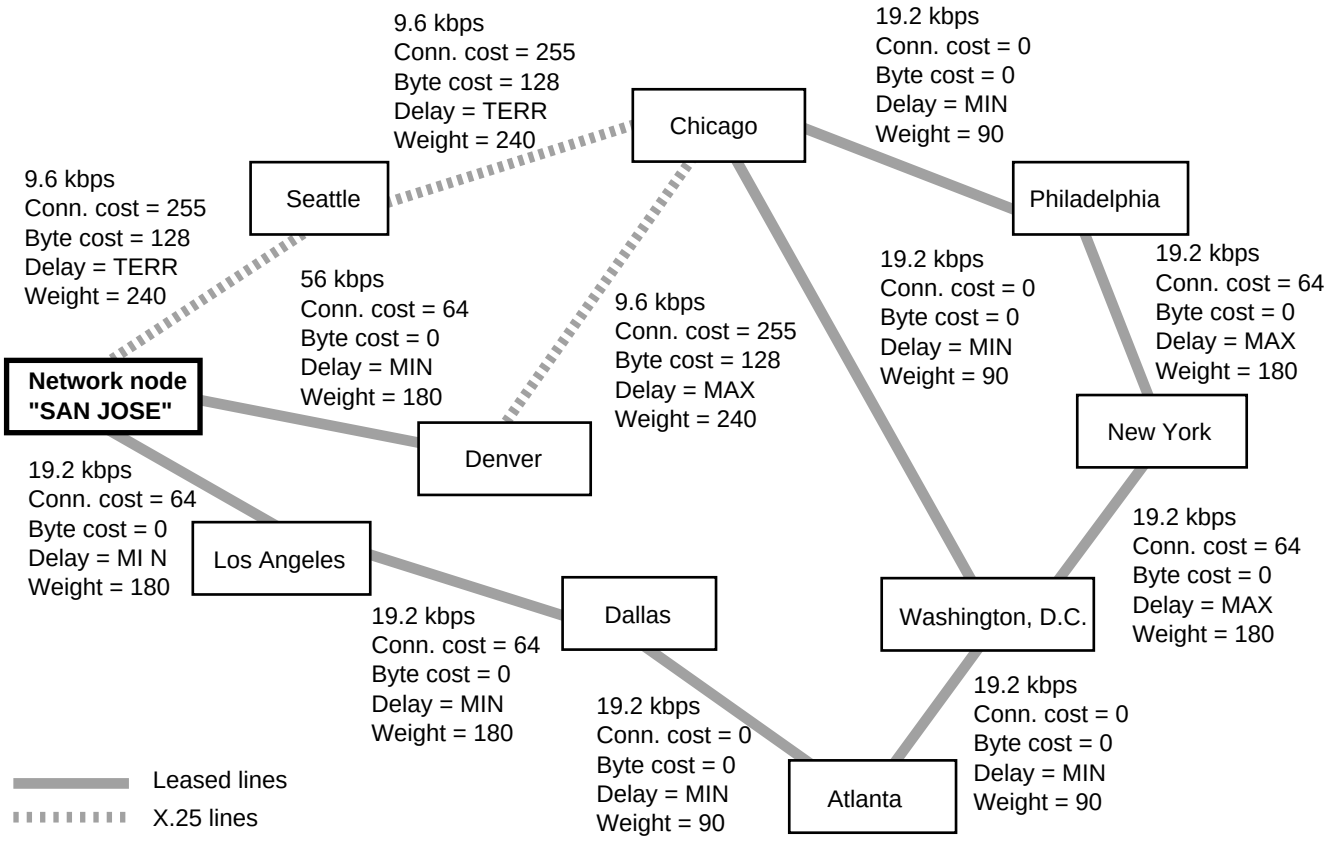


Figure 12-3 COS Example (Configuring TG Weights)

Table 12-4 lists the same information contained in the figure. It lists the attributes of each TG used in calculating the weight of each TG. These attributes are for the example only. No user-defined parameters are used in the example.

Table 12-4 TG Attributes Example

Transmission Group (Link Between Two Network Nodes)	Conn. Cost	Byte Cost	Prop. Delay	Encode Capacity	Security
San Jose→Seattle	255	128	TERR	9,600	MINIMAL
San Jose→Denver	64	0	MIN	56,000	MINIMAL
San Jose→Los Angeles	64	0	MIN	19,200	MINIMAL
Seattle→Chicago	255	128	TERR	9,600	MINIMAL
Denver→Chicago	255	128	MAX	9,600	MINIMAL
Los Angeles→Dallas	64	0	MIN	19,200	MINIMAL
Chicago→Philadelphia	0	0	MIN	19,200	MINIMAL
Chicago→Washington D.C.	0	0	MIN	19,200	MINIMAL
Dallas→Atlanta	0	0	MIN	19,200	MINIMAL
Atlanta→Washington D.C.	0	0	MIN	19,200	MINIMAL
Philadelphia→New York	64	0	MAX	19,200	MINIMAL
Washington D.C.→New York	64	0	MAX	19,200	MINIMAL

To determine the weight of each TG, the class of service TG table is checked. The first row in the TG table that meets the requirements of that TG is used to calculate the weight of the TG.

For example, the TG between San Jose and Seattle has a connection cost of 255 and a byte cost of 128. It has an encoding capacity of 9,600. Table 12-8 on page 12-11 shows the default TG values for the default class of service "#CONNECT."

When the TG row table is used, each row is checked to find the first row that will accept the conditions. The process is as follows:

1 TG row 1 is checked. The conditions are not satisfied because both the connection cost and the byte cost exceed the maximum in TG row 1. (If only one of the attributes exceeded the maximum, that would have been enough to reject TG row 1.)

2 TG rows 2 through 5 are checked and are rejected because the TG's connection cost and byte cost exceed the maximums in those rows.

3 TG row 6 is checked, and the TG's byte cost of 128 matches the maximum allowed in the TG row. The row does not satisfy all the conditions because the TG's connection cost is 255, and the maximum connection cost allowed in row 6 is 128.

4 TG row 7 is checked and is again rejected because the maximum connection cost allowed is not high enough.

5 TG row 8 is checked, and because it allows a maximum connection cost of 255, TG row 8 is the row assigned to the TG. Because the weight for TG row 8 is 240, this is the weight assigned for the TG between San Jose and Seattle.

Using Table 12-9 on page 12-11 and the checking process, the weight for each TG is calculated. Table 12-5 lists the weights calculated based on this class of service mode. (If a different class of service mode is used, a different TG table is used, which changes the various calculations.)

Table 12-5 TG Weights Based on Default Class of Service TG Table

Transmission Group	Weight Based on COS TG Row (for IBM-default COS #CONNECT)
San Jose→Seattle	240
San Jose→Denver	180
San Jose→Los Angeles	180
Seattle→Chicago	240
Denver→Chicago	240
Los Angeles→Dallas	180
Chicago→Philadelphia	90
Chicago→Washington D.C.	90
Dallas→Atlanta	90
Atlanta→Washington D.C.	90
Philadelphia→New York	180
Washington D.C.→New York	180

After the weights of all the TGs are calculated, the weights of the four paths are calculated by adding the weights of each TG on the path. Table 12-6 lists the total TG weights for the four paths in the example.

Table 12-6 Total TG Weight for Each Path (Example)

Path	Total TG Weight
PATH A	750
PATH B	750
PATH C	690
PATH D	690
PATH E	720

After the total TG weight for each path is calculated, this total TG weight is added to the total node weight to calculate the total weight for each path. Proceed to the next section.

Step 3: Calculating the Total Weight for Each Path

To calculate the total weight of each path, the total node weight is added to the total TG weight. Table 12-7 lists the weight values for the four paths.

Table 12-7 Total Calculated Weight for Each Path (Example)

Path	Total Node Weight +	Total TG Weight =	Total Weight for Each Path
PATH A	185	750	935
PATH B	285	750	1035
PATH C	240	690	930
PATH D	340	690	1030
PATH E	380	720	1100

After calculating the total path weight, the class of service determines that the best route from the network node in San Jose to the network node in New York is Path C (San Jose→ Denver→Chicago→Philadelphia→New York) because Path C has the lowest weight.

Dynamic network conditions can affect the weight of a node. For example, if a node that is congested becomes uncongested, then the weight of the node will be lower. The changed node weight will affect the calculation of total path weights and change which is the best route. The APPN class of service calculates the best route at the time of the session request.

Default Class of Service Tables

This section lists the default SNA class of service tables that are used for calculating routes. In all tables, the user-defined values are not shown; the minimum user-defined value is 0 and the maximum is 255.

Default Node Table

Table 12-8 lists the default node table that applies to the different modes. The same node table is used regardless of the mode; it is the mode that determines the transmission priority that differentiates the calculation for node tables. See Table 12-1 on page 12-2 for a list of the default modes and corresponding class of service names.

Table 12-8 Node Table for Default Classes of Service

Row Number	Weight		Congestion	Node Resistance
1	5	Min.	No	0
		Max.	No	31
2	10	Min.	No	0
		Max.	No	63
3	20	Min.	No	0
		Max.	No	95
4	40	Min.	No	0
		Max.	No	127
5	60	Min.	No	0
		Max.	No	159
6	80	Min.	No	0
		Max.	No	191
7	120	Min.	No	0
		Max.	Yes	223
8	160	Min.	No	0
		Max.	Yes	255

Default TG Tables

This section lists the default TG tables for each class of service.

Table 12-9 lists the default TG table for the default class of service #CONNECT. The corresponding mode name is blank (that is, no characters are entered), and the transmission priority is medium.

Table 12-9 #CONNECT Default Class of Service TG Table

Row Number	Weight		Conn. Cost	Byte Cost	Security	Prop. Delay	Encode Capacity
1	30	Min.	0	0	MINIMAL	MIN	0x76
		Max.	0	0	RAD_GUARD	NEGL	MAXIMUM
2	60	Min.	0	0	MINIMAL	MIN	56000
		Max.	0	0	RAD_GUARD	TERR	MAXIMUM
3	90	Min.	0	0	MINIMAL	MIN	19200
		Max.	0	0	RAD_GUARD	TERR	MAXIMUM
4	120	Min.	0	0	MINIMAL	MIN	9600
		Max.	0	0	RAD_GUARD	TERR	MAXIMUM
5	150	Min.	0	0	MINIMAL	MIN	19200
		Max.	0	0	RAD_GUARD	PKT	MAXIMUM
6	180	Min.	0	0	MINIMAL	MIN.	9600
		Max.	128	128	RAD_GUARD	PKT	MAXIMUM
7	210	Min.	0	0	MINIMAL	MIN	4800
		Max.	196	196	RAD_GUARD	MAX	MAXIMUM
8	240	Min.	0	0	MINIMAL	MIN	0x00
		Max.	255	255	RAD_GUARD	MAX	MAXIMUM

Table 12-10 lists the default TG table for the default class of service #BATCH. The corresponding mode name is #BATCH, and the transmission priority is low.

Table 12-10 #BATCH Default Class of Service TG Table

Row Number	Weight		Conn. Cost	Byte Cost	Security	Prop. Delay	Encode Capacity
1	30	Min.	0	0	MINIMAL	MIN	57
		Max.	0	0	RAD_GUARD	NEGL	603979776
2	60	Min.	0	0	MINIMAL	MIN	19
		Max.	0	0	RAD_GUARD	TERR	603979776
3	90	Min.	0	0	MINIMAL	MIN	19
		Max.	128	128	RAD_GUARD	TERR	603979776
4	120	Min.	0	0	MINIMAL	MIN	9
		Max.	0	0	RAD_GUARD	TERR	603979776
5	150	Min.	0	0	MINIMAL	MIN	9
		Max.	128	128	RAD_GUARD	PKT	603979776
6	180	Min.	0	0	MINIMAL	MIN.	9
		Max.	0	0	RAD_GUARD	PKT	603979776
7	210	Min.	0	0	MINIMAL	MIN	4
		Max.	196	196	RAD_GUARD	MAX	603979776
8	240	Min.	0	0	MINIMAL	MIN	0
		Max.	255	255	RAD_GUARD	MAX	603979776

Table 12-11 lists the default TG table for the default class of service #BATCHSC. The corresponding mode name is #BATCHSC, and the transmission priority is low.

Table 12-11 #BATCHSC Default Class of Service TG Table

Row Number	Weight		Conn. Cost	Byte Cost	Security	Prop. Delay	Encode Capacity
1	30	Min.	0	0	PUB_SWITCH	MIN	57
		Max.	0	0	RAD_GUARD	NEGL	603979776
2	60	Min.	0	0	PUB_SWITCH	MIN	19
		Max.	0	0	RAD_GUARD	TERR	603979776
3	90	Min.	0	0	PUB_SWITCH	MIN	19
		Max.	128	128	RAD_GUARD	TERR	603979776
4	120	Min.	0	0	PUB_SWITCH	MIN	9
		Max.	0	0	RAD_GUARD	TERR	603979776
5	150	Min.	0	0	PUB_SWITCH	MIN	9
		Max.	128	128	RAD_GUARD	PKT	603979776
6	180	Min.	0	0	PUB_SWITCH	MIN.	9
		Max.	0	0	RAD_GUARD	PKT	603979776
7	210	Min.	0	0	PUB_SWITCH	MIN	4
		Max.	196	196	RAD_GUARD	MAX	603979776
8	240	Min.	0	0	PUB_SWITCH	MIN	0
		Max.	255	255	RAD_GUARD	MAX	603979776

Table 12-12 lists the default TG table for the default class of service #INTER. The corresponding mode name is #INTER, and the transmission priority is high.

Table 12-12 #INTER Default Class of Service TG Table

Row Number	Weight		Conn. Cost	Byte Cost	Security	Prop. Delay	Encode Capacity
1	30	Min.	0	0	MINIMAL	MIN	4300
		Max.	0	0	RAD_GUARD	NEGL	603979776
2	60	Min.	0	0	MINIMAL	MIN	57
		Max.	0	0	RAD_GUARD	TERR	603979776
3	90	Min.	0	0	MINIMAL	MIN	57
		Max.	128	128	RAD_GUARD	TERR	603979776
4	120	Min.	0	0	MINIMAL	MIN	19
		Max.	0	0	RAD_GUARD	TERR	603979776
5	150	Min.	0	0	MINIMAL	MIN	19
		Max.	128	128	RAD_GUARD	PKT	603979776
6	180	Min.	0	0	MINIMAL	MIN.	9
		Max.	0	0	RAD_GUARD	PKT	603979776
7	210	Min.	0	0	MINIMAL	MIN	9
		Max.	196	196	RAD_GUARD	MAX	603979776
8	240	Min.	0	0	MINIMAL	MIN	0
		Max.	255	255	RAD_GUARD	MAX	603979776

Table 12-13 lists the default TG table for the default class of service #INTERSC. The corresponding mode name is #INTERSC, and the transmission priority is high.

Table 12-13 #INTERSC Default Class of Service TG Table

Row Number	Weight		Conn. Cost	Byte Cost	Security	Prop. Delay	Encode Capacity
1	30	Min.	0	0	PUB_SWITCH	MIN	4300
		Max.	0	0	RAD_GUARD	NEGL	603979776
2	60	Min.	0	0	PUB_SWITCH	MIN	57
		Max.	0	0	RAD_GUARD	TERR	603979776
3	90	Min.	0	0	PUB_SWITCH	MIN	57
		Max.	128	128	RAD_GUARD	TERR	603979776
4	120	Min.	0	0	PUB_SWITCH	MIN	19
		Max.	0	0	RAD_GUARD	TERR	603979776
5	150	Min.	0	0	PUB_SWITCH	MIN	19
		Max.	128	128	RAD_GUARD	PKT	603979776
6	180	Min.	0	0	PUB_SWITCH	MIN.	9
		Max.	0	0	RAD_GUARD	PKT	603979776
7	210	Min.	0	0	PUB_SWITCH	MIN	9
		Max.	196	196	RAD_GUARD	MAX	603979776
8	240	Min.	0	0	PUB_SWITCH	MIN	0
		Max.	255	255	RAD_GUARD	MAX	603979776

Table 12-14 lists the default TG table for the default class of service CPSVCMG. The corresponding mode name is CPSVCMG, and the transmission priority is network.

Table 12-14 CPSVCMG Default Class of Service TG Table

Row Number	Weight		Conn. Cost	Byte Cost	Security	Prop. Delay	Encode Capacity
1	30	Min.	0	0	MINIMAL	MIN	4300
		Max.	0	0	RAD_GUARD	NEGL	603979776
2	60	Min.	0	0	MINIMAL	MIN	57
		Max.	0	0	RAD_GUARD	TERR	603979776
3	90	Min.	0	0	MINIMAL	MIN	9
		Max.	0	0	RAD_GUARD	TERR	603979776
4	120	Min.	0	0	MINIMAL	MIN	9
		Max.	0	0	RAD_GUARD	TERR	603979776
5	150	Min.	0	0	MINIMAL	MIN	19
		Max.	0	0	RAD_GUARD	PKT	603979776
6	180	Min.	0	0	MINIMAL	MIN.	9
		Max.	128	128	RAD_GUARD	MAX	603979776
7	210	Min.	0	0	MINIMAL	MIN	4
		Max.	196	196	RAD_GUARD	MAX	603979776
8	240	Min.	0	0	MINIMAL	MIN	0
		Max.	255	255	RAD_GUARD	MAX	603979776

Table 12-15 lists the default TG table for the default class of service SNASVCMG. The corresponding mode name is either SNASVCMG or CPSVRMG, and the transmission priority in both cases is network.

Table 12-15 SNASVCMG Default Class of Service TG Table

Row Number	Weight		Conn. Cost	Byte Cost	Security	Prop. Delay	Encode Capacity
1	30	Min.	0	0	MINIMAL	MIN	4300
		Max.	0	0	RAD_GUARD	NEGL	603979776
2	60	Min.	0	0	MINIMAL	MIN	57
		Max.	0	0	RAD_GUARD	TERR	603979776
3	90	Min.	0	0	MINIMAL	MIN	19
		Max.	0	0	RAD_GUARD	TERR	603979776
4	120	Min.	0	0	MINIMAL	MIN	9
		Max.	0	0	RAD_GUARD	TERR	603979776
5	150	Min.	0	0	MINIMAL	MIN	19
		Max.	0	0	RAD_GUARD	PKT	603979776
6	180	Min.	0	0	MINIMAL	MIN.	9
		Max.	128	128	RAD_GUARD	PKT	603979776
7	210	Min.	0	0	MINIMAL	MIN	4
		Max.	196	196	RAD_GUARD	MAX	603979776
8	240	Min.	0	0	MINIMAL	MIN	0
		Max.	255	255	RAD_GUARD	MAX	603979776

13 IPX ROUTING

This chapter describes the procedures for configuring your system to perform Internetwork Packet Exchange (IPX) routing. It also describes how the router works and gives guidelines for operating, managing, and troubleshooting it.

For conceptual information, refer to "How the IPX Router Works" on page 13-33.

Setting Up a Basic IPX Router

Use the following procedures to set up your system to route IPX packets. After you complete the procedures in this section, verify that the system is routing packets properly using the procedures in "Verifying the Configuration" on page 13-9.

Configuring for Local Area Networks and Point-to-Point Links

Use this procedure to configure basic IPX routing over LAN ports and Point-to-Point Protocol (PPP) links.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router using the procedure in Chapter 1.

Procedure

To configure the basic IPX router for LANs and PPP links, follow these steps:

1 Configure the network number connected through each router interface using:

```
SETDefault !<port> -IPX NETnumber = &<number>(0-FFFFFFFD)
 [Ethernet | Ieee | Llc | Snap | PPP]
```

Valid network numbers consist of up to eight hexadecimal digits in the range &0 to &FFFFFFFD. The network numbers &FFFFFFFE and &FFFFFFFF are reserved. You do not have to specify leading zeros in this network number.

You may also configure the port as unnumbered PPP. Refer to Chapter 6 for more information.

2 Verify dynamic route learning is enabled using:

```
SHow !<port> -NRIP CONTrol
```

By default, NetWare Routing Information Protocol (NRIP) is set to Auto. If you do not enable NetWare Link Services Protocol (NLSP) as the routing protocol, Auto means both Talk and Listen. If NLSP is enabled, Auto means Talk if there are non-NLSP routers detected. The router is constantly listening.

When NRIP is listening, the router receives Routing Information Protocol (RIP) broadcasts and can maintain the routing table. When NRIP is talking, the router can send RIP broadcasts.

3 Enable IPX routing for each port using:

```
SETDefault !<port> -IPX CONTrol = ROute
```

4 If there are more users to serve than a primary server is licensed to handle and there is a backup server available, specify a preferred backup server using:

```
ADD !<port> -SAP PreferredServer "<server name>", ["<server
 name>"...]
```

After a list of preferred servers is configured, the IPX router responds to "get nearest server" requests with one of the reachable preferred servers regardless of the server location or number of hops. If no preferred server is available, the normal selection process of the nearest server takes place. In this way, the primary server and backup server can alternately serve all the users and lessen the burden on the primary server.

NetWare 4.0 clients and pre-4.0 clients specify different service types in their "get nearest server" requests. Pre-4.0 clients use File Server type (0x0004) while 4.0 clients are looking for Directory Name Server type (0x026B); appropriate preferred servers must be added.

5 Verify the IPX configuration by entering:

```
SHow -IPX CONFiguration
```

The router displays the IPX configuration information. If the -IPX CONTrol parameter is not set to ROute, if the network numbers are incorrect, or if the -NRIP and -SAP CONTrol parameters are not set to Talk and Listen for each port you are configuring, repeat steps 2, 3, and 4. Additional verification steps are provided in "Verifying the Configuration" on page 13-9.

To complete the configuration for PPP links, refer to Chapter 34.

Configuring Secondary Networks with Different Header Formats

For LAN interfaces, IPX allows one physical network to be segmented into different logical networks, or secondary networks, and configured with different header formats. The header formats correspond to different encapsulation methods that allow the IPX protocol to deliver IPX packets. Table 13-1 lists the header formats supported by IPX encapsulation and the values associated with these formats.

Table 13-1 IPX Packet Header Formats

Values	Header Formats Supported under IPX Encapsulation
Ieee	IPX packets are encapsulated in IEEE 802.3 header format (Ethernet and FDDI).
Ethernet	IPX packets are encapsulated in Ethernet V2 header format (Ethernet only).
Snap	IPX packets are encapsulated in SNAP header format.
Llc	IPX packets are encapsulated in IEEE 802.2 header format.

3Com recommends using Ethernet V2 for Ethernet and SNAP for FDDI and token ring.

The number of secondary networks differs between interface types:

- For Ethernet interfaces, four different networks can be configured: Ethernet V2 headers (identified as Ethernet_II on Novell servers); IEEE headers (802.3 raw), identified as Ethernet_802.3 on Novell servers, and the default for Ethernet and Phone Line Gateway (PLG) lines; Logical Link Control (LLC) headers (identified as Ethernet_802.2 on Novell servers); and SubNetwork Access Protocol (SNAP), identified as Ethernet_SNAP on Novell servers.
- For token ring, three different networks can be configured: LLC, SNAP, and IEEE.
- For the Fiber Distributed Data Interface (FDDI), three different networks can be configured: LLC, SNAP, and IEEE.

  For each of the interface types, configure the primary network with the SETDefault command; configure the secondary networks with the ADD command.

Figure 13-1 shows a router with three LAN ports of different types:

- Port 1 (Ethernet) is connected to network 100 with IEEE header format, network 200 with Ethernet header format, network 300 with LLC header format, and network 400 with SNAP header format.
- Port 2 (token ring) is connected to two networks: network 500 with LLC header format and network 600 with SNAP header format.
- Port 3 (FDDI) is connected to two networks: network 700 with SNAP header format and network 800 with LLC header format.

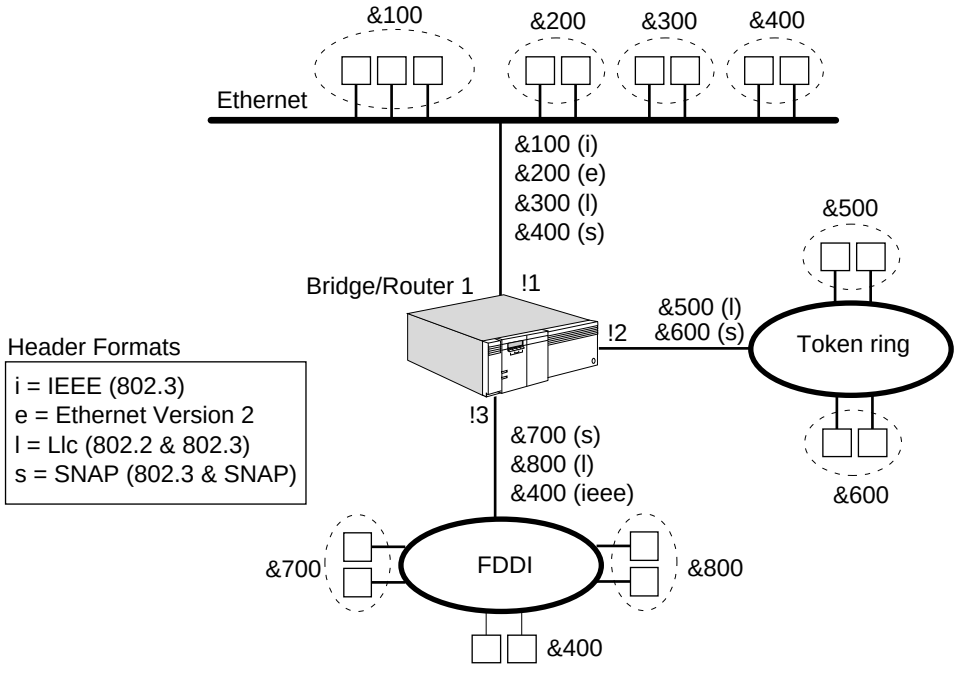


Figure 13-1 Configuring Multiple Networks for Different Header Formats

To configure the primary and secondary network for port 1 shown in Figure 13-1, follow these steps:

1 Configure the primary network for port 1 by entering:

```
SETDefault !1 -IPX NETnumber = 100 Ieee
```

The primary networks for ports 2 and 3 are configured using the SETDefault command, the appropriate port number, and the appropriate header format specifier (LLC for network 500 on port 2 and SNAP for network 700 on port 3).

2 Configure the Ethernet secondary network for port 1 by entering:

```
ADD !1 -IPX NETnumber = 200 Ethernet
```

The remaining secondary networks for port 1 are configured using the ADD command, the port specifier !1, and the appropriate header format specifier (LLC for network 300 and SNAP for network 400).

The remaining secondary networks for ports 2 and 3 are configured using the ADD command, the appropriate port number, and the appropriate header format specifier (SNAP for network 600 on port 2 and LLC for network 800 on port 3).

Configuring for Wide Area Networks

To configure your IPX router to perform routing over Frame Relay, Switched Multimegabit Data Service (SMDS), X.25, or ATM, refer to Chapter 42, Chapter 44, Chapter 45, or Chapter 47. For information on wide area networking using Integrated Services Digital Network (ISDN), refer to Chapter 35.

Routing IPX over Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), X.25, and ATM is supported over fully meshed, partially meshed, and nonmeshed topologies.

If you plan to route IPX over Frame Relay, ATM DXI, X.25, or ATM in a partially meshed or nonmeshed topology, you must be sure that the next-hop split horizon feature is enabled by configuring neighbors. For complete information on configuring IPX routing over Frame Relay, ATM DXI, X.25, or ATM, including a discussion on fully meshed, partially meshed, and nonmeshed topologies and next-hop split horizon, refer to Chapter 42, Chapter 43, Chapter 45, or Chapter 47.

Routing IPX over SMDS is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your IPX router to perform routing over SMDS, refer to Chapter 44. Nonmeshed topology may be used with virtual ports. To configure IPX routing over PPP/PLG, refer to Chapter 34.

For WAN interfaces, you do not need to specify a header format. The formats are as follows:

- PLG uses the Ethernet header format.
- PPP uses the PPP header format.
- X.25 uses the X.25 header format.
- SMDS uses the SMDS header format.
- Frame Relay uses the Frame Relay header format.
- ATM uses the ATM header format.

You can assign secondary networks on WAN interfaces, but the status of those networks will be down.

Configuring IPXWAN over PPP

Novell has published a specification for IPX communications over wide area network services (such as PPP, X.25, Frame Relay) called IPXWAN. The specification outlines how IPX negotiations take place in these environments; for example, Novell IPX uses IPXWAN to exchange necessary router-to-router information before exchanging IPX NRIP, Service Advertising Protocol (SAP), and NLSP information over various WAN links. The 3Com implementation of the IPXWAN Protocol currently supports PPP, Frame Relay, and X.25.

To achieve interoperability between a 3Com bridge/router and a Novell Multi-Protocol Router (MPR) across a WAN link, you must configure IPXWAN over PPP on your bridge/router as shown in Figure 13-2.

If you are using the nonperiodic mode of NRIP and SAP, both sides of the WAN link must be configured the same way.

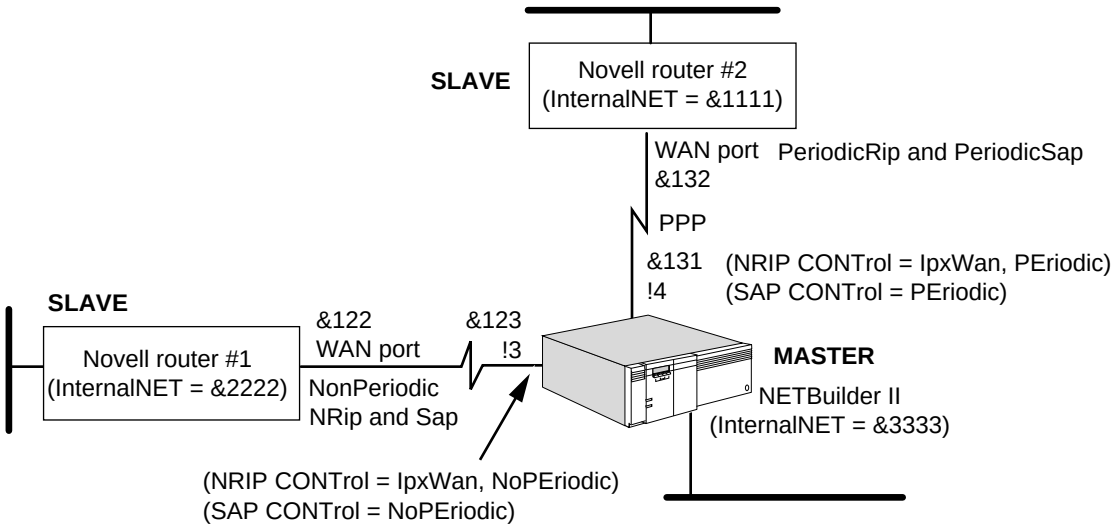


Figure 13-2 IPXWAN over PPP Using NRIP and SAP

Prerequisites

Before beginning this procedure, perform the following steps:

- Configure IPX routing on the LAN ports as described in "Configuring for Local Area Networks and Point-to-Point Links" on page 13-1.
- Configure PPP over the port as described in Chapter 34.

Procedure

To configure IPXWAN over PPP, follow these steps:

1 Configure the network numbers on the wide area interfaces that will be running IPXWAN using:

```
SETDefault !<port> -IPX NETnumber = &<number>(0-FFFFFFFD)
  [Ethernet|Ieee|Llc|Snap|X25|PPP|Frame]
```

Valid network numbers consist of up to eight hexadecimal digits in the range &0 to &FFFFFFFD. The network number &FFFFFFFE and &FFFFFFFF are reserved. You do not have to specify leading zeros in this network number.

2 Assign an internal network number to each router.

To assign the internal network number, use:

```
SETDefault -IPX InternalNET = &<number>(0-FFFFFFFD)
```

The InternalNET number must be unique throughout the IPX Internet. Valid network numbers consist of up to eight hexadecimal digits in the range &0 to &FFFFFFFD. The network number &FFFFFFFE and &FFFFFFFF are reserved.

The routers use the internal network number during IPXWAN negotiation to determine which router is the master and which router is the slave. The router with the lowest internal network number becomes the slave during link establishment and information exchange.

As shown in Figure 13-2, the 3Com bridge/router has the highest internal network number and is designated as the master over both Novell router #1 and #2. When packets are routed between the 3Com bridge/router and the Novell routers, the network number of the bridge/router is used. Consequently, the network numbers on port 3 and port 4 of the Novell routers do not need to be assigned.

If you assign network numbers to port 3 and port 4 of the Novell routers, the 3Com bridge/router negotiates the network numbers, and the network number of the master is used. For example in Figure 13-2, on port 3 of the 3Com bridge/router, network number &123 is used; on port 4 of the 3Com bridge/router, network number &131 is used during packet transmission.

3 For network management purposes, assign a symbolic name to each router.

The router uses this name during IPXWAN negotiation to build NRIP/SAP Information Request/Response packets. The router name must be unique throughout the IPX Internet and can be up to 48 characters in length.

To assign a symbolic name, use:

```
SETDefault -IPX RouterName = "<string>"
```

Because the IPX router does not provide a service, the router name is not advertised in SAP updates, which substantially reduces the network traffic in a large network configuration.

4 Determine whether to use periodic or nonperiodic (incremental) NRIP/SAP update modes on your LAN or WAN ports.

All participating routers and servers must use the same update mode to avoid stale NRIP and SAP entries and loss of network connectivity.

When used in a stable and reliable network, nonperiodic updates can eliminate the constant and expensive network traffic of IPX NRIP and SAP updates on all media, except at initialization time. After initialization, updates also are sent incrementally when changes occur.

For a LAN, use periodic updates. If two bridge/routers are connected over a WAN, use nonperiodic updates. Use periodic updates on the WAN only when mixing 3Com and non-3Com routers on the same WAN link.

To enable nonperiodic updates, use:

```
SETDefault !<port> -NRIP CONTrol = NoPEriodic
SETDefault !<port> -SAP CONTrol = NoPEriodic
```

As shown in Figure 13-2, Novell router #1 (port 3) and the bridge/router (port 3) are configured for nonperiodic NRIP and SAP updates. The IPX router sends out

NRIP and SAP updates immediately after a LAN or WAN path comes up, which completes NRIP and SAP updates more quickly.

Set the -NRIP and -SAP CONTrol parameter to PEriodic on networks in which frequent topology changes occur.

5 Enable the IPXWAN protocol on the specified port of each 3Com router using:

```
SETDefault !<port> -IPX CONTrol = IpxWan
```

Configuring for NLSP

The NLSP provides a hierarchical structure for large IPX routing environments. NLSP uses a link-state routing algorithm that provides faster network convergence with reduced network resource overhead (bandwidth and CPU cycles) than other routing algorithms, for example, NRIP and SAP, which use a distance vector algorithm.

NLSP runs over all networking media, including LANs (Ethernet, token ring, and FDDI), and WAN/MAN (X.25, Frame Relay, ATM, SMDS, and PPP links).

Prerequisites

Before beginning this procedure, perform the following steps:

- Configure an internal network number (-IPX InternalNET parameter) to the router.
- Configure IPX network numbers on all the LAN and WAN ports.
- Enable IPX routing on those ports.
- If there are multiple logical networks on a port, make sure the primary network is configured the same for all routers on the LAN. NLSP routes communicate with each other using the primary network only.

Procedure

To configure NLSP, follow these steps:

1 Determine and assign the area address for the router using:

```
ADD -NLSP AreaAddress <net> <mask>
```

NLSP uses a portion of the 32-bit IPX network number to identify an area. The AreaAddress parameter is used to describe the value and length of the area number. The area address is a pair of 32-bit integers expressed in hexadecimal format. The first set of numbers identified as <net> describes the value of the area number, while the second set identified as <mask> determines the length of the area address, or number of bits in the IPX network number field that are used to identify the area.

The mask is a number of leading 1 bits, followed by 0 bits. The leading 1 bits must be contiguous. Similar to the concept of IP subnet masks, the number of leading 1 bits in the mask determines the number of leading bits in the <net> field, which is considered to be the area number instead of the network number. Any bit position identified by a 0 in the mask is considered to be the network number. The following example shows the syntax of the area and mask:

```
ADD -NLSP AreaAddress 12345600 FFFFFF00
```

The mask of FFFFFF00 indicates that the first 6 characters (24 bits) in the <net> field are considered to be the area number; the last two characters (8 bits) are

used to identify a network within that area. The network number is defined using the -IPX NETnumber parameter.

All network numbers assigned to routers within an area must fall within a configured area prefix. In this example, any router within the area identified as AreaAddress 12345600 FFFFFF00 must be assigned network numbers beginning with the prefix 123456XX. The valid range for network numbers within this area is 12345600–123456FF.

An area address must meet the following requirements:

- A mask is required, identifying a range of networks residing within the area.
- For example, all network numbers in the range 12345600 to 123456FF reside within the area 12345600 to FFFFFF00. It is not necessary that all of the area network numbers are addressed and operational.
- All network numbers within the area must fall within the address range.
- With an area address of 12345600 FFFFFF00, all IPX networks must begin with 123456XX. The area address 00000000 00000000 is the default and this area address includes all IPX network numbers.

2 Determine which interfaces to enable for NLSP.

NLSP routing should be enabled on all ports, including ports that have no NLSP routers connected to them. When NLSP is enabled on a port, and if there are other NRIP and SAP routers on the same port, NLSP automatically imports the NRIP and SAP information into the NLSP domain. NLSP automatically exports NLSP learned information to NRIP and SAP routers. The importing and exporting of information allows smooth operation between NLSP and non-NLSP routers.

If NLSP is disabled on the port, the import and export of NRIP and SAP routing information does not occur and causes network segmentation.

If you set the -NRIP and -SAP CONTrol parameters to Auto, the NLSP router determines if NRIP and SAP need to be enabled on the port. When the router detects a non-NLSP router or file server, it enables both NRIP and SAP to communicate with them; otherwise, it disables NRIP and SAP to conserve bandwidth.

If NLSP is disabled, the Auto setting for -NRIP and -SAP CONTrol means that both protocols are talking and listening. NRIP and SAP updates are continuously sent out. To disable these updates, use the NoTalk and/or NoListen values to override the Auto value.

By disabling NRIP and SAP, you conserve network bandwidth which is useful over Frame Relay or PPP lines. If Auto is selected for NRIP and SAP and all routers on the network support NLSP, and NLSP is enabled, the RIP/SAP traffic will automatically disappear from the network (except where files servers are present).

Enable the NLSP protocol on the specified port of each 3Com router using:

```
SETDefault !<port> -NLSP CONTrol = Enable
```

3 Display the configuration information for all ports by entering:

```
SHow -NLSP CONFiguration
```

4 Display the NLSP adjacencies by entering:

```
SHow -NLSP ADJacencies
```

Verifying the Configuration

This section explains how to verify the status of networks that are reachable from the router and how to get statistics from the router and from other networks and stations.

Before you use the router for interconnecting networks, verify the following items on your network:

1. Check the state of the current configuration by entering:

   SHow -IPX DIAGnostics

   The diagnostics command will display any configuration errors that have occurred.

2. Check the state of the NRIP and SAP Services by entering:

   SHow -NRIP CONTrol
   SHow -SAP CONTrol

   The control parameter for both of these services should be set to Talk and Listen to enable dynamic route learning.

3. Check the state of all networks assigned to the ports of a router by entering:

   SHow -IPX NETnumber
   SHow -IPX CONTrol

   The first command displays the network numbers assigned to each port on this router and the state that each network is in. Each network should be in the UP state. If a network is in the DOWN state, check that the -IPX CONTrol parameter is enabled. If the network is in the DISABLE state, make sure that all PORT and PATH parameters are configured appropriately. The second command allows you to verify if routing is enabled on the ports.

4. Verify that the router can access the networks it was configured to access by entering:

   SHow -IPX AllRoutes Long

   This command displays all known routes (dynamic, static, and default, if configured), hop counts, and cost in the IPX Routing Table. Adding "Long" to the command also displays gateway information.

5. Verify that the router can learn and exchange service information from servers on the directly connected networks and other routers, by entering:

   SHow -IPX AllServers Long

   The router displays a server table. For more information on the contents of the server table, refer to "Learning Routes and Service Information" on page 13-37. Adding "Long" to the command also displays gateway information.

6. Display the configuration information for all paths by entering:

   SHow -PATH CONFiguration

   Check that the configuration information is correct for all paths.

7. Display the configuration information for all ports by entering:

   SHow -PORT CONFiguration

   Check that the configuration information is correct for all ports.

8. Verify the setting of the -PORT ProtMacAddrFmt parameter using:

   SHow !<port> -PORT ProtMacAddrFmt

If you did not configure the ProtMacAddrFmt parameter, the software automatically selects DefaultIPX, and based on the port media type, automatically selects either DefaultIPX(NC) for the SuperStack II NETBuilder bridge/router LAN and high-speed serial (HSS) ports or DefaultIPX(C) for Ethernet, FDDI, and HSS ports.

For more information about this parameter, refer to Chapter 43 in *Reference for NETBuilder Family Software*.

9 Display the current IPX configuration parameters by entering:

```
SHow -IPX CONFiguration
```

Check that the configuration information is correct.

10 Determine connectivity to an IPX node on the network using:

```
NetwarePING &<network>%<host> [timeout (1–300 seconds)]
```

11 Display the NLSP, NRIP, and SAP Services and verify the configuration information by entering:

```
SHow -NLSP CONFiguration
SHow -NRIP CONFiguration
SHow -SAP CONFiguration
```

12 Make a connection from a workstation on one attached network to a file server on another network to see if packets can be routed across the router.

13 Obtain configuration information from a NetWare server using:

```
NetwareView &<network>%<host> [timeout (1–300 seconds)]
```

14 Obtain the status of the router by entering:

```
SHow -IPX DIAGnostics
```

Getting Statistics

To view statistics, enter:

```
SHow -SYS STATistics -IPX
SHow -SYS STATistics -NRIP
SHow -SYS STATistics -SAP
SHow -SYS STATistics -NLSP
```

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information on these parameters, refer to Chapter 58 in *Reference for NETBuilder Family Software*. For information on interpreting the statistics displays, refer to Appendix H.

Troubleshooting the Configuration

If you are unable to make connections to other networks after setting up the router, review the following troubleshooting procedure. This procedure can help correct problems in making single-hop (involving one router) and multiple-hop (involving more than one router) connections. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier for assistance.

To troubleshoot the configuration, follow these steps:

1 If you are experiencing problems because of configuration errors, examine the service diagnostics information using:

```
SHow !<port> -IPX DIAGnostics
```

The diagnostics command displays troubleshooting information about IPX routing and gives suggestions for corrective actions. The troubleshooting information consists of global diagnostic messages, port specific diagnostic messages, NRIP diagnostic messages, and SAP diagnostic messages.

The following display appears:

```
--------------------IPX Diagnostic Information-----------------
No global diagnostic information available.
-------------------------Port 1-----------------------------
This port seems to be normal.
-------------------------Port 2-----------------------------
Network &00000300 conflicts with &DDDDD200 on node 080002A078DB.
-------------------------Port 3-----------------------------
IPX Routing is not enabled.Please configure IPX CONTrol parameter.
-------------------------Port 4-----------------------------
IPX Routing is not enabled.Please configure IPX CONTrol parameter.
```

In this example, the network assigned to port 2 is shown as &00000300, but node 2 at 080002A078DB thinks that the network should be &DDDDD200.

2 Check that all cables are properly connected and that the router is properly installed.

For installation instructions, refer to the installation guide provided with your bridge/router.

3 Verify that routing is enabled by entering:

```
SHow -IPX CONTrol
```

The router displays the current values for the CONTrol parameter for each port. If the values are set to ROute, no action is necessary. If the values are set to NoROute, to enable the IPX router use:

```
SETDefault !<port> -IPX CONTrol = ROute
```

4 Check the network number and status by entering:

```
SHow -IPX NETnumber
```

Look at the status of the networks: each configured network should be in the UP state. If it is in the DOWN state, check to make sure that all PORT and PATH parameters are configured correctly. If the port is in the DISABLED state, make sure IPX routing is enabled for the port.

Look at the current network configuration: if no network is configured on the specific port, to add a network number to that port use:

```
SETDefault !<port> -IPX NETnumber = &<number>(0-FFFFFFFD)
  [Ethernet | Ieee | Llc | Snap | X25 | PPP | Frame | SMDS | ATM]
```

Make sure that you assign the network number to the correct port. Network numbers consist of eight hexadecimal digits. For example, to assign network number 4321 to port 2 on the router, enter:

```
SETDefault !2 -IPX NETnumber = &4321
```

If this is an Ethernet port, all IPX packets sent from this port will be encapsulated with the IEEE header format, because IEEE is the default format and no format is specified in the command. Make sure that the header type configured matches that of the NetWare servers and clients.

The software detects the media type and sets the header format correctly. You do not need to specify the header format type in the NETnumber parameter for a WAN. Each LAN must be configured if the default is not appropriate.

To detect a mismatch of encapsulation type or network, enter:

```
SHow -IPX DIAGnostics
```

5 Verify that dynamic learning and NRIP updates are enabled on the port by entering:

```
SHow -NRIP CONTrol
```

The router displays the current values for the NRIP CONTrol parameter. If dynamic learning and NRIP updates are disabled on the port, enable it using:

```
SETDefault !<port> -NRIP CONTrol = (Talk, Listen, PEriodic)
SETDefault !<port> -SAP CONTROL = (Talk, Listen, PEriodic)
```

6 Verify that the network you are trying to reach is in the IPX Routing Table by entering:

```
SHow -IPX AllRoutes
```

The IPX router displays the routing table entries. From the table entry, you can determine which path is being used. Examine the entries to make sure a route in the table is taking the appropriate path. You can also specify a network number using the SHow -IPX AllRoutes <NETnumber> syntax to verify single route reachability.

If the entry in the table has a hop number of 16, the network is unreachable at the present time. Wait several minutes and use the SHow -IPX AllRoutes <NETnumber> syntax again. Optionally, you can use the FLush -IPX AllRoutes command to remove dynamically learned routes and services. After flushing the table, wait a few minutes before entering the SHow -IPX AllRoutes command again.

7 Verify that the server you are trying to reach is in the IPX Server Table by entering:

```
SHow -IPX AllServers
```

The IPX router displays all known servers in the IPX Server Table, including server addresses, server names, and the number of hops involved. Make sure the server name to which you are trying to connect is in the table.

You can also specify a server name using the SHow -IPX AllServers "<string>" syntax to verify single server reachability.

8 If you are experiencing connectivity problems due to routing and service tables that are not synchronized between IPX routers on your internetwork, flush the routing and service table entries by entering:

```
FLush -IPX AllRoutes
FLush -IPX AllServers
```

These commands remove all dynamically learned entries from the routing table and all entries from the server table, and then rebuild these tables.

9 Display statistics for the IPX Service by entering:

```
SHow -SYS STATistics -IPX
```

For information on interpreting statistics displays, refer to Appendix H.

10 Display statistics for the NRIP and SAP Services by entering:

```
SHow -SYS STATistics -NRIP
SHow -SYS STATistics -SAP
```

Customizing the IPX Router

This section provides additional procedures you can use to configure your IPX router. For details on IPX parameters, refer to Chapter 31, Chapter 38, Chapter 39, and Chapter 49 in *Reference for NETBuilder Family Software*.

Controlling NRIP and SAP Advertisements

The -NRIP and -SAP CONTrol parameters determine how the router sends the routing table information to the network. For NRIP and SAP parameter information, refer to Chapter 39 and Chapter 49 in *Reference for NETBuilder Family Software*. For information on periodic and nonperiodic NRIP and SAP updates, refer to "Controlling NRIP and SAP Updates" on page 13-14.

You only need to specify values that differ from the default values.

Enabling and Disabling Dynamic Learning and NRIP Updates

The router maintains a routing table of all networks it can reach. The router adds routes to the routing table automatically from its neighbors' route advertisements unless you disable dynamic learning. If you do not disable dynamic learning, the router eventually will learn all of the networks it can reach.

Enable dynamic learning for a given port using:

```
SETDefault !<port> -NRIP CONTrol = Listen
```

For ports that connect non-broadcast multiaccess (NBMA) networks, you can enable dynamic neighbor learning using:

```
SETDefault !<port> -NRIP CONTrol = DynamicNbr
```

You may want to disable dynamic learning if you are configuring static routing on a port and want to eliminate traffic associated with the route advertisements. Disabling dynamic learning frees bandwidth on slow serial data links and is especially cost-effective on an X.25 or Frame Relay interface where packet charges are enforced. Disable dynamic learning using:

```
SETDefault !<port> -NRIP CONTrol = (NoListen, NoTalk)
```

For ports that connect NBMA networks, you can disable dynamic neighbor learning using:

```
SETDefault !<port> -NRIP CONTrol = NoDynamicNbr
```

The effect of setting the Listen | NoListen value for the -NRIP CONTrol parameter depends on the setting of the ROute | NoROute value for the -IPX CONTrol parameter. If the -IPX CONTrol parameter is set to NoROute, dynamic learning is disabled and the NRIP update is also disabled.

If you disable dynamic learning, you must add a static route for each network to which you want to connect. For more information, refer to "Adding and Deleting Static Routes" on page 13-17.

For a description of additional -NRIP CONTrol parameter values, refer to Chapter 39 in *Reference for NETBuilder Family Software*. For a discussion of split horizon, refer to "Solving the Slow Convergence Problem with Poison Reverse" on page 13-41.

Enabling Triggered NRIP Updates

Setting the -NRIP CONTrol parameter to Trigger causes the router to send an update packet when the network topology reflected in its routing table changes. The advantage is that the network immediately knows a potentially better route to a particular network. Setting the -NRIP CONTrol parameter to NoTrigger reduces the amount of data packets broadcast over the network during topology changes, and normal update packets will be sent only at the time interval specified by the UpdateTime parameter.

Enable the trigger feature for a given port using:

```
SETDefault !<port> -NRIP CONTrol = Trigger
```

Using Poison Reverse or No Poison Reverse

The poison reverse and no poison reverse implementations are described in detail in "Solving the Slow Convergence Problem with Poison Reverse" on page 13-41.

To enable the poison reverse feature for a given port, use:

```
SETDefault !<port> -NRIP CONTrol = POison
```

To disable the poison reverse feature for a given port, use:

```
SETDefault !<port> -NRIP CONTrol = NoPOison
```

Another way to solve the slow convergence problem is to run NLSP instead of NRIP and SAP.

Controlling NRIP and SAP Updates

In a stable and reliable network in which topology changes are infrequent, you can eliminate most of the traffic of NRIP and SAP updates by using the NoPEriodic values of the -NRIP and -SAP CONTrol parameters. You can select these values using:

```
SETDefault !<port> -NRIP CONTrol = NoPEriodic
SETDefault !<port> -SAP CONTrol = NoPEriodic
```

You can always use NoPEriodic on WANs. NoPEriodic can only be used on LANs if the servers also support NoPEriodic. The default setting for a WAN is NoPEriodic, the default for a LAN is PEriodic. Because multiprotocol ATM is treated like a LAN, it also uses the PEriodic setting for its default. Do not change the setting to NoPEriodic on a LAN unless NoPEriodic is supported by the servers.

When you select these values, the IPX router shuts off periodic NRIP and SAP updates and switches to incremental updates, allowing the transmission of updates only when topology changes occur. When selecting these options, make sure that all participating routers use the same option. You can use the PEriodic and NoPEriodic settings for NRIP and SAP for all media.

If you are using the Boundary Routing system architecture, use smart filters and NoPEriodic on the WAN links to the remote sites.

If your network has frequent topology changes, NRIP and SAP updates need to occur on a periodic basis. However, setting NRIP and SAP updates to a periodic basis should only be used on the WAN when mixing 3Com and non-3Com routers on the same link. Selecting periodic updates in an all-3Com network can create severe traffic problems. If you have a mixed network on the WAN link, you can enable the periodic updates using:

```
SETDefault !<port> -NRIP CONTrol = PEriodic
SETDefault !<port> -SAP CONTrol = PEriodic
```

When you select these values, the IPX router sends NRIP and SAP updates when topology changes occur (triggered updates) and each time the value of UpdateTime parameter expires.

You can use the Auto option for NRIP and SAP to allow the router to transition from the RIP and SAP routing protocols to NLSP as NLSP routers are configured on the network. If there is a mix of RIP, SAP, and NLSP routers on a LAN, and the NLSP routers have selected Auto, the NLSP routers will interoperate with the NRIP and SAP routers by announcing NLSP learned routes and services to the non-NLSP routers using RIP and SAP updates.

Learning of routes and services continue in the same way. When the last router on a network is configured with NLSP, all RIP and SAP traffic automatically disappears from the network unless there are file servers present. The Auto option can be overridden in the NRIP and SAP Services if you require some flexibility in the control of your network.

For conceptual information, refer to "Learning Routes and Service Information" on page 13-37. To eliminate NRIP and SAP updates in a Boundary Routing environment, you can use the smart filtering feature; for more information, refer to Chapter 32. Another way to solve this problem is to run NLSP instead of NRIP and SAP.

Controlling Route and Service Aging

The UpdateTime parameter controls learned route aging when dynamic learning is enabled: the router purges learned routes from its routing table if they are not readvertised by a neighbor within three times the update time interval. The UpdateTime parameter also defines, in seconds, the interval at which the router generates NRIP and SAP updates. Valid settings are integer values from 10 to 65535; for most situations, the default setting of 60 seconds is sufficient.

CAUTION: *To avoid loss of connectivity, make sure all nodes on the network are set to the same UpdateTime value. Because NetWare servers use a default of 60 seconds, make sure all nodes on the network are set to the same value especially if you have NetWare servers on your network.*

Set the update time interval using:

```
SETDefault !<port> -NRIP UpdateTime = <seconds>(10–65535)
```

or

```
SETDefault !<port> -SAP UpdateTime = <seconds>(10–65535)
```

A higher value for the UpdateTime parameter increases the time it takes for all routers on the network to converge on the same topology and allows dynamic learning to occur. A lower value reduces the time it takes for the convergence to occur, at the expense of network overhead. All routers on the same network must have the same UpdateTime value.

Another way to solve this problem is to run NLSP instead of NRIP and SAP.

Flushing Dynamic Routes and Server Table Entries

If you are experiencing connectivity problems due to routing and service tables that are not synchronized between IPX routers on your internetwork, you can flush the route and server table entries instead of waiting for them to time out or powering down each router.

To remove all entries learned dynamically from the routing table, enter:

```
FLush -IPX AllRoutes
```

To remove all entries from the server table, enter:

```
FLush -IPX AllServers
```

For more information on the FLush command, refer to Chapter 1 in *Reference for NETBuilder Family Software*.

Flushing Dynamically Learned WAN Neighbors

IPX allows dynamic learning of neighbors from its WAN interface. If you need to change the WAN interface type, you can also flush the dynamically learned neighbors. To flush the neighbor table, use:

```
FLush !<port> -IPX ADDRess
```

If further regular updates are not received, dynamically learned WAN neighbors can also be aged out of the neighbor table.

Built-in IPX Masks

Table 13-2 lists the built-in IPX masks. These predefined masks identify different types of IPX packets. To display this table, enter:

```
SHow -FIlter MASK BuiltIn
```

Table 13-2 Built-in IPX Masks

Built-in Mask	Use
IPXRIP	Matches a RIP Packet
SAP	Matches a SAP packet
FSP	Matches a Netware File Service NCP packet
WANBC	Matches a broadcast packet of IPX packet type 20
TRACERT	Matches a 3Com-proprietary Trace packet (soc = 0x874e)
IPXPING	Matches an IPX Ping packet (soc = 0x9086)
IPXDIAG	Matches an IPX Diagnostic packet (soc 0x456)
NWSEC	Matches a Netware Security packet (soc = 0x457)

User-defined IPX Masks

Table 13-3 lists valid IPX field mnemonics and match values. You can use these field mnemonics to specify the offset or location in an IPX packet. ALL is a valid match mnemonic for certain field categories. When ALL is specified, any value in the location is considered to match the criteria. The percent sign (%) is used to specify a hexadecimal value; otherwise, the value is considered to be decimal.

To display a list of valid locations supported for IPX, enter:

```
SHow -FIlter MNEmonics
```

Table 13-3 IPX Built-in Mnemonics for User-defined Masks

Field	Description	Matching Value
DstNETwork	IPX destination network	%<hexadecimal network number> ALL
SrcNETwork	IPX source network	%<hexadecimal network number> ALL
NETwork	Either IPX destination or source network	%<hexadecimal network number> ALL
DstNodeAddr	IPX destination node address	%<hexadecimal node address>
SrcNodeAddr	IPX source node address	%<hexadecimal node address>
NodeAddr	Either IPX destination or source node address	%< hexadecimal node address>
DstSocKeT	IPX destination socket	FileServicePacket ServerAdvPkt RouteInfoPkt IpxPING IpxDIAG IpxTraceRt NWSecPkt %<hexadecimal socket number>
SrctSocKeT	IPX source socket	FileServicePacket ServerAdvPkt RouteInfoPkt IpxPING IpxDIAG IpxTraceRt NWSecPkt %<hexadecimal socket number>
SocKeT	Either IPX destination or source socket	FileServicePacket ServerAdvPkt RouteInfoPkt IpxPING IpxDIAG IpxTraceRt NWSecPkt %<hexadecimal socket number>
PacketLength	IPX packet length	%<hexadecimal value>
PacketType	IPX packet type	%<hexadecimal value>
TransportCtl	IPX transport control	%<hexadecimal value
DATA+[%]<offset> [:[%]<length>]	Starting <offset> bytes after the end of the IPX header and <length> bytes long	%<hexadecimal value> <“ascii string”>

The maximum length allowed for a string-match value is 10. For a numerical value, only 1, 2, or 4 are valid lengths.

Adding and Deleting Static Routes

Routes dynamically learned are automatically purged from the routing table if they are not readvertised within a certain period of time (for details refer to “Controlling Route and Service Aging” on page 13-15).

If you want to add a route to the routing table that will not be purged from the table, eliminate route advertisements required for dynamic route learning, and

optimize the use of the available bandwidth on slow serial data links, you must add the route as a static route.

If a destination network is reachable with both a static route and a learned route, the router uses the static route unless you specify the optional Override value in the ADD ROUte command. If a learned route of higher precedence is available, it overrides the static route.

If you want to eliminate NRIP and SAP advertisements (bandwidth protection), you can configure the bridge/router for nonperiodic updates through the -NRIP and -SAP CONTrol parameters. For more information, refer to "Controlling NRIP and SAP Updates" on page 13-14.

The IPX router ignores any dynamic updates or backup routes on the network when a static route is configured for a specific network. Static routes are recommended only where the network topology remains constant.

Prerequisites

For each router port on which you want to add static routes, you must configure a network number (refer to "Setting Up a Basic IPX Router" on page 13-1).

Procedure

Define a static route using:

```
ADD !<port> -IPX ROUte {&<remote network> | Default} [<network>]
  <media address> <hops> [hdrfmt]
```

Figure 13-3 shows router 1 (Santa Clara Office) that can reach three remote routers through different media:

- Router 2 (Santa Clara Branch) is reachable by LAN.

  Whenever token ring is involved, as in this example, make sure that the ProtMacAddrFmt parameter is set to the correct address format.
- Router 3 (Los Angeles Branch) is reachable by X.25.
- Router 4 (New York Branch) is reachable by Frame Relay.
- Router 5 (New Jersey Branch) is reachable by SMDS indirectly through router 4.

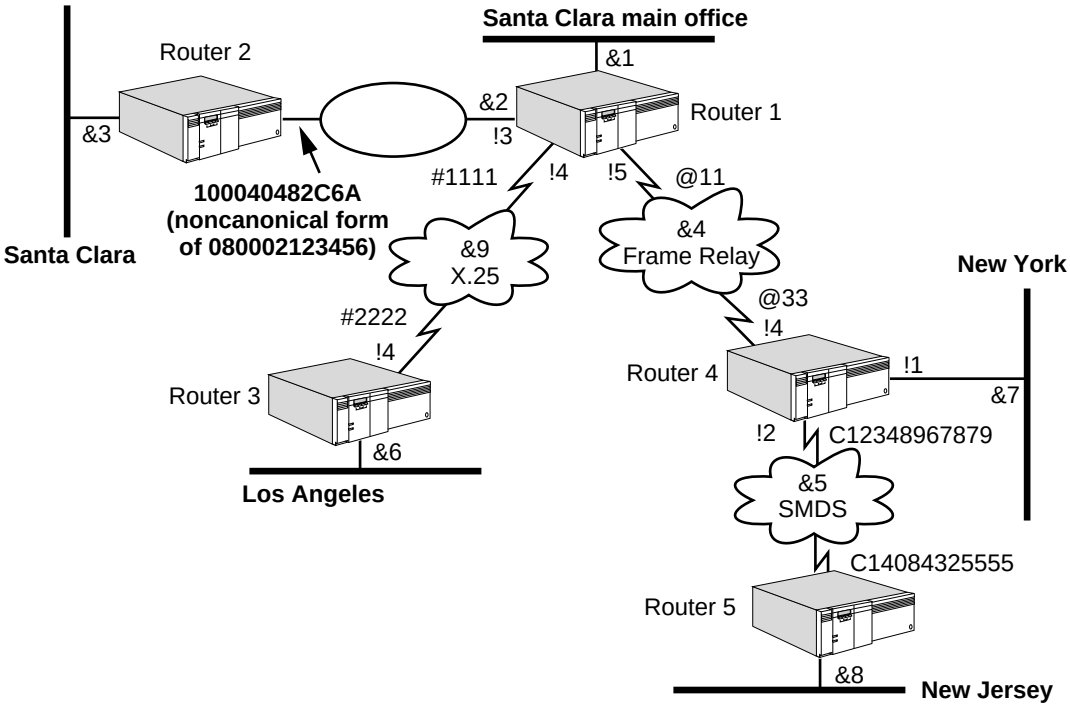


Figure 13-3 Adding Static Routes

Steps 1 through 4 of the following procedure are performed either from a console attached to router 1 or via a Telnet connection to router 1. To communicate with router 1 via Telnet, router 1 must have an IP address that is reachable from the workstation or router console from which the Telnet connection is initiated.

On router 1, follow these steps:

1 Add a static route to Santa Clara (network &3) by entering:

```
ADD !3 -IPX ROUte &3 %100040482C6A 1
```

This command specifies that network 3 is reachable through the device identified by MAC address %100040482C6A on network &2, and that the route to network &3 has a hop count of 1. The command is identical for all neighbors reachable through LAN connections (Ethernet, FDDI, or token ring) except the MAC address, which must be set appropriately depending on the -PORT ProtMacAddrFmt parameter value.

2 Add a static route to Los Angeles (network &6) by entering:

```
ADD !4 -IPX ROUte &6 #2222 1
```

3 Add a static route to New York (network &7) by entering:

```
ADD -IPX ROUte &7 &4 @33 1
```

4 Add a static route to New Jersey (network &8) by entering:

```
ADD !5 -IPX ROUte &8 @33 2
```

The routes to networks &7 and &8 (defined in steps 3 and 4, respectively) are identical except for the destination network identifier and the hop count. This is because the "next hop" for any packet routed by router 1 to either network &7 or &8 is router 4.

5 If dynamic learning is disabled on router 4, you will also need to add a static route from Router 4 to network &8 by entering:

```
ADD !2 -IPX ROUte &8 $C14084325555 1
```

To display the static routes configured far, enter:

```
SHow -IPX ROUte
```

Because static routes do not age out, they must be removed manually. To delete a static route, use:

```
DELete -IPX ROUte &<remote network>
```

Configuring a Static Default Route

You can configure a static default route, which is subsequently added to the routing table and propagated by NRIP or NLSP. Once a default route is specified, packets destined to unknown networks (networks not explicitly known or listed in the routing table) are routed to the default router for subsequent routing. You can configure only one default route per port.

Use this procedure to configure a default route so that unknown destination packets can be properly forwarded. For conceptual information, refer to "Default Routes" on page 13-36.

Procedure

To configure a default route, see to Figure 13-4 and follow these steps:

1 Assign a default route on the router port to point to the default router using:

```
ADD !<port> -IPX ROUte Default <media address> <hop>
```

Substitute the MAC address of the default router for <media address>.

For example, in Figure 13-4, configure the ROUte parameter on port 3 of Router B and use the MAC address of Router A. Router B adds a static route (labeled as the default route) to its routing table and advertises the route to downstream routers (Router C and Router D). When Router B receives an unknown destination packet (for example, a client on network &600 transmits a packet destined for network &1000), Router B uses the default route, and routes the packet out port 3 to the default router, which sends the packet toward its destination.

The special network number &FFFFFFFE has been reserved for the default route. If this address has already been used within an organization, it must be renumbered.

The default route implementation for NRIP is not supported in software versions prior to version 8.2. All routers must be upgraded to software version 8.2 or later so that all NRIP routers recognize &FFFFFFFE as the default route and forward packets for unknown destinations toward it.

2 Verify that the default route has been added to the routing table of the remote router by entering:

```
SHow -IPX AllRoutes
```

The default route is the first route in the routing table and is labeled Default.

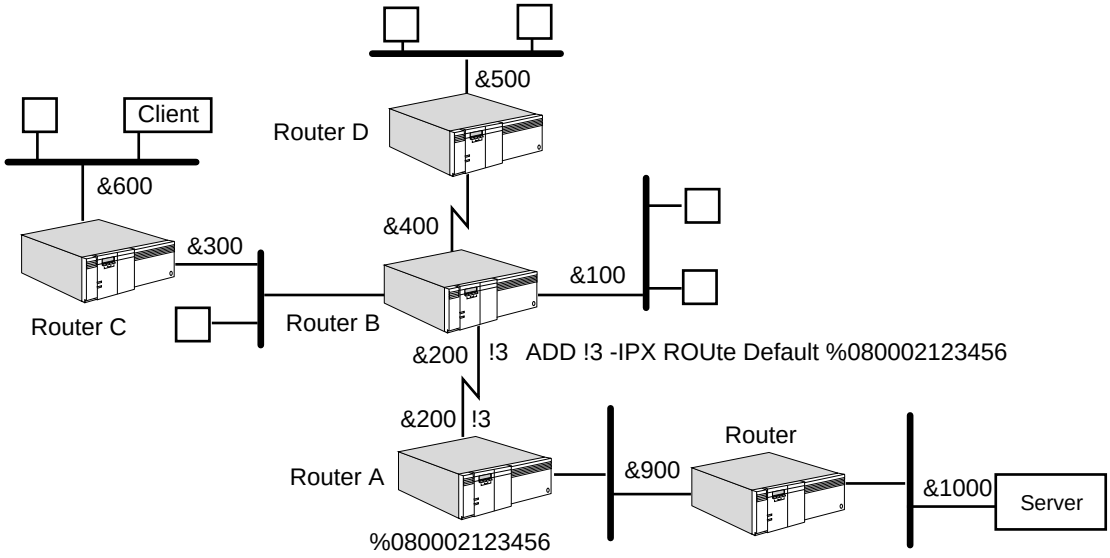


Figure 13-4 Configuring an IPX Default Route

Configuring a Default Metric

You can configure a default metric on a router to advertise a default route to other routers. The default metric allows default route advertisements to be sent in RIP updates. Other routers, receiving such advertisements, send all unknown destination packets to this router. Without default route advertisements, unknown destination packets are dropped before they can reach this router.

To configure a default metric on a router and enable default route advertisement, see to Figure 13-5 and use:

```
SETDefault !<port> -NRIP DefaultMetric = <hops(1–15)> [<ticks>]
```

For <hops>, select a value between 1 and 15 to enable advertisement of this route within the corresponding hop count. For example, on port 3 of router A, configure a default metric of 1.

Optionally, for <ticks>, you can select a value between 1 and 65535. If there is more than one route to the same destination, the router uses the one that has the lowest tick value.

For example, router A advertises the default route over port 3. When router B receives the advertisement, it adds a static default route to its routing table and propagates the metric to the other downstream routers (routers C and D). When the downstream routers receive unknown destination packets, they route them tor router B, which uses the default route in its routing table to route the packet over port 3.

By using the default metric, the routing tables of the remote routers (routers C and D in the figure) can be reduced in size; the routing tables of routers C and D do not need to contain routes to router A or have knowledge of other networks that are attached to router A.

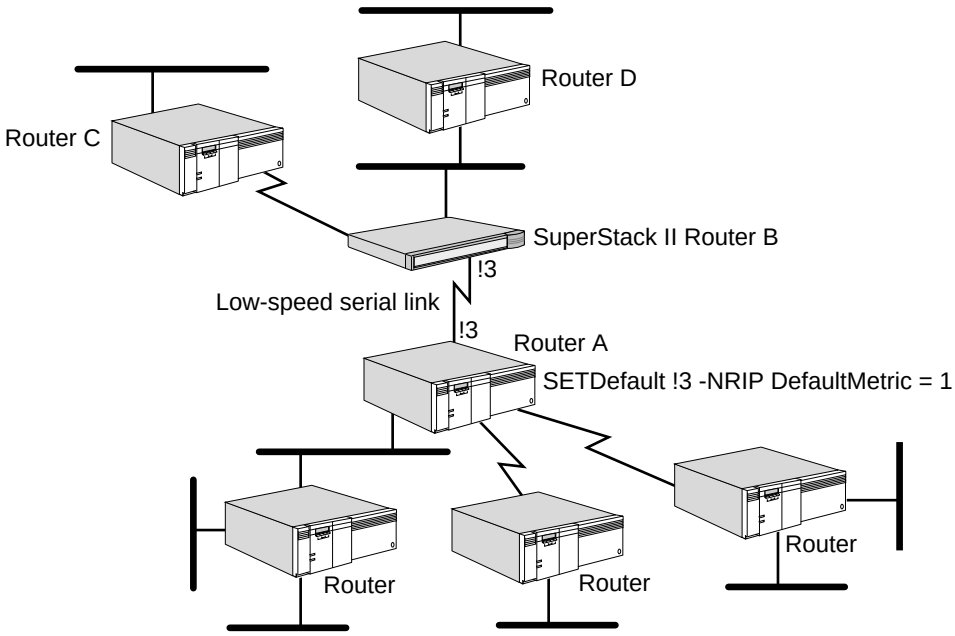


Figure 13-5 Configuring an IPX Default Metric

Adding and Deleting Static Servers

The IPX Service allows you to enter static servers into the SAP information table. The static server will not be aged out, so it will not be purged in the aging-out time frame. The IPX router dynamically updates the server once a static server is configured. Specifying a static server is recommended only when the network topology remains constant.

Define a static route using:

```
ADD -IPX SERver <sname> <type> <snet>%<shost>:<sskt> <hops>
```

Where <sname> is the name of the static server being configured, <type> is the type of service, <snet>%<shost>:<sskt> is the server address and <hops> is the hop count away from this IPX router. For more information, refer to "SERver" on page 31-9 in *Reference for NETBuilder Family Software*.

Configuring Neighbor Policy

When you enable route advertisements to neighbors by setting the -NRIP and -SAP PolicyControl parameters to AdvToNbr, broadcast NRIP and SAP updates are automatically disabled, and only those neighbors specified with the RcvFromNbr attribute receive unicast NRIP and SAP updates. The NRIP and SAP Services can maintain a different neighbors list.

There are two reasons to configure neighbors:

- To use the next-hop split horizon scheme on a neighbor basis, as described in "How the IPX Router Works" on page 13-33.
- To control routing domains for security (advertise routes only to specified neighbors).

To configure neighbors, follow these steps:

1 Enable AdvToNeighbor using:

```
SETDefault !<port> -NRIP PolicyControl = AdvToNbr
```

2 If dynamic neighbor learning is enabled using the CONTrol parameter, and if the port is configured for Frame Relay or X.25, then this step can be skipped. Specify all of the neighbors to which NRIP updates are to be sent using:

```
ADD !<port> -NRIP AdvToNeighbor <network>%<mac address> [...]
```

If the physical connection is made with a 3Com bridge/router that has an HSS module installed, use the MAC address of the HSS module interface connecting the neighbor to the network.

Writing NRIP and SAP Policies for IPX

Using NRIP and SAP policies can provide security, reduce route and service table sizes on file servers and bridge/routers, and help reduce excessive traffic across WAN links. The NRIP and SAP Services can maintain different policies. The policies consist of lists of network numbers, service names, and SAP types. Lists can be created as *normal lists* or *inverse lists*. Normal lists list every network number, server name, and service type that is included in a policy. Inverse lists list every network number, server name, and service type that is excluded from a policy. For background information on policies, refer to "Route, Service, and Neighbor Policies" on page 13-41. For a listing of Novell service advertising type descriptions, refer to Table 13-4 in "Novell Service Types" on page 13-46.

Figure 13-6 shows a NETBuilder II bridge/router and a SuperStack II NETBuilder bridge/router serving different networks in which IPX is being used for NetWare environments. Refer to the figure in the examples that follow.

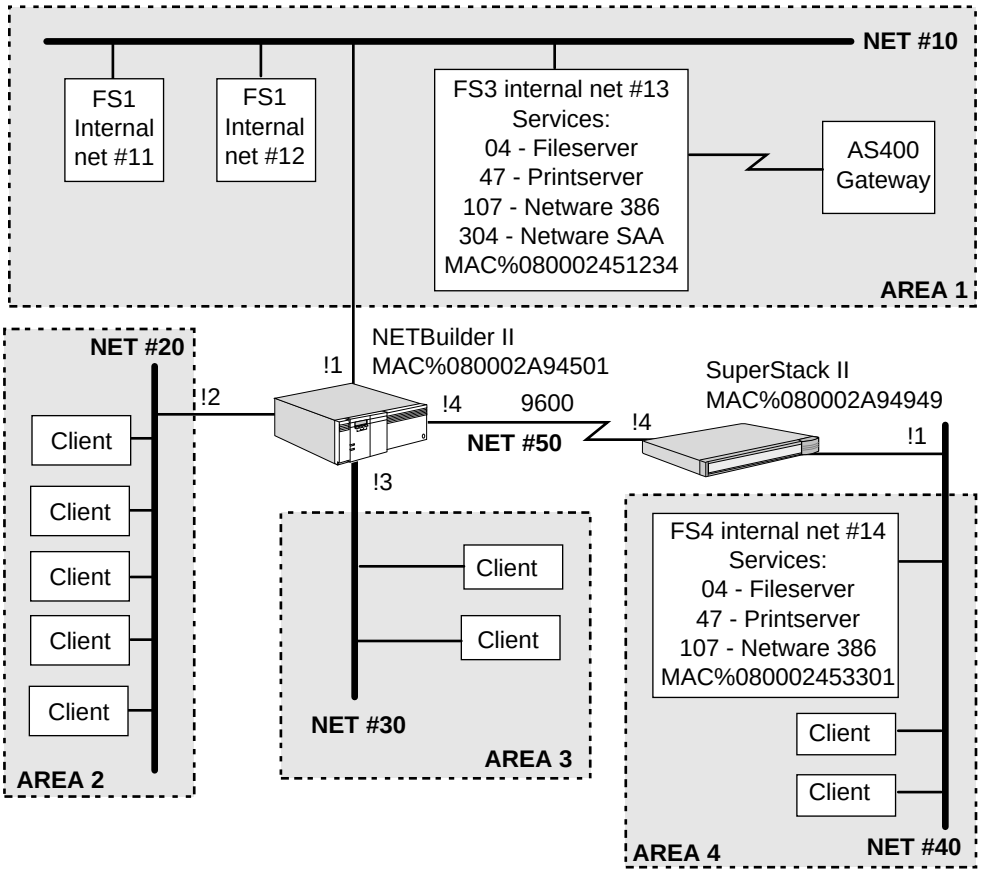


Figure 13-6 Using RIP and SAP Policies in IPX Environments

NETBuilder II Examples

You can configure the NRIP and SAP policies in different ways on the NETBuilder II bridge/router, as shown in the following examples.

Example 1 Because area 2 and area 3 do not need to access the file server in area 4, and clients in area 4 do not need to communicate with clients in area 2 or area 3, their routes do not need to be broadcast across the 9600 baud link. In this situation, you could set up a route policy on the NETBuilder II bridge/router to keep traffic off the 9600 baud link by entering:

```
ADD !4 -NRIP AdvertisePolicy ~&20-30
SETDefault !4 -NRIP PolicyControl = AdvPolicy
```

The result of these commands, is that all networks except 20 and 30 are advertised to area 4.

Example 2 If no remote segments need to access the two file servers in area 1, you could write a route policy to keep the NETBuilder II bridge/router from receiving the internal IPX net number of the file server. To write such a route policy, enter:

```
ADD !1 -NRIP ReceivePolicy ~&11-12
SETDefault !1 -NRIP PolicyControl = RcvPolicy
```

Example 3 If you want other segments to have access to area 1, but not to all the services available on all file servers in area 1, you could set a SAP policy by following these steps:

1. Configure the advertise policy using:

   ```
   ADD !<port> -SAP AdvertisePolicy
   ```

   If you want to advertise only file server 3, you can enter the address used in the policy in three different ways:

   - By specifying the internal IPX net, server host address, and service type.

     For example, to set the advertise policy in this way, enter:

     ```
     ADD !4 -SAP AdvertisePolicy &0000013:%000000000001:04
     ```

     In this example, :04 indicates the service type in Figure 13-6, for example, file server.

     The server address is not the 48-bit MAC address of the host on which the service is located. NetWare servers usually advertise themselves with address 000000000001. To determine the address of the server, enter:

     ```
     SHow -IPX AllServers Long
     ```

     If your NetWare servers advertise themselves with address 000000000001, specifying this address filters all servers on the network. To filter a server individually, specify it by server name.

   - By specifying the actual file server name and service type.

     For example:

     ```
     ADD !4 -SAP AdvertisePolicy “FS3”:04
     ```

     You can also advertise all services from FS3 by using the asterisk character as a wildcard, as shown in the following example:

     ```
     ADD !4 -SAP AdvertisePolicy “FS3”:*
     ```

- By specifying the policy number command.

For example, if FS3 is using NetWare 2.X, in which no internal IPX network numbers are used, enter the policy number command specifying the IPX network number, server MAC address, and service type as follows:

```
ADD !1 -SAP AdvertisePolicy &00000010:%000000000001:04
```

Instead of entering:

```
ADD !1 -SAP AdvertisePolicy &0000013:%080002451234:04
```

The result of these commands is that only file server 3 is advertised to area 4. No other file servers are advertised to area 4.

2 Set the policy control to enable using:

```
SETDefault !<port> -SAP PolicyControl = AdvPolicy
```

3 To view addresses on the file servers, enter:

```
SHow -IPX AllServers Long
```

Example 4 If you need to access some, but not all services on all networks, then a SAP policy can be used to control the specific services that are available. The SAP policy on the NETBuilder II bridge/router could advertise type 4 file services and AS400 gateway (type 304) but not print service type 47 or NetWare 386 (type 107).

To keep SAP broadcasts type 47 from being received on port 1 of the NETBuilder II bridge/router (no printer SAP type 47 or NetWare 386 (type 107) from server FS3 but still receiving the file service type 4 and the AS400 gateway type 304), set the policy by entering:

```
ADD !1 -SAP ReceivePolicy ~"FS3":47
ADD !1 -SAP ReceivePolicy ~"FS3":107
SETDefault !1 -SAP PolicyControl = RcvPolicy
```

Example 5 If area 2 and area 3 need printer services available, but you do not want them advertised out to area 4, enter:

```
ADD !4 -SAP AdvertisePolicy ~"FS3":47
SETDefault !4 -SAP PolicyControl = AdvPolicy
```

SuperStack II Examples

You can configure the NRIP and SAP policies in different ways on the SuperStack II bridge/router, as shown in the following examples.

Example 1 You can define a route policy on the SuperStack II bridge/router to keep unnecessary packets off the local area network. To configure this route policy, enter:

```
ADD !1 -NRIP AdvertisePolicy ~&50
SETDefault !1 -NRIP PolicyControl = AdvPolicy
```

In this situation, the net number for the WAN link does not need to be broadcast out on the local area network.

Example 2 You can add a SAP policy for restricting the advertisement of other services (the print server and the NetWare 386) available on FS4 file services by entering:

```
ADD !1 -SAP ReceivePolicy "FS4":04
SETDefault !1 -SAP PolicyControl = RcvPolicy
```

In this situation, the receive policy keeps the SuperStack II bridge/router server from even storing the other services in its SAP table. The SuperStack II bridge/router will not respond to SAP request for service type 47 (printer) or any service on FS4 except type 04 file services.

Example 3 If clients in area 1, area 2, and area 3 do not need access to area 4, you can set a policy for this situation by entering:

```
ADD !4 -NRIP AdvertisePolicy ~&40,~&14
SETDefault !4 -NRIP PolicyControl = AdvPolicy
```

Note that in this example, &14 is the internal IPX number for FS4.

Configuring Other Policy Settings

You can configure other settings for NRIP and SAP policies, including setting up lists of IPX neighbors (next hop routers, or file servers broadcasting NRIP and SAP packets). These lists are defined by the MAC address and are used to determine who the router should accept from or advertise to the NRIP and SAP information.

These policy settings are configured using:

```
ADD !<port> -NRIP or -SAP AdvToNeighbor
ADD !<port> -NRIP or -SAP RcvFromNeighbor
```

These syntaxes are used in the same way that the following syntaxes are used in this section:

```
ADD !<port> -NRIP or -SAP AdvertisePolicy
ADD !<port> -NRIP or -SAP ReceivePolicy
```

The AdvToNeighbor parameter cannot accept the ~ (inverse) policy.

One example of where to apply a neighbor policy as shown in Figure 13-6 on page 13-23 is to have the bridge/router receive NRIP broadcasts from only properly configured file servers to protect itself from servers that may send conflicting NRIP and SAP broadcasts. For example, to receive NRIP and SAP information only from FS3, enter the following commands on the NETBuilder II system:

```
ADD !1 -NRIP RcvFromNeighbor %080002451234
SETDefault !1 -NRIP PolicyControl = RcvFromNbr
```

You can use the PolicyOverride setting to override the configured policies on locally connected routers (not to be used across serial links) when the router issues responses to specific NRIP or SAP requests. Use the PolicyOverride setting when one side of a router is all clients that do not need to see NRIP and SAP broadcasts, but the 3Com bridge/router still needs to respond to RIP and SAP requests. In this case, AdvertisePolicy is set for both NRIP and SAP, and PolicyControl is configured with the AdvPolicy RcvPolicy, and PolicyOverride settings.

In this situation, on the 3Com bridge/router, leave the route list blank for the AdvertisePolicy parameter, and enable PolicyControl for both route and services being advertised. No NRIP or SAP broadcasts are advertised on port 2, but with policy override enabled, the 3Com bridge/router can still respond to the client's specific request for connections to the file servers. To set this configuration, enter:

```
SETDefault !2 -NRIP PolicyControl = (AdvPolicy, RcvPolicy,
 PolicyOverride)
```

Configuring IPX Spoofing over a DOD Link

To help you better control IPX traffic over DOD lines, software version 9.1 and later can spoof NetWare 3.0 and 4.0 NetWare Core Protocols (NCP) KeepAliveRequest and Sequenced Packet Exchange 1 (SPX1) keepalive packets to reduce the time a DOD line is kept in the up state. Spoofing these packets makes more efficient use of the DOD WAN links.

NCP Spoofing over a DOD Link

An NCP connection between a NetWare client and a server is maintained through an exchange of KeepAliveRequest and Response packets between the two. For each client connection, the server maintains information about the connection. For example, any connection that is idle or no longer used is terminated by the server, and all resources that were allocated to that connection can be reused. The server determines that a connection is no longer needed as described in the following paragraphs.

The server sends a KeepAliveRequest packet to the client to see if it is still attached to the server after a time interval has elapsed since the server last sent a NCP request or received a data transfer from the client. For example, in Figure 13-7, server S10 sends KeepAliveRequest packets to clients C1 and C20; server S24 sends KeepAliveRequest packets to client C1.

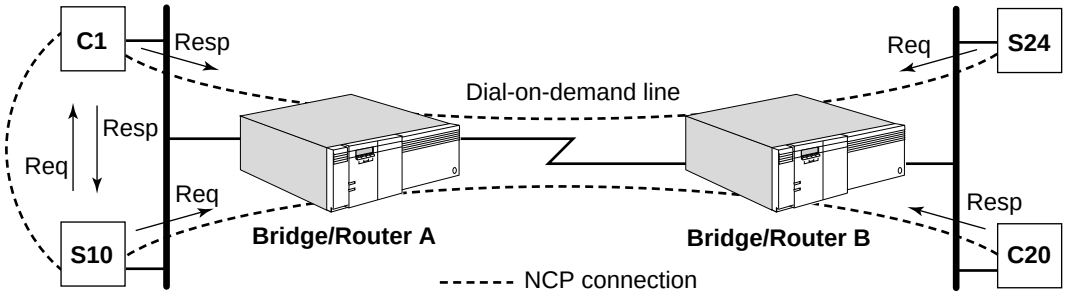


Figure 13-7 NCP KeepAliveRequest Packet Exchange

If the server receives the KeepAliveResponse packet from the client within a certain time interval, then the client is still considered to be a client and its connection to the server is maintained. For example, in Figure 13-7, C1 and C20 respond to the KeepAliveRequest packets from S10 by sending KeepAliveResponse packets if they are still active; C1 also responds to the KeepAliveRequest packet from S24.

If the server does not receive a KeepAliveResponse packet from the client within a certain time interval, then the server continues to resend the KeepAliveRequest at regular time intervals until it either receives a response or it has exhausted its KeepAliveRequest retry counts. In the latter case, the client is considered to be no longer a client and its connection is terminated.

On NetWare 3.0 and 4.0 servers, the number of KeepAliveRequest retries, time interval (delay) before sending the first request, and the time interval between retries are all user-configurable parameters.

NCP Keep Alive Mechanism

In a LAN environment and on non-DOD WAN lines where the path is always up, the keep alive mechanism operates properly. On DOD WAN lines, the function of DOD is to bring the path down and, in the absence of any other traffic over this path, keep the path down to reduce phone charges. In contrast, the sending of KeepAliveRequest packets by the server to query its client brings the path up (and down) constantly, and may also prevent a DOD path from idling to a down state.

To resolve these problems, spoofing of the NetWare 3.0 and 4.0 NCP keep alive packets has been implemented. First available in software version 8.0, spoofing is a mechanism that allows the bridge/router to respond to an incoming KeepAliveRequest packet that is to be routed over a DOD line, by sending a KeepAliveResponse packet to the originating server of the request on behalf of the intended client. The bridge/router with spoofing software spoofs only when the DOD path is down to prevent the DOD path from constantly coming up and going down due to the transmission of KeepAliveRequest packets from the server. When the DOD path is up, the bridge/router routes the KeepAliveRequest and Response packets as expected in the normal NCP connection process.

For example, in Figure 13-8, a quiet NCP connection over an idle DOD path exists between C1 and S24, and between C20 and S10. With spoofing, router A responds to the request from S10 to C20 for the client. The request packet is intercepted and processed by the IPX software, replied to, and discarded without being transmitted over the DOD line. The discarded packet does not trigger the raising of the DOD path as a normal routed packet does. Similarly, Router B spoofs the request from S24 to C1. With server S10 and client C1, the normal NCP connection process occurs; the bridge/router performs spoofing only across a DOD link, not on a LAN.

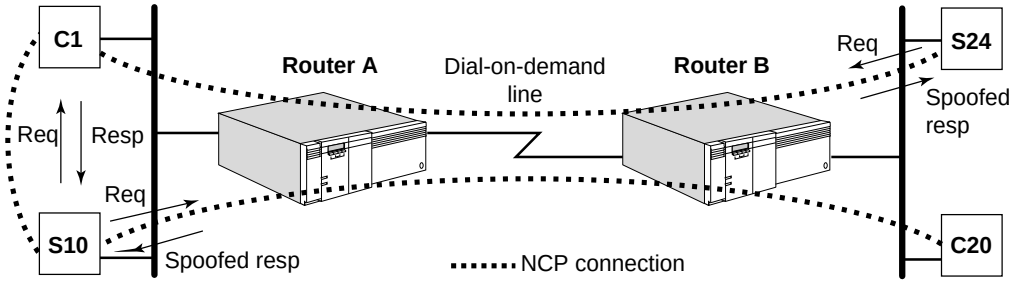


Figure 13-8 Spoofing of KeepAliveRequest Packets over DOD Paths

The maximum number of spoofed client-server connections that are handled by the router is not limited.

In the NCP keep alive mechanism, the clients are totally passive; it is up to the server to maintain the connections based solely on the responses to its requests. This characteristic allows the spoofing software to spoof only on the server side (spoof response) without having to worry about the client side (spoof request).

Supported Configurations

The NCP spoofing feature can be used with DOD in the following IPX network configurations:

- Router-to-router configuration
- Boundary Routing configuration

In a router-to-router configuration, NCP spoofing can be used to support symmetrical two-way client-server access. For example, clients on either side of a DOD line can access servers on either side of the line with NCP spoofing of the server's keep alive requests. For example, in Figure 13-9, router A spoofs the connection between C1 and S2 on port 3; router A spoofs the connection between C2 and S3 on port 3; router C spoofs the connections between C3 and S1. There is no spoofing between S3 and C5.

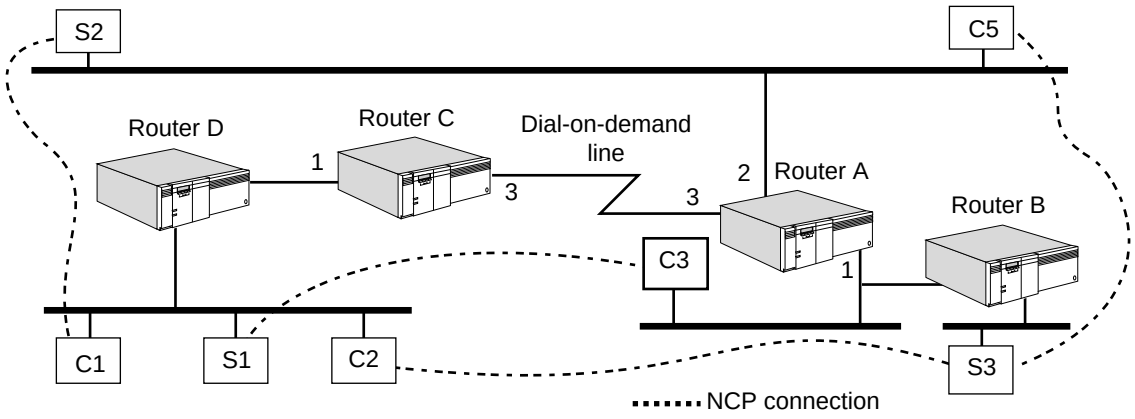


Figure 13-9 NCP Spoofing over DOD Lines in a Router-to-Router Configuration

In a Boundary Routing configuration, NCP spoofing can be used to support remote clients' access to central site servers, and also central site clients' access to remote servers. For example, in Figure 13-10, router A acting as a central node spoofs the connection between C1 and S1 on port 3; router A also spoofs the connections among C2, S2, and S3 on port 3. For the connection between S10 on the peripheral network and C3 on the central site, the SuperStack II boundary router peripheral node spoofs NCP keepalive packets on port 2.

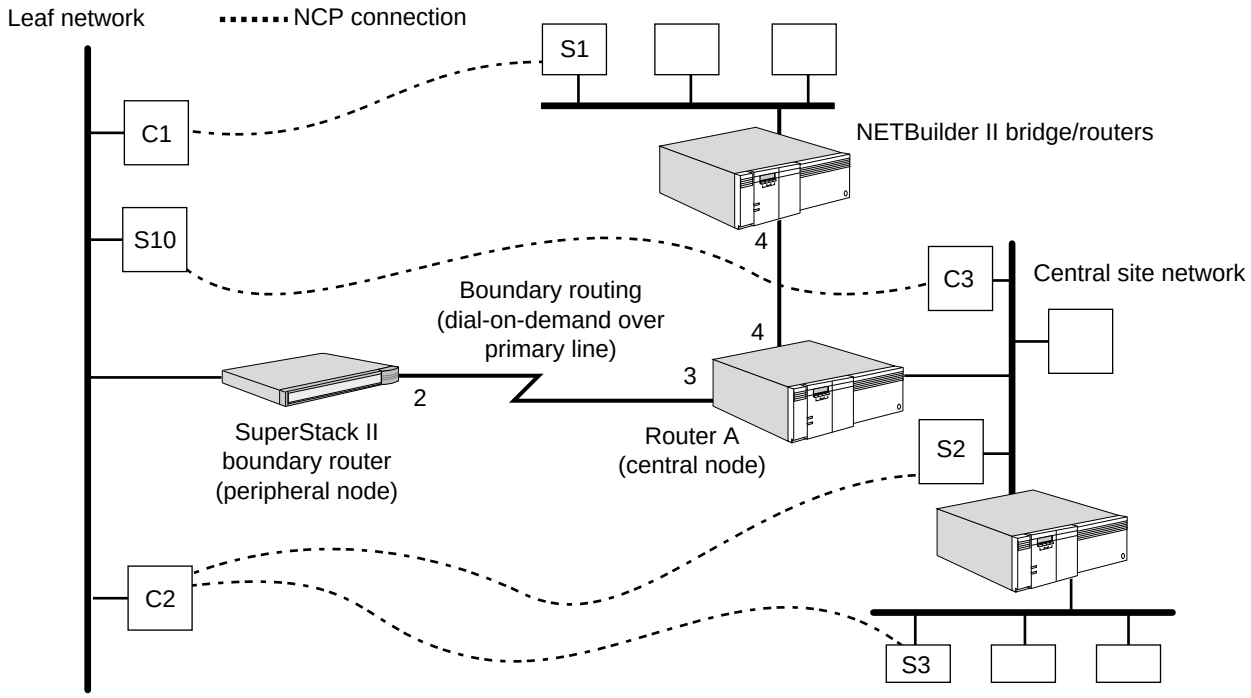


Figure 13-10 NCP Spoofing over DOD Lines in a Boundary Routing Configuration

SPX1 Spoofing Lite over a DOD Link

Sequenced Packet Exchange 1 (SPX1) is a transport-level connection protocol used by certain applications in the NetWare environment. An SPX1 client initiates a connection to an SPX1 host to transfer data with guaranteed delivery. This is done by transmitting an acknowledgment (ACK) packet. Depending upon the application, these data transfers can be bursty followed by long periods when the connection is quiet. To maintain this quiet period, SPX1 uses a process where an exchange of the ACK packets is performed by two SPX1 connection end nodes. These packets are referred to as *keepalive* or *watchdog* packets. When the watchdog packets are transmitted over a DOD line, the line is kept up unnecessarily, incurring extra costs. SPX1 Spoofing Lite is the 3Com solution to this problem. During the non-data transfer keepalive period, SPX1 Spoofing Lite spoofs SPX1 watchdog packets using an allocation window of 1. Only those applications such as Novell Rconsole or Lotus Notes that are not sensitive to this allocation setting are supported. It is also recommended that SPX1 Spoofing Lite be used only with ISDN DOD links. For some applications, the link must be brought up whenever there is SPX1 data to be exchanged within a short interval or the application will time out.

In Figure 13-11, SPX1 end nodes A and B exchange SPX1 watchdog packets to prevent their internal timers from expiring during times when there are no active SPX1 data transfers. If these timers expire, the connection will be aborted. When the connection is over a DOD link, the packets keep the line up.

When SPX1 Spoofing Lite is enabled with the -IPX SPoofCONTrol parameter, all SPX1 packets to be forwarded out the DOD port are intercepted and processed, as follows:

- All data packets and all system packets except the watchdog packets are always forwarded across the DOD link.
- If the link is down, DOD brings the line up and the packets are forwarded.
- Watchdog packets are forwarded only when the link is up.
- When the link is down, as occurs during quiet times, watchdog packets will be discarded and spoofed.

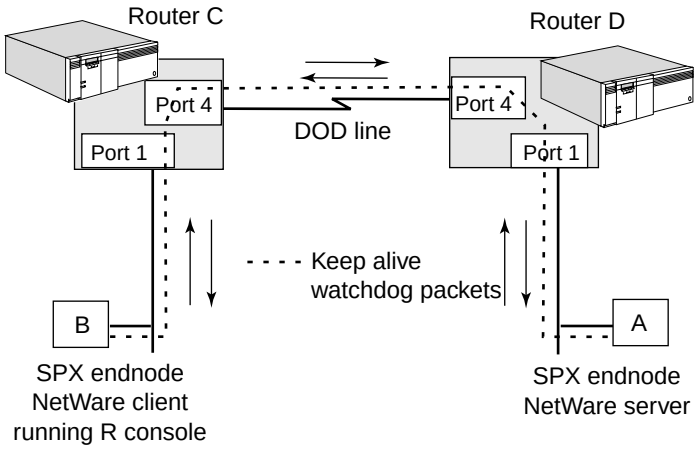


Figure 13-11 SPX Watchdog Packets over a DOD Link

In Figure 13-12, when the DOD link is down, watchdog packets from end node A are intercepted by router D and recycled as spoofed watchdog packets, then sent back out on port 1 to end node A. Watchdog packets from end node B are intercepted by router C and recycled as spoofed packets and sent back out on port 1 to end node B.

With SPX1 Spoofing Lite, watchdog packets are spoofed with a an SPX1 allocation window of 1. Only those applications that can handle this allocation size (Novell Rconsole, Lotus Notes, etc.) are supported. Use the -IPX SPoofCONTrol parameter to enable or disable spoofing on each port. For non-DOD ports, spoofing does not apply and is always disabled.

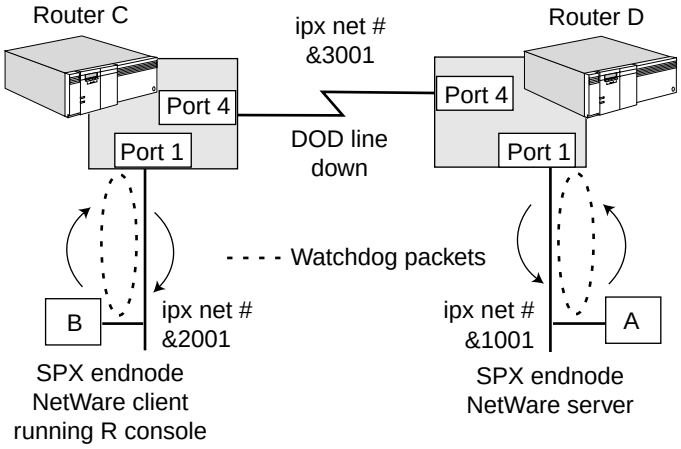


Figure 13-12 Spoofing SPX Watchdog Packets

Supported Configurations

The SPX1 Spoofing Lite feature can be used with DOD in the following IPX network configurations:

- Router-to-router
- Boundary Router central node to Boundary Router peripheral node

Configuring SPX1 Spoofing Lite over a DOD Link See Figure 13-12 for an illustration of the following configuration. To configure SPX1 Spoofing Lite, follow these steps:

1. Enable IPX routing on router C and router D by entering:

```
SETDefault !1 -IPX CONTrol = ROute
SETDefault !4 -IPX CONTrol = ROute
```

2. Assign the IPX network numbers for port 1 and port 4 on router C by entering:

```
SETDefault !1 -IPX NETnumber = &2001
SETDefault !4 -IPX NETnumber = &3001
```

3. Assign network numbers for port 1 and port 4 on router D by entering:

```
SETDefault !1 -IPX NETnumber = &1001
SETDefault !4 -IPX NETnumber = &3001
```

4. Use incremental NRIP and SAP to reduce broadcast traffic on port 4 of both bridge/routers by entering:

```
SETDefault !4 -NRIP CONTrol = (Talk, Listen, NoPEriodic)
SETDefault !4 -SAP CONTrol = (Talk, Listen, NoPEriodic)
```

5. Enable SPX1 Spoofing Lite on both bridge/routers by entering:

```
SETDefault !4 -IPX SPoofCONTrol = Spx1WatchDog
```

6. Verify that SPX1 Spoofing Lite is enabled by entering:

```
SHow !4 -IPX SPoofCONTrol
```

Configuring SPX1 Spoofing Lite for the Boundary Routing Peripheral Node SPX1 Spoofing Lite is disabled by default on the boundary routing peripheral node. To enable it, a special filter policy has to be configured. When configured, this policy enables SPX1 spoofing. When deleted, spoofing is again disabled. Filtering does not need to be enabled for this special policy to take effect. This special policy is to be used only on the peripheral node for enabling and disabling SPX1-spoofing. On the boundary routing central node, SPX1 spoofing is enabled or disabled using the -IPX SPoofCONTrol parameter as described in the previous sections.

To enable SPX1 spoofing on the peripheral node and configure the special filter policy, follow these steps:

1. Add a user-defined mask called "SPX" for IPX packet type of 5 by entering:

```
ADD -fi ma spx ipx.pt = 5
```

2. Add the special SPX1 spoofing filter policy to enable spoofing by entering:

```
ADD -fi pol spoofspx1 dod spx
```

3 To disable SPX1 spoofing on the peripheral node, enter:

```
DELete -fi pol spoofspx1
```

Macros can be defined with these filter configurations to enable and disable spoofing. For example, a macro called SPOOFON could be defined which configures both the mask and the special policy. Another macro called SPOOFOFF can be used to delete the policy.

On a DOD link with infrequent data traffic, the bridge routes may age out because of the infrequency of packets arriving from the central site to refresh those routes. In such a situation, the ageout timer should be disabled, or its value increased, for SPX1 spoofing to function properly.

4 To disable the ageout timer, enter:

```
SETDefault -brln cont = na
```

How the IPX Router Works

The 3Com IPX router provides network connectivity between Novell NetWare clients and servers located in the same building or in distant cities. The 3Com IPX router supports a subset of Novell's NetWare communication protocols that includes the IPX Protocol, NRIP, and SAP, NLSP and minimal support of NetBIOS by propagating IPX WAN packets (packet type 0x14). However, the 3Com IPX router does not participate in any of NetWare Communication Protocol (NCP) or Sequenced Packet Exchange (SPX).

The 3Com IPX router can run over various types of data link media: Ethernet, token ring, FDDI, PLG, PPP, X.25, Frame Relay, and SMDS, and will support new media as they become available in the future. IPX has different types of encapsulation methods to run over various media. On Ethernet, four different encapsulation formats are available. The 3Com IPX router supports all of them, even simultaneously (one physical network can be segmented into four different logical networks). Additional information on what encapsulation formats are available for each medium and how to configure them, and examples are in "Configuring Secondary Networks with Different Header Formats" on page 13-2.

IPX Router Features

The 3Com IPX router offers features including various NRIP and SAP policies, manageability via SNMP, static routing capability, and next-hop split horizon and NLSP. Various parameters are available to tune the IPX router to enhance network performance by reducing network overhead. For example, the nonperiodic (incremental) update mechanism reduces the number of NRIP and SAP updates on WAN interfaces and NLSP reduces routing updates throughout your IPX network. For conceptual information, refer to "Learning Routes and Service Information" on page 13-37. For procedural information, refer to "Controlling NRIP and SAP Updates" on page 13-14.

Each IPX host is uniquely identified with an IPX Internet address that consists of two parts:

- A four-byte IPX network number
- A six-byte IPX node address

The four-byte IPX network number (represented in hexadecimal) is assigned by a network administrator. The network number must be unique throughout the IPX

Internet. Be careful not to assign duplicate networks; otherwise it causes network-wide confusion. When using NLSP, a portion of the network number also identifies the NLSP area.

The IPX node address (represented in hexadecimal) is permanently associated with each port and is not assignable except on the NetWare server's internal address. The 3Com IPX router has multiple ports and an internal network number. For instructions on assigning network numbers, refer to *New Installation for NETBuilder II Software*.

The static routing feature allows network managers to eliminate traffic associated with the route advertisements required for dynamic route learning, which frees bandwidth on slow serial data links for critical data traffic. IPX routing capability can still be achieved without sending a single NRIP update by setting the -NRIP CONTrol parameter to "NoTalk" and adding static routes on the port. Static routes can be especially cost-effective on any service where packet charges are enforced. One disadvantage of static routes is that these routes are not updated automatically. After being configured, they remain in the routing table until they are manually removed (even if the corresponding route no longer exists). For this reason, static routes are recommended only where the network topology remains constant. Another solution to this problem is to run NLSP.

The following sections provide more detailed discussions of important concepts related to IPX routing.

Local and Wide Area Network Configuration

An IPX network must be configured on each local port on which IPX packets are to be received and sent. WAN ports using PPP may be configured with or without a network number, provided an internal network number has been configured. The port can be a local Ethernet, FDDI, token ring port or a serial line port on a wide area network, such as a point-to-point link or an X.25 link. Figure 13-13 is an example showing a wide area router connecting two local Ethernet networks (Santa Clara) to two wide area networks (Los Angeles and Santa Barbara).

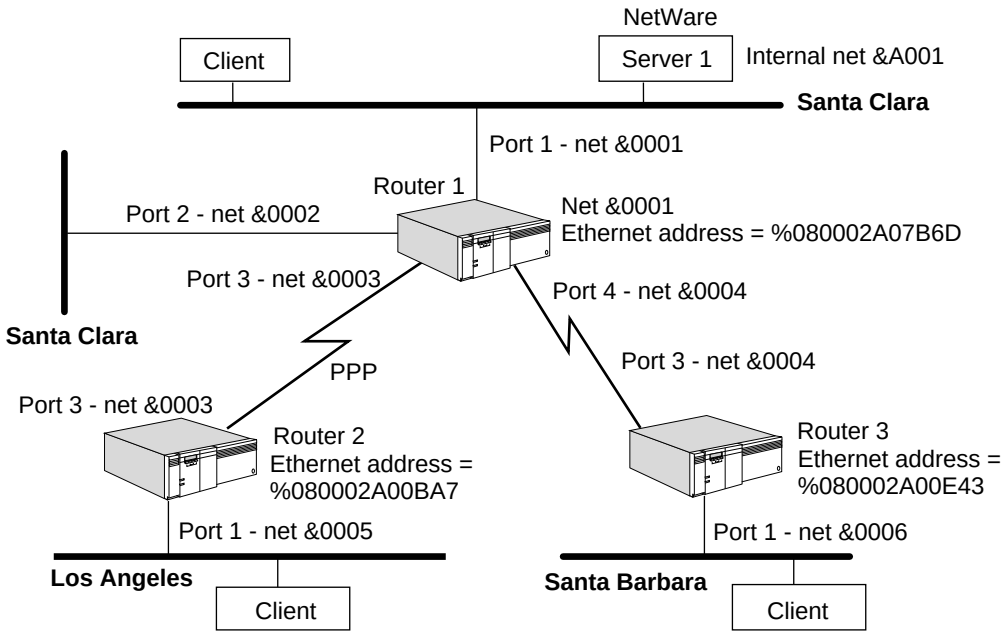


Figure 13-13 Wide Area Router Connecting Four IPX Networks

All IPX network numbers assigned must be unique within the IPX Internet.

Any physically attached network, Ethernet, or serial line is considered a directly connected network. If more than one serial line (path) is assigned to one port, that port is considered a single directly connected IPX network.

A router must look up the destination network in its routing table to determine where to route a packet. If the destination is on an attached network, the router can send it directly to the network. But if the destination is not directly connected, the router must route the packet to another router that is closer to the destination. The route to a remote network can be statically configured or dynamically learned through NRIP and NLSP routing protocols. For details, refer to "Enabling and Disabling Dynamic Learning and NRIP Updates" on page 13-13 and "Adding and Deleting Static Routes" on page 13-17.

When two routers are located on the same network (that is, each of them has at least one port to the network) they are called *neighbors*.

You can set up routing without the assignment of IP subnets. This feature is called unnumbered links. Unnumbered links are useful only between two routers; in other words, you cannot connect a router to a host using unnumbered links. For more information about unnumbered links, refer to Chapter 6.

Routing Tables

To display the routing table, enter:

```
SHow -IPX AllRoutes
```

The AllRoutes parameter has three display options. The Short option (the default) displays only network numbers and hop counts. The Long option additionally displays port numbers, network numbers, gateway addresses, hops, and costs. If you specify a network number for the NETnumber option, the port number, gateway address, hop count, and costs for the specified network are displayed. Refer to Chapter 31 in *Reference for NETBuilder Family Software* for information on the AllRoutes command.

Depending on the AllRoutes option selected, the routing table can include the following data, which determines how a packet is routed:

- Port number

  This is the port associated with the attached network.
- NETnumber or "Default" label (which indicates a default route)
- Gateway address

  This is the IPX address of the gateway to which a router must send the packet before the packet can be routed to the destination. For more information on the gateway address, refer to "Adding and Deleting Static Routes" on page 13-17.
- Number of hops between router and destination

  The number of hops is equal to the number of gateways traversed.
- Costs associated with the route
- Status
- TTL
- Source

For each destination address, the router can support up to four routes (that is, four gateways). These routes, either learned or configured, are stored in the routing table. For information on how the router makes the routing decision, refer to "Routing Selection" on page 13-37.

Service information is maintained in a server table. To display the contents of the routing table, enter:

```
SHow -IPX AllRoutes
```

The following display appears:

```
-------------------------- IPX Routing Table --------------------
00000001   5   0000000C   7   00000032   3   00000045  7  00022222  6
0002ED49   9   00034562   9   0004065B   7   00044C34  9  000464FB  9
00049001   10  00049003   10  0004AD0D4  9   0004CFEC  8  0000502E6 13
000502E8   12  00053376   11  0005419    8   00054669  5  00055A1E  8
```

```
SHow -IPX AllRoutes Long
```

The following display appears:

```
----------------------- IPX Routing Table ----------------------
NETnumber Gateway         Hops     Cost     Status   TTL      Source
00000001                  5        6        UP       240      RIP
 &DDDDD200%080002A078DB
0000000C                  7        31       UP       240      RIP
 &DDDDD200%080002A078DB
00000032                  3        4        UP       240      RIP
 &DDDDD200%080002A078DB
00000045                  7        9        UP       240      RIP
 &DDDDD200%080002A078DB
00022222                  6        7        UP       240      RIP
&DDDDD200%080002A078DB
```

If you have a large routing table, you can specify a network number to verify its reachability using:

```
SHow -IPX AllRoutes <NETnumber>
```

Default Routes

When a router needs to route a packet destined for an address for which there are no entries in the routing table, it uses the default route if one exists. The network number &FFFFFFFE is reserved and represents the default route.

Default routes are important in building large, enterprise-wide networks. They allow an organization to perform route filtering at a border router and substitute the default routes with a single default route advertisement. A default route is useful over dial-on-demand lines, and can also be used as a backup route when the primary path is not available.

Effect on NRIP

NRIP recognizes and accepts the default route in NRIP advertisements received from other routers, enters it in the routing table, and propagates it if necessary. When forwarding IPX packets, an NRIP router forwards all unknown destination packets toward the default route.

The default route implementation for NRIP is not supported in software versions prior to version 8.2. All routers must be upgraded to software version 8.2 or later so that all NRIP routers recognize &FFFFFFFE as the default route and forward packets for unknown destinations toward it.

Effect on NLSP

An NLSP router can learn a default route in two ways:

- If there is an attached Level 2 NLSP router present, this router is considered the default route.
- Learning of network &FFFFFFFE from NRIP advertisements. This network number is imported and advertised to all other NLSP routers.

Forwarding unknown destination packets to the Level 2 router has higher precedence than forwarding an imported NRIP route. If there are no attached Level 2 routers, an NLSP Level 1 router forwards unknown destination packets toward the NRIP default route. If neither an attached Level 2 router nor an imported NRIP route is available, the NLSP Level 1 router drops the unknown destination packet.

Effect on SAP

The configuration of a default route has no effect on SAP advertisements, which list the network addresses of the available services. If the address is unreachable according to either an NRIP or NLSP update, the advertisement is dropped. This behavior is unchanged by the implementation of the default route.

Routing Selection

The IPX router selects the most efficient path for information. The most efficient path is the path that takes the least time to reach a destination. The amount of time needed to reach a destination is not configurable; it is based on the type of interface your router uses. The faster the line your router uses, the less time it will take for a packet to reach its destination. For example, an Ethernet (10 Mbps) is faster than a T1 (1.54 Mbps) serial line; it takes less time for a packet to reach its destination via an Ethernet than a T1 serial line.

You can affect the amount of time it takes a packet to traverse a serial line by using a faster line and changing the baud rate using the -PATH BAud parameter. This method of affecting the time a packet takes to traverse a serial line is effective only if the clock source for the serial line uses the internal on-board clock oscillator (TestMode value of the -PATH CLock parameter). When two paths require the same amount of time for a packet to traverse (same cost delay), the router will select the path with the lowest hop count. The router selects the path learned first if they have the same hop count.

Learning Routes and Service Information

To report route changes to its neighbors and learn about other services that are available on the network, the router or server (file server, printer, etc.) sends NRIP and SAP updates, respectively. In a large IPX environment, these update packets create the major network overhead. The frequency of the updates depends on the settings of the UpdateTime and CONTrol parameters as follows:

Periodic updates	By default, the router sends both NRIP and SAP updates at initialization and every 60 seconds (the default value of the UpdateTime parameter). When topology changes occur, updates are sent because Trigger is enabled by default. For details, refer to "Controlling NRIP and SAP Updates" on page 13-14 and "Controlling Route and Service Aging" on page 13-15.
Nonperiodic (incremental) updates	The router sends NRIP and SAP updates only when topology changes occur. Incremental NRIP and SAP updates are enabled by the NoPEriodic values of the -NRIP and -SAP CONTrol parameters. The -SAP CONTrol NoPEriodic value is the default setting on WAN interfaces. Nonperiodic is the preferred method on a WAN because it uses less bandwidth. For details, refer to "Controlling NRIP and SAP Updates" on page 13-14.

On LAN interfaces, the IPX router generates regular NRIP and SAP updates every 60 seconds. On slow WAN links, these NRIP and SAP updates can take up the bulk of the network traffic. In order to minimize network overhead, the router pays special attention to NRIP and SAP updates on WAN interfaces. By using the nonperiodic (incremental) update mechanism (enabled if the -SAP CONTrol parameter is set to NoPEriodic), the router does not send any NRIP or SAP updates over WAN interfaces except those containing new information after the system is initialized. NoPEriodic updates can substantially reduce network overhead over WAN links and can also be used on LAN interfaces if the NetWare servers on that network also support nonperiodic updates. All routers and servers on the same network should use the same update mechanism (periodic or nonperiodic).

You can also control if and how the router advertises routes to a neighbor from which it learned the same route. For details, refer to "Enabling and Disabling Dynamic Learning and NRIP Updates" on page 13-13, "Enabling Triggered NRIP Updates" on page 13-14, and "Using Poison Reverse or No Poison Reverse" on page 13-14. Another solution to routing overload is to use NLSP.

Regular route update packets contain the following types of information:

- The networks it can reach
- The number of hops and the amount of time associated with each network it can reach

For information on routing table entries, refer to "Routing Tables" on page 13-35.

Regular SAP updates packets contain the following types of information:

- Server type
- Name of the server
- Network address of the server
- Number of hops associated with each server

For information on service table entries, refer to the next section.

Server Tables

Server information is maintained in a server table. To display the contents of the server table, enter:

```
SHow -IPX AllServers
```

or

```
SHow -IPX AllServers Long
```

Adding "Long" to the command displays gateway information along with the server table contents.

If you have a large server table, you can specify a server name to display single server information using the SHow -IPX AllServers "<string>" syntax. An entry in the server table times out in the same way as a routing table entry (refer to "Controlling Route and Service Aging" on page 13-15 for details). When a server becomes unreachable, an update packet with this information is sent out immediately (refer to "Controlling NRIP and SAP Advertisements" on page 13-13 for details).

Network Reachability

When dynamic learning of routes is enabled, a router learns new routes from RIP update packets broadcast by its neighbors. The following are considered *reachable* when a router broadcasts its RIP update packets:

- Directly connected networks
- Static routes
- Dynamic routes learned through RIP that are currently in the routing table (that is, dynamic routes that have not timed out)

Solving the Slow Convergence Problem with Split Horizon

All routers need to learn of new routes and discard obsolete routes immediately. That is, the contents of their respective routing tables converge rapidly so that all routing tables always contain correct information. An undesirable side effect of NRIP is the possibility that the time during which the unreachable network is thought to be reachable is prolonged. One solution to this problem of slow convergence is called *split horizon*.

The following explanation describes split horizon for NRIP, but also applies to SAP.

The 3Com IPX router offers two methods for achieving split horizon: split horizon per network number and split horizon per neighbor, also known as next-hop split horizon. In a WAN environment, next-hop split horizon eliminates the need for a fully meshed network. With next-hop split horizon, the router learning of new routes records the IPX Internet address (network number and host address) of the advertising router and applies the split horizon algorithm per neighbor. Connectivity between different remote offices in a nonmeshed WAN topology can be maintained with next-hop split horizon while split horizon per network always expects a fully meshed topology.

Figure 13-14 shows a nonmeshed network on which router R is the root router and routers A, B, and C are remote routers that are configured as neighbors on router R. (This example applies to Frame Relay, ATM, and X.25 networks.) When both advertise and receive neighbor policies are disabled, split horizon per network takes effect. In this case, Router R excludes from its RIP updates on network &3333 all routes (&2222, &3333, &4444, and &5555) learned from

network &3333 if you select the NoPOison option of the -NRIP CONTrol parameter. If you select the POison option, router R includes routes but sets their hop count to 0xFFFF.

By applying next-hop split horizon, refer to "How the IPX Router Works" on page 13-33 for information about next-hop split horizon, router R does not advertise network &2222 to router A, because it learned of &2222 from router A (identified by router A's IPX address) or include it, but set its hop count to 0xFFFF depending on the POison/NoPOison option. For the same reason, router R does not advertise network &4444 to router B, nor does it advertise &5555 to router C, because it learned of those networks from those routers.

On Frame Relay, ATM, or X.25 networks, you must configure the host-to-media address mappings (ADDRess parameter). On Frame Relay networks, the bridge/router performs automatic DLCI learning and automatic host-to-DLCI address learning based on incoming IPX packets. Manually configure the host-to-DLCI address mapping because incoming IPX packets are not always guaranteed.

The host-to-media mappings (either configured or automatically learned) are used for transmitting NRIP and SAP advertisements. For NLSP, the host-to-media mappings are used for establishing adjacencies. The mapping information is useful regardless if the topology is full- or partially meshed.

LAN Networks On a LAN, you do not need to configure neighbors, but if neighbor policies are enabled and neighbors are configured, NRIP unicasts the updates to each neighbor. If neighbor policies are disabled, NRIP broadcasts the updates over the LAN.

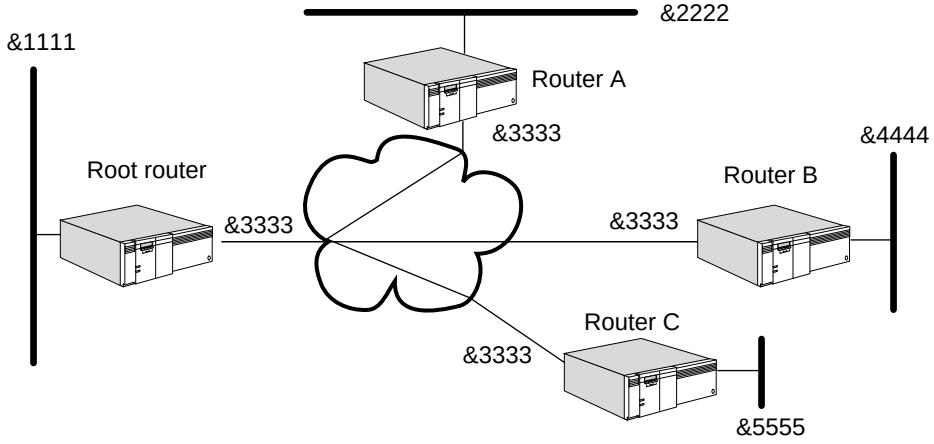


Figure 13-14 Route Advertisement Over Nonmeshed Frame Relay Network

No additional configuration is necessary to use the next-hop split horizon feature. It is automatically configured when neighbors are configured.

Solving the Slow Convergence Problem with Poison Reverse

Poison reverse or no poison reverse is configurable via the POison and NoPOison values for the -NRIP CONTrol parameter.

If poison reverse is enabled, the router advertises all routes to all neighbors, but when advertising a route to a neighbor that has advertised the same route, the router sets the hop count to infinity (0xFFFF) to prevent the recipient from adding the route to its routing table. Poison reverse speeds convergence but adds to network overhead.

If poison reverse is disabled, the router omits routes learned from one neighbor from NRIP updates sent to that neighbor. No poison reverse has the advantage of minimizing network overhead in large network configurations at the expense of slower convergence.

Route, Service, and Neighbor Policies

Route policies can be used to limit the view of the IPX Internet as seen from a specific segment, suppress reachability to selected networks in the Internet from specific segments, and provide security or segment isolation. Route policies also allow control of the propagation of routes to areas of the Internet where these routes are not needed, with the effect of controlling the sizes of the routing tables.

Route policy applies to the following events:

- NRIP updates received from other routers, called receive policy for routes.
- NRIP updates sent by the router, called advertise policy for routes. The NRIP updates are broadcast at regular intervals or whenever there are changes to the routing table.
- NRIP responses sent by the router whenever a NRIP request is received from a specific IPX host. The advertise policy can also be used to answer NRIP requests from a specific IPX host.

Service policies can be used to limit access to service from specific segments in the Internet, provide security or access-control, and reduce overhead by not advertising unnecessary resources. For example, access to a print server can be restricted to the segment where that printer's designated users are located, and the print service on that server is not advertised to the rest of the IPX Internet. Similar to route policies, the size of the service-related tables can be controlled by advertising only those services that need to be made available.

Service policy applies to the following events:

- SAP updates received from other routers, called receive policy for services.
- SAP updates sent by the router, called advertise policy for services. SAP updates are broadcast at regular intervals or whenever there are changes to the SAP table.
- SAP responses sent by the router whenever a SAP request is received from a specified IPX host. The advertise policy can also be used to answer SAP requests from a specified IPX host.

Neighbor policies are used to ensure that the router accepts routing information from and sends routing information to routers that are designated as neighbors.

Neighbor policy applies to the following conditions:

- The source or originator of NRIP and SAP updates. The neighbor is identified by the MAC address of the originator. The neighbor identification restricts information received.
- The destination of IPX hosts identified by the IPX network number and its MAC address. The neighbor identification selectively sends NRIP and SAP updates when responding to NRIP requests or SAP queries. If dynamic neighbors are enabled, the NRIP and SAP updates and responses are sent to all known neighbors.

Neighbor policies affect NRIP and SAP updates received from neighboring routers, regular and triggered NRIP and SAP updates sent to neighboring routers, and NRIP and SAP responses sent because of specific queries made by a client. If NRIP and SAP responses are sent because of a query by a client and the requesting client is not in the neighbor list that the router uses for sending NRIP and SAP updates, then no response is issued.

Policy Control

You can control route, service and neighbor policies as follows:

- You can disable policies during network operations.

  When a policy is disabled, the configured items corresponding to that policy are retained but are not used. Disabling policies at runtime is done through the PolicyControl parameter.

- You can configure the router to override the policies when responding to specific route and service requests using the PolicyControl parameter.

  That is, the policies are used for regular updates and triggered updates that are sent by the router during normal operation, but regular updates and triggered updates are overridden when the router responds to NRIP and SAP requests. The response to NRIP and SAP requests are sent directly to the requestor.

- You can configure the router to derive the routes being advertised on any specific interface from the configured service policies for that interface.

  Route advertisement decisions can be made using the service policy list. When service advertisement policies are configured and enabled, while route advertisement policy is enabled, but no route policies are explicitly configured, then the router policies are derived from the service policies. That is, if a service is identified on a network for inclusion in the SAP advertisement, then the network is also included in the NRIP advertisement.

- You can configure policy lists (lists of routes that are filtered out of NRIP updates received on a specified interface) as inclusion or normal policies, or exclusion or inverse policies.

Inclusion policies specify those items in the lists for inclusion in the NRIP updates and all other list items are excluded or filtered. Exclusion policy specifies the items for exclusion or filtering and all other items in the list are included in the NRIP updates. Policy lists can be applied to all parameters except AdvToNeighbor by prefixing the policy items with the tilde (~) character which indicates excluded list items.

Route Receive Policy

You can use the route receive policy to restrict the routes accepted from NRIP updates received on a specified port before the update is processed.

To restrict the routes that are accepted from NRIP broadcasts follow these guidelines:

- Use the ReceivePolicy parameter to identify the networks or routes that you want to include or exclude from the router's routing table when they are received in a NRIP update on the interface specified.

  Routes are identified by network number. Network number ranges can be specified to include or restrict a group of networks in the ReceivePolicy parameter.

- Use the ReceivePolicy attribute of the PolicyControl parameter to enable route receive policy.

If the PolicyControl attribute ReceivePolicy is set with no route receive policies configured, the router will not accept any routes that are being advertised to it by other routers on the specified interface.

Route Advertisement Policy

To restrict the routes that are advertised on a specified interface through regular and triggered updates, and those that are sent in NRIP responses to specific NRIP requests, follow these guidelines:

- Use the AdvertisePolicy parameter to identify the networks or routers that must be included in or excluded from NRIP updates or NRIP response broadcast from the specified interface.

  Routes are identified by network number. Network number ranges can be specified to include or restrict groups of networks in the AdvertisePolicy parameter.

- Use the AdvPolicy attribute of the PolicyControl parameter to enable route advertise policy.

  To restrict the routes advertised on a specified interface through regular and triggered updates, without causing restriction of any routes that are otherwise included in NRIP responses to specific NRIP requests, follow these guidelines:

- Use the AdvertisePolicy parameter to identify the networks or routes that must be included in or excluded from regular and triggered NRIP updates that are sent out the specified interface.

  Routes are identified by network number. Network number ranges can be specified to include or restrict groups of networks in the AdvertisePolicy parameter.

- Use the PolicyControl parameter to enable route advertise filtering by setting the attribute AdvPolicy.

- Use the PolicyControl parameter to enable the policy override option for NRIP responses by setting the PolicyOverride attribute.

  The PolicyOverride option applies to both NRIP responses and to service queries. To determine the routes that you want to include in regular and

triggered updates and responses to specific NRIP requests from the service policies that are configured for a specified interface, follow these guidelines:

- Use the AdvertisePolicy parameter to identify the services that are required for inclusion or exclusion from SAP updates and responses.

  Routes are identified by network number. Network number ranges can be specified to include or restrict groups of networks in the AdvertisePolicy parameter.

- Use the PolicyControl parameter to activate the service policies by setting the AdvPolicy attribute.
- Use the PolicyControl parameter to enable the route advertisement policy by setting the AdvPolicy attribute.

This method of determining the route policies from the service policies works only when the service advertisement policy is enabled. If the PolicyControl attribute AdvPolicy is set, no route advertise policies are configured, and there are no effective service advertise policies, then the router will not advertise any routes that are in its routing table to other routers on the specified interface.

Service Receive Policy

To restrict services from being accepted from SAP updates received on a specific port before the update is processed, follow these guidelines:

- Use the ReceivePolicy parameter to identify the services that are received in a SAP update on the specified interface that you want included in or excluded from the router's routing table.

  A service is identified by the network where the service is located, the host's MAC address, or the name of the server where the service and service type are located. Network number ranges and wildcards for network numbers, server host address or name and service types can be used to group services in the ReceivePolicy parameter.

- Use the PolicyControl parameter to enable service receive policy by setting the RcvPolicy attribute. If the PolicyControl attribute RcvPolicy is set, and there are no service receive policies configured, then the router will not accept any services that are being advertised to it by other routers on the specified interface.

Service Advertisement Policy

To restrict the services that are advertised from a specified interface through regular and triggered updates and those that are sent in SAP responses to specific SAP requests, follow these guidelines:

- Use the AdvertisePolicy parameter to identify the services you want included in or excluded from SAP updates or SAP responses sent out of the specified interface.

  A service is identified by the network where the service is located, the host's MAC address, or the name of the server where the service and service type are located. Network number ranges and wildcards for network numbers, server host address or name and service types can be used to group services in the ReceivePolicy parameter.

- Use the PolicyControl parameter to enable the service advertise policy by setting the AdvPolicy attribute.

  To restrict the services that are advertised from a specified interface through regular and triggered updates, but not restricting any services that are included in SAP responses to specific SAP requests, follow these guidelines:

- Use the AdvertisePolicy parameter to identify the services that must be included in or excluded from the regular and triggered SAP updates broadcast from the specified interface.

  A service is identified by the network where the service is located, the host's MAC address, or the name of the server where the service and service type are located. Network number ranges and wildcards for network numbers, server host address or name and service types can be used to group services in the ReceivePolicy parameter.

- Use the PolicyControl parameter to enable service advertise filtering by setting the AdvPolicy attribute.

- Use the PolicyControl parameter to enable the policy override option for SAP responses by setting the PolicyOverride attribute.

If the PolicyControl attribute AdvPolicy is set, and there are no service advertise policies configured, then the router will not advertise any services that are in its SAP table to other routers on the specified interface.

Neighbor Policy

To restrict the number and identity of routers that the listening router should accept NRIP and SAP updates from, follow these guidelines:

- Use the RcvFromNeighbor parameter to identify the routers.

  Neighbors are identified by their host's MAC address in the RcvFromNeighbor parameter.

- Use the PolicyControl parameter to enable the neighbor policy for received NRIP and SAP updates by setting the RcvFromNbr attribute.

If the PolicyControl attribute RcvFromNbr is set, and a list of neighbors to receive from has not been configured, then none of the NRIP and SAP updates received are accepted.

To restrict the number and identify the neighbors the sending router can broadcast NRIP and SAP updates to, and those the router can accept NRIP requests or SAP queries from, follow these guidelines:

- Use the AdvToNeighbor parameter to identify the neighbors.

  The router can be configured to send a unicast copy of the NRIP and SAP update. Each neighbor is identified by the IPX network number and its MAC address in the AdvToNeighbor parameter.

- Use the PolicyControl parameter to enable the neighbor policy for advertisement of NRIP and SAP updates and for responses to NRIP requests and SAP queries by setting the AdvToNbr attribute.

  If the PolicyControl attribute AdvToNbr is set, and dynamic neighbors are enabled, all NRIP and SAP updates are sent to all know neighbors individually.

If the PolicyControl attribute AdvToNbr is set, and no neighbors are identified, then NRIP and SAP updates will not be broadcast from the specified interface and there will be no response to any requests or queries received on that interface.

Novell Service Types

When setting IPX NLSP, NRIP, and SAP policies, you may need information for Novell Service Types available on file servers. Table 13-4 lists the Novell Service Types and the object type (in hex) that should be used.

Table 13-4 Novell Service Descriptions

Description	Object Type (hex)
User	0x0001
User Group	0x0002
Print Queue	0x0003
File Server	0x0004
Job Server	0x0005
Gateway	0x0006
Print Server	0x0007
Archive Queue	0x0008
Archive Server	0x0009
Job Queue	0x00A
Administration	0x00B
NAS SNA Gateway	0x0021
NACS	0x0023
Remote Bridge Server	0x0024
Bridge Server	0x0026
TCP/IP Gateway	0x0027
Gateway	0x0029
Time Synchronization Server	0x002D
Archive Server SAP	0x002E
Advertising Print Server	0x0047
BTrieve VAP 5.0	0x004B
SQL VAP	0x004C
XTREE Network Version	0x004D
BTrieve VAP 4.11	0x0050
Print Queue User	0x0053
WANcopy Utility	0x0072
TES - NetWare for VMS	0x007A
NetWare Access Server	0x0098
Portable NetWare	0x0107
NetWare 386	0x0107
Communications Executive	0x0130
NSS Domain	0x0133
NetWare 386 Print Queue	0x0137
NetWare 386 SAA Server	0x0304
Wildcard	0xFFFF

NLSP Routing

The NLSP routing protocol was developed by Novell to provide network layer connectivity in IPX networks. NLSP provides faster convergence and less overhead than other routing protocols by using a link-state-based routing algorithm. NLSP is designed as a router-to-router protocol. Clients and servers are not expected to participate in the NLSP packet exchange and continue to expect RIP and SAP updates. NSLP, RIP, and SAP coexist on the same internetwork: NLSP manages route and server information exchanges between routers and RIP and SAP advertise route and server information to end systems. Figure 13-15 shows the NLSP coexistence with RIP and SAP.

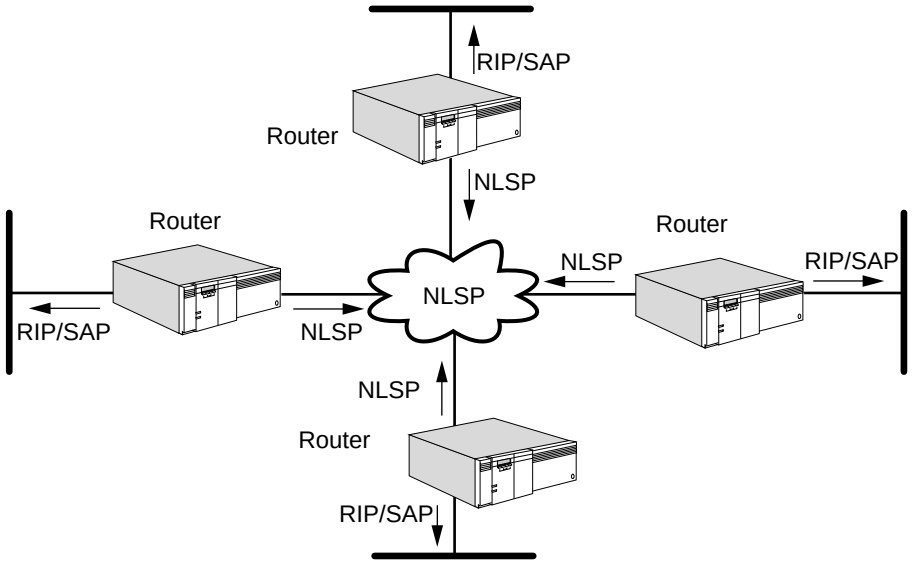


Figure 13-15 NLSP and RIP/SAP Coexistence

Hierarchical Routing

NLSP provides a hierarchical network topology that reduces overhead and allows the internetwork to scale because the NLSP routing overhead is confined to a particular area. Routing domains provide administrative boundaries in the internetwork.

In the NLSP hierarchical topology, networks are organized into areas, and areas are grouped into multiple routing domains as shown in Figure 13-16. A routing domain is a stand-alone administrative entity (such as a company, a university, or an agency). Routing domains are interconnected by Level 3 routers. Each routing domain can be further subdivided into multiple areas. An area can be a department, a building, or a group of highly connected and functionally related workstations or servers. An area can be as small as a single LAN, or as large as several hundred networks and hundreds of routers. areas are interconnected by Level 2 routers. All routers within an area are Level 1 routers.

The current implementation for NLSP operates within an area only.

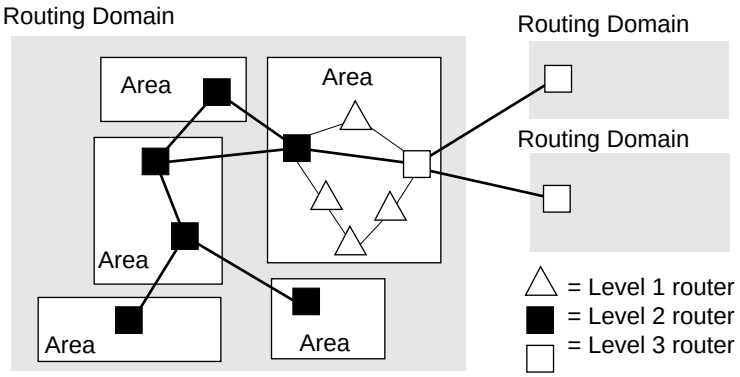


Figure 13-16 NLSP Hierarchical Routing

A single router is the minimum area that can be formed. The maximum area can contain hundreds of routers and networks, however, because memory overhead on a router is proportional to the size of its home area, the real size of an area will be conservative.

All routers belonging to the same area must be directly interconnected through physical paths. Any router must be able to reach any other router in the same area through intra-area routes by going through other routers belonging to the same area.

Routers in an NLSP environment form adjacencies with each other, and exchange information with adjacent routers about the status of their connected networks through link state packets (LSPs). The LSPs are used to build link-state databases, which are synchronized between adjacent routers to ensure accuracy. The LSPs are flooded throughout the area and all routers maintain identical detailed information about the topology of that area. If a network in that area changes status, an LSP are flooded quickly throughout the area to record the change.

Area Addressing

Each router must identify one to three area addresses, which are communicated to adjacent routers in the LSP packets and are also reflected in the network number portion of the IPX address. The IPX network number is a 32-bit integer, of which some bits identify the area and others identify the network within that area. The identification of both the value and length of the area address is configured in the -NLSP areaAddress parameter using:

```
ADD -NLSP AreaAddress <net> <mask>
```

Each of the <net> and <mask> fields are 32-bit integers, the <net> field representing the value of the area address and the <mask> field representing how many of the 32 bits in the IPX network number are used to identify the area. For example:

```
ADD -NLSP AreaAddress 12345600 FFFFFF00
```

A router can be configured with up to three area addresses, in which case a single area still exists but has three possible identifiers. A maximum of three area addresses are allowed in any area. If there exists more than three addresses within an area, the higher area addresses are dropped.

IPX routing Terms

spoof — A process that allows the bridge/router to respond to incoming NCP KeepAliveRequest or SPX1 watchdog packets that are to be routed over a DOD line, by sending a packet to the originating server of the request on behalf of the intended client. Spoofing occurs only when the DOD path is down to prevent the DOD path from constantly being brought up and down due to the transmission of packets from the server.

14 APPLETALK ROUTING

This chapter describes how to configure, customize, and troubleshoot a basic AppleTalk router.

For conceptual information, refer to "How the AppleTalk Router Works" on page 14-16.

Setting Up a Basic AppleTalk Router

This section describes how to set up a basic AppleTalk router. After you perform these minimum configuration steps to configure your AppleTalk router, you can use the default values of other parameters, or you can further customize the AppleTalk router as described in "Customizing the AppleTalk Router" on page 14-7.

Prerequisites

This section assumes that you have logged on to the system with Network Manager privilege and set up the ports and paths of your bridge/router according to Chapter 1.

Before setting up an AppleTalk router, create a router plan. The router plan will help you determine how the AppleTalk internetwork will look and which router will "seed" each network. Remember that each internetwork is unique. There are no absolute rules that govern placement of seed routers in an internetwork.

Creating a Router Plan

To create a router plan, follow these steps:

1. Make a diagram of your proposed AppleTalk internetwork.

   Include the physical network layout and connecting points (for example, routers and bridges) in your diagram. For an example of a diagram, see Figure 14-4.

2. For each network, determine the following information:

   - The number of AppleTalk devices (for example, workstations, servers, and printers) present and projected.
   - The quantity of network numbers sufficient to satisfy capacity requirements (up to *n* x 253 devices can be supported, where *n* is the number of network numbers in the range). 3Com recommends leaving gaps between network number ranges in order to accommodate network growth.
   - The number of zones and names needed and which devices will be in each zone for those networks with more than one. You will also need to identify which zone will be the default zone of the network.

3 Create a table of your router seeding plan, indicating which router will seed each network.

For definitions of seed and nonseed routers, refer to "Related Information" on page 14-4.

When you complete this table, you should have a record of all network number ranges in use, all zones in use, and which AppleTalk routers define zones and network numbers for each connected network.

In the simplest router seeding plan, you may pick one bridge/router per physical network as the seed router for that network. A single bridge/router can seed multiple networks (up to the maximum number of ports available).

An alternative plan is to set up multiple seed routers that supply identical information for a network. If the seed router hardware stops functioning and all seed routers have to be rebooted, you will not have to configure a new router to replace the disabled router at an inconvenient time. Another router with redundant seeding information can fill the role of seed router immediately. For more information, refer to "Setting Up Multiple Seed Routers" on page 14-7.

4 For maintenance purposes, you should create a database from your router seeding plan. Include the following information:

- Router location

  Router location includes physical location and router name. The router name can be common to all names of ports (as specified by the RouterName parameter) on the router.

- Router type and version
- Networks connected to the router with the following information for each:
  - Cabling identification
  - Port type (EtherTalk, TokenTalk, LocalTalk, Fiber Distributed Data Interface (FDDI), Point-to-Point Protocol (PPP), Phone Line Gateway (PLG), X.25, Switched Multimegabit Data Service (SMDS), or Frame Relay)
  - Seed information, if configured: network range, zone list, and default zone
  - Data link address for each router port (media access control (MAC) address, X.25 Data Terminal Equipment (DTE), Frame Relay, Data Link Connection Identifier (DLCI), SMDS individual and group address)

Procedures

This section provides information on configuring local and wide area networks.

Configuring for Local Area Networks

This section provides information on how to configure AppleTalk routers on Ethernet, token ring, and FDDI networks.

Your router plan will help you determine which routers need to be configured as seed routers. All other routers not configured as seed routers must be configured as nonseed routers. This section provides procedures on how to set up your router as a seed or nonseed router.

To set up a seed router, follow these steps:

1. Specify the range of network numbers that can be used on the cable to which the router port is attached using:

   ```
   SETDefault !<port> -AppleTalk NetRange = <network-range>
   ```

2. If most end nodes on a cable will be in a single zone, use that zone as the default. Specify the default zone name for the network attached to a port using:

   ```
   SETDefault !<port> -AppleTalk DefaultZone = “<zone-string>”
     (1–32 char)
   ```

3. Specify additional zone names for nodes to be placed in different zones using:

   ```
   ADD !<port> -AppleTalk ZONe “<zone-string>” (1–32 char)
   ```

   All seed routers must have the same net range, zone list, and default zone.

4. Enable AppleTalk routing on the port using:

   ```
   SETDefault !<port> -AppleTalk CONTrol = (ROute, AppleTalk,
     SeedingAllowed)
   ```

This step must be performed after network number range and zone information are configured.

To set up a nonseed router, enable AppleTalk routing and disable seed router capability on a particular port using:

```
SETDefault !<port> -AppleTalk CONTrol = (ROute, AppleTalk,
  NoSeedingAllowed)
```

Repeat this step for other ports if appropriate.

For complete information on all parameters used in these procedures, refer to Chapter 4 in the *Reference for NETBuilder Family Software*.

Configuring for Wide Area Networks

Routing AppleTalk over Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies.

If you plan to route AppleTalk over Frame Relay, ATM DXI, or X.25 in a partially meshed or nonmeshed topology, you must make certain that static AppleTalk address mappings are defined. Defining these mappings enables the next-hop split horizon feature. For complete information on configuring AppleTalk routing over Frame Relay, ATM DXI, or X.25, including a discussion on fully meshed, partially meshed, and nonmeshed topologies and next-hop split horizon, refer to Chapter 42, Chapter 43, and Chapter 45.

Routing AppleTalk over SMDS is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your AppleTalk router to perform routing over SMDS, refer to Chapter 44.

PPP and PLG links should be configured as non-AppleTalk data links. No static configuration is required. For more information, refer to “Setting Up AppleTalk Routing over a Non-AppleTalk Data Link” on page 14-8 and to Chapter 34.

For information on wide area networking using Integrated Services Digital Network (ISDN), refer to Chapter 35.

Related Information

AppleTalk routing involves the following two types of routers:

- Seed routers

  These routers serve as initial information and query points for other routers and end systems on AppleTalk networks. Each network cable, or set of bridged segments that are to be treated as a single AppleTalk network, must have at least one seed router. Seed routers require more configuration than nonseed routers and should be the first AppleTalk devices booted on a network. It is suggested that multiple routers be configured with identical seed information for redundancy.

- Nonseed routers

  These routers require a minimum of configuration steps. Nonseed routers connected to AppleTalk networks must obtain information such as network numbers and zone lists from another router acting as a seed router on a connected network. The specific router that provides information to a new nonseed router is usually the first discovered by the new router.

3Com routers can also be used to route AppleTalk across non-AppleTalk backbone networks or point-to-point wide area links. These routers do not need to share seed information; they only share routing and zone information about the AppleTalk networks of which they are aware. Refer to "Setting Up AppleTalk Routing over a Non-AppleTalk Data Link" on page 14-8.

After enabling routing on a port or when booting the bridge/router, a SHow command executed before the AppleTalk router has completed the initialization phase may display parameter values that imply that the router is still configured to NoRoute. The SHow -AppleTalk DIAGnostics command gives you the current state of each port.

A router can be a seed router on all ports; however, a router does not have to be a seed router for all the ports over which AppleTalk is routed. For example, a router with connections to networks over three ports may serve as a seed router for two of these and not as a seed router for the third.

During configuration, you must decide whether or not a port will be seeding. If it is, you must configure seeding information. If it is not to be a seed router, it is assumed that the connected network will be seeded by another AppleTalk router attached to the same network.

A seed router port must be configured to contain the following information:

- Network number range (the NetRange parameter)
- A list of one or more AppleTalk zones (the ZONe parameter)
- The default zone for the network if more than one zone is configured (the DefaultZone parameter)

The CONTrol parameter options also control how seed information is used and provide inter-router seed information validation. For more information, refer to "Port Startup Operations" on page 14-21 and the description of the CONTrol parameter in Chapter 4 in *Reference for NETBuilder Family Software*.

Verifying the Configuration

To verify that the routers you configured are recognized by the network and are functional, follow these steps:

1 Check for possible problems using:

```
SHow !<port> -AppleTalk DIAGnostics
```

The router displays a variety of information, depending on conditions detected by the software. For a general description of information available through the DIAGnostics parameter display, refer to Chapter 4 in *Reference for NETBuilder Family Software.*

2 Check the routing table by entering:

```
SHow -AppleTalk AllRoutes
```

The routing table displays all the networks to which a router has access directly or indirectly. Make sure that all expected networks are listed. Check that the expected *next routers* to the networks listed appear in the routing table. You may need to refer to your planning documents to associate data link addresses with routers.

3 Display the mapping information between zone names and network numbers and between network numbers and zone names by entering:

```
SHow -AppleTalk ZoneNetMapping
SHow -AppleTalk NetZoneMapping
```

For the mapping information between zone name and network number, the router displays a list of all zones and their associated networks on the AppleTalk internetwork that are known to the router. Make sure all expected zones are present. It usually takes a minute or less to acquire network and zone information, but may take longer depending on the size of the AppleTalk internetwork.

For the mapping information between network number and zone name, the router displays a list of associated zones for each known network. Make sure that all zone lists are complete (check the display for messages.)

Check these displays for accuracy. If a discrepancy appears, you must check and adjust the zone lists for seed routers directly connected to the networks in question. Refer to "Changing a Zone List" on page 14-16.

4 Check the AppleTalk-specific configuration using:

```
SHow !<port> -AppleTalk CONFiguration
SHowDefault !<port> -AppleTalk CONFiguration
```

The SHow configuration command displays live values. The SHowDefault command displays the values you have configured.

To obtain seed router status for an interface, the network range and at least one zone need to be specified for the network zone list. If there are unexpected results, enter:

```
SHow -AT DIAGnostics
```

In addition to performing checking procedures, the AppleTalk router is also an AppleTalk echo protocol responder. Reachability can be checked from another NETBuilder AppleTalk router on the AppleTalk internetwork using the APING command. For more information, refer to Chapter 1 in Reference for NETBuilder Family Software.

Getting Statistics

To gather statistics, enter:

```
SHow -SYS STATistics -AppleTalk
```

For a sample display and an explanation of the display, refer to Appendix H.

You can collect statistics for a specific time period by using the -SYS SampleTime and -SYS STATistics parameters. For more information, refer to Chapter 58 in *Reference for NETBuilder Family Software*.

Troubleshooting the Configuration

If you are unable to make connections to nodes within a local area or nodes in other areas after setting up the router, review the following troubleshooting procedure. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier for assistance.

To troubleshoot your configuration, follow these steps:

1. Display diagnostic information stored by the router by using:

   ```
   SHow !<port> -AppleTalk DIAGnostics
   ```

   The router displays a variety of information, depending on conditions detected by the software. For a general description of what is available through the DIAGnostics parameter display, refer to Chapter 4 in *Reference for NETBuilder Family Software*.

2. If the diagnostic information displayed indicates that a port is down, enter:

   ```
   SHow -PORT CONFiguration
   SHow -PATH CONFiguration
   ```

3. Check the AppleTalk-specific configuration using:

   ```
   SHow !<port> -AppleTalk CONFiguration
   SHowDefault !<port> -AppleTalk CONFiguration
   ```

   The SHow configuration command displays live values. The SHowDefault command displays the values you have configured.

   Check that the displayed configuration is the correct one for this router.

4. Check for a misconfigured port owner using:

   ```
   SHow [!<port>] -PORT OWNer
   ```

5. Check whether the network you are trying to reach is in the AppleTalk routing table using:

   ```
   SHow !<port> -AppleTalk AllRoutes <network range>
   ```

   If the network you are trying to reach is in the routing table, a router that connects the network may not be passing packets because of filters that may have been set up; if the network you are trying to reach is not in the routing table, it is unreachable. From the table entries, or lack of table entries, you can determine which path is being used and in what direction you can continue to investigate.

6. Use the APING and ANameLookup commands to determine the connectivity to different router and end stations.

   You can determine where the connectivity is broken by how far you can see. Refer to your network planning documentation for the intended connectivity.

For a detailed description of the APING and ANameLookup commands, refer to Chapter 1 in *Reference for NETBuilder Family Software*.

Unless you have fully meshed Frame Relay or X.25 AppleTalk network topologies, the APING and ANameLookup commands may not work with router ports attached to these wide area network media. It is recommended that you use the APING command against AppleTalk local area network ports on these routers to determine reachability.

7 If your router has a serial line interface, check the transmit clock to see if it is correctly set using:

```
SHow !<path> -PATH CLock
```

8 Check that all cables on all routers in a specific path in the routing table are properly connected and that the routers are properly installed.

For instructions, refer to the installation guide provided with your bridge/router.

9 Check AppleTalk statistics by entering:

```
SHow -SYS STATistics -AppleTalk
```

For complete information on AppleTalk statistics, refer to Appendix H.

Customizing the AppleTalk Router

Most AppleTalk parameters are automatically configured to their default values. (With few exceptions, the only parameters that need to be configured to enable routing are discussed in "Setting Up a Basic AppleTalk Router" on page 14-1.) In some cases, you may want to change the default configuration.

This section is intended for those who want to go beyond the minimum configuration of a nonseed or seed router. It explains how to:

- Set up AppleTalk routing over a non-AppleTalk data link.
- Change the frequency at which a routing table propagates routes.
- Set up filters.
- Change a zone list for an AppleTalk network.

Not all available parameters are discussed in this section. For more information on all available parameters, refer to Chapter 4 in *Reference for NETBuilder Family Software*.

Setting Up Multiple Seed Routers

This section provides information on setting up multiple seed routers on a network.

Procedure

To install multiple seed routers on a network, refer to "Setting Up a Basic AppleTalk Router" on page 14-1.

Related Information

To provide redundancy in case of system crashes and power outages, you can install multiple seed routers on the same network.

When you install more than one 3Com AppleTalk router as a seed router for a particular network, all the routers should seed the same information configured

for that network. The first seed router that establishes itself (is started and goes active) on the network becomes the actual seed router. After one or more AppleTalk routers are started up, the seed information provided by the seed router can be supplied by any of the routers connected to a particular network.

To display any network number inconsistencies between routers, enter:

```
SHow -AppleTalk DIAGnostics
```

The first seed router that establishes itself on a network defines the values. The subsequent NETBuilder seed routers discovering the inconsistency can optionally, if the SeedCheck option is selected (default setting), disable the port connected to the network and note the condition that is displayed.

Different brands of AppleTalk routers handle conflicting seed information differently. For details of their operation, refer to their respective documentation.

Setting Up AppleTalk Routing over a Non-AppleTalk Data Link

To configure a local or wide area port of a router connected to a non-AppleTalk data link, follow these steps:

1. Enable AppleTalk routing over a non-AppleTalk network using:

   ```
   SETDefault !<port> -AppleTalk CONTrol = (ROute, NonAppleTalk)
   ```

2. Verify the configuration of each router port using:

   ```
   SHow !<port> -AppleTalk CONFiguration
   SHowDefault !<port> -AppleTalk CONFiguration
   ```

   The SHow configuration command displays live values. The SHowDefault command displays the values you have configured.

   To complete the configuration for PPP/PLG, Frame Relay, SMDS, or X.25 wide area ports, refer to Chapter 34, Chapter 42, Chapter 44, or Chapter 45.

Related Information

Where AppleTalk routing is supported, any data type such as Ethernet, FDDI, token ring, PPP, PLG, X.25, SMDS, or Frame Relay can be treated as a non-AppleTalk link, backbone, or "cloud." 3Com AppleTalk routers can communicate across these links, connecting the AppleTalk networks that exist as offshoots of the data link.

This feature is especially useful for configuring the point-to-point links (PPP, PLG) and cloud links (X.25, Frame Relay), where no AppleTalk end systems can reside. Although any of the remaining data links (Ethernet, token ring, FDDI, SMDS) can support AppleTalk end nodes, they may not support them in actual installations. They may operate as a backbone network, or only support non-AppleTalk network devices.

When AppleTalk end nodes are not supported, if you configure the links as non-AppleTalk, you do not need to configure seed information, which saves network range numbers and zone lists. Unwanted name lookup multicasts on the link are also eliminated (most commonly generated by using the Chooser interface on the Macintosh).

A disadvantage to configuring Frame Relay and X.25 ports when connected to non-AppleTalk networks is the work involved in moving configured neighboring router information to another port. If you move a serial interface to a different port, you need to define the neighbor information for the new port and delete the same information from the old port using the -AppleTalk ADDRess parameter. If you treat the port as connected to an AppleTalk network, you only need to define the network range on the new port and remove the same range from the old port. (To define and delete the network range, use the SETDefault -AppleTalk NetRange command.) The software automatically associates the configured neighbor information (for example, 20.30 @56) with the new port when it is activated.

Changing Frequency of Routing Table Route Propagation

This section provides information on how to change the frequency at which a routing table propagates routes.

Procedure

To change the frequency, follow these steps:

1. Change the frequency at which a router sends out routing information packets using:

   ```
   SETDefault -AppleTalk RouteUpdateTime = <seconds> (1–300)
   ```

2. Change the frequency at which routes in the routing table are verified using:

   ```
   SETDefault -AppleTalk RouteAgingTime = <seconds> (20–300)
   ```

Related Information

Every 10 seconds (the default setting of the RouteUpdateTime parameter), the router sends broadcast packets to its neighboring routers to report the following types of information:

- The networks it can reach
- The number of hops associated with each network it can reach

You can configure the RouteUpdateTime parameter to change the frequency at which the router sends out routing information packets.

When other AppleTalk routers that cannot change the time interval are present do not use a value other than the default of 10 seconds. The value of the RouteUpdateTime parameter and the frequency of AppleTalk routing table aging are related. Table aging is set through the RouteAgingTime parameter, which has a default of 20 seconds. If broadcasts are less frequent, but aging is left the same or reduced, increased table entry deletions and additions may occur, which can affect routing capability and increase table maintenance overhead.

Try to keep at least a 1-to-2 ratio between RouteUpdateTime and RouteAgingTime. However, increasing the value of both parameters increases the time for topological changes to propagate through the routers. Route update packets also are not reliably received and may be lost on a busy network. Their frequency should be enough to ensure reception on a busy network before other routers age out the routes. Decreasing the value of both parameters improves route propagation and route convergence to new paths, but at the expense of higher bandwidth utilization for route information exchange.

When a route is learned, it goes into the routing table. The router then sends a query asking for zone lists for the networks for which it does not have complete zone list information. Other routers pass back zone list information to the querying router. This occurs as information about other networks are propagated. A NETBuilder II router does not propagate information about route information for a network until it has complete zone list information associated with that network.

Setting Up Filters

The following types of filtering are available for restricting access to the AppleTalk internetwork through a specified port:

- Network number-based filtering
- Entity filtering

The use of both network and entity filters is only effective if there are no alternate, non-filtered routes to the filtered networks or services. The use of filtering also slows down the performance of your AppleTalk router.

The following sections describe each type of filtering. For more examples and details on using the parameters described in these sections, refer to Chapter 4 in *Reference for NETBuilder Family Software.*

Setting Up Network Number-Based Filtering

This section provides procedures on how to set up positive and negative network number-based filtering. A sample topology is also provided to illustrate each step of the procedures.

The following example is an application of network-number based filtering. In this example, three AppleTalk networks are interconnected through a backbone (networks 10–20, 30–40, 50–60 in Figure 14-1). These three networks are said to be in a "trusted" environment; that is, nodes on these networks can access resources on all three networks. A second network (90–100 in Figure 14-1) is said to be outside the trusted environment. Nodes on that network are permitted to access resources on network 50–60 (and vice versa) but are prevented from accessing resources on the other networks connected to the backbone (namely 10–20 and 30–40).

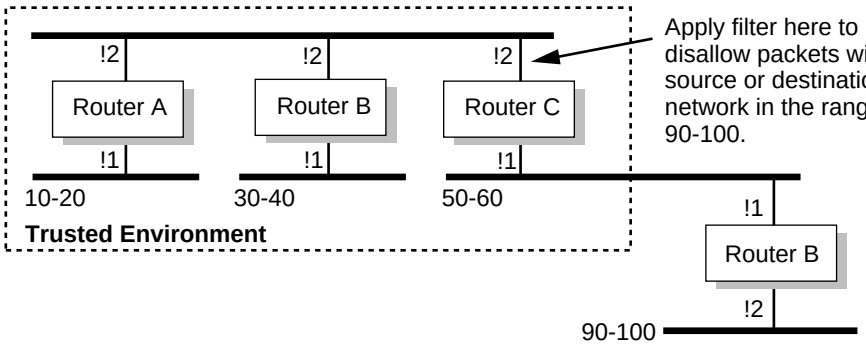


Figure 14-1 AppleTalk Network Filter Example

One way you can satisfy these requirements is with *positive filtering*. As shown in Figure 14-1, this filtering is implemented by applying network filtering of 90–100 on port 2 of router C. This filter stops the propagation of packets either

originating from a node or destined to a node with a network number in the range 90–100 beyond this interface. In other words, if a packet from a node on network 90 is received on port 1 of router C and is destined to a node on network 10, then it is not forwarded out of port 2 of router C. Similarly, if a packet is received from a node on network 10 on port 2 and is destined to a node on network 90, it is not forwarded out of port 1.

Setting Up Positive Filtering To set up a positive network filter, follow these steps. The sample topology described previously will be used to illustrate each step.

1 Enable network number filtering using:

```
SETDefault !<port> -AppleTalk CONTrol = NetFilter
```

2 Create a set of filter network ranges using:

```
ADD !<port> -AppleTalk NetFilter = <network range>
```

3 Specify that the newly created network filter range is to be used for positive filtering using:

```
SETDefault !<port> -AppleTalk NetFilterType = Positive
```

In the previous procedure on how to set up positive filtering, the filtered set of networks is included within the specific range of 90–100. You can achieve the same results with *negative filtering*, which is the application of filtering through exclusion. In this case, the filtered set of networks are all networks *not* in the range 10–60. You can apply this filter at the same point, that is, port 2 of router C.

Setting Up Negative Filtering The sample topology described above will be used to illustrate each step of the following procedure.

To set up negative filtering, follow these steps:

1 Enable network number filtering using:

```
SETDefault !<port> -AppleTalk CONTrol = NetFilter
```

2 Create a set of filter network ranges using:

```
ADD !<port> -AppleTalk NetFilter = <network range>
```

3 Specify that the newly created network filter range is to be used for negative filtering.

For example, to set the network filter range specified on port 2 of router C to positive, enter:

```
SETDefault !2 -AppleTalk NetFilterType = Negative
```

For complete information on each of the parameters used in this section, refer to Chapter 4 in *Reference for NETBuilder Family Software*.

Related Information Network filtering allows you to filter received packets on a per-port basis based on source and destination network numbers. The following criteria apply:

- Packets are filtered on receipt at a port based on a packet's final destination network.
- Packets are filtered on forwarding (transmission) out of a port based on the network from which the packet originated.

These criteria control the flow of packets between the various ports of a router. The following events also occur as a result of filtering:

- Networks are not included in Routing Table Maintentance Protocol (RTMP) routing updates out a port if their range is completely included in the set of filtered networks for the port.
- Zone information is suppressed from being sent out a port if all networks associated with a zone are in the set of filtered networks for the port and the zone is not associated with the directly connected network out the port.

The following types of network filtering are available:

- Positive network filters discard all packets destined to or originating from a set of network number ranges that you specify.
- Negative network filters discard all packets except for those destined to or originating from a set of network number ranges that you specify.

Setting Up Entity Filters

This section provides a procedure on how to set up entity filtering. A sample topology is provided to illustrate each step of the procedure. At the end of the procedure, an additional example of implementing entity filtering is provided.

In Figure 14-2, router A has three ports. Port 1 is connected to a network that contains two pools of resources, labeled POOL-A and POOL-B. These resources could be a collection of printers, file servers, communication servers, etc. Port 2 and port 3 are connected to two network segments that contain users who access the resources in POOL-A and POOL-B. The requirement in this example is to partition the pool of resources so that all users on the segment attached to port 2 can only access resources in POOL-A and all users on the segment attached to port 3 can only access resources in POOL-B. To simplify the filter specification, assume that all resources in POOL-A have object names with the prefix "POOL-A" and all resources in POOL B have object names with the prefix "POOL-B," for example, "POOL-A-LASERWRITER," "POOL-B-DBSERVER," etc.

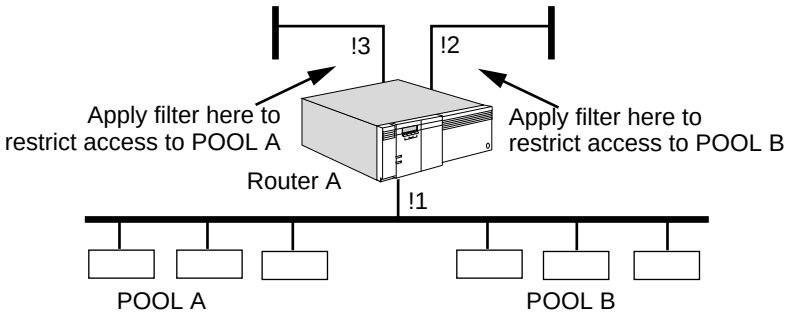


Figure 14-2 AppleTalk Entity Filter

As shown in Figure 14-2, the entity filters are applied at ports 2 and 3. At port 2, the filtered set will be all entities whose object names start with the pattern "POOL-B." At port 3, the filtered set will be all entities whose object names start with the pattern "POOL-A." The configuration of these filters is shown in the following procedure.

Procedure To set up your AppleTalk router to perform entity filtering, see Figure 14-2 and follow these steps :

1 Enable entity filtering.

In the topology shown in Figure 14-2, entity filtering should be enabled on ports 2 and 3. For example, to enable entity filtering on port 2, enter:

```
SETDefault !2 -AppleTalk CONTrol = EntityFilter
```

2 Create one or more entity filters.

Create entity filter specification "POOL-A~:=@=" and make it filter number 1 in the entity filter table by entering:

```
ADD -AppleTalk EntityFilter 1 "POOL-A~:=@="
```

Create entity filter specification "POOL-B~:=@=" and make it filter number 2 in the entity filter table by entering:

```
ADD -AppleTalk EntityFilter 2 "POOL-B~:=@="
```

3 Assign an entity filter to a particular port and specify whether it is a positive or negative filter.

To assign entity filter number 1 to port 3 and specify that it is a positive filter, enter:

```
ADD !3 -AppleTalk EntityFilterNum 1 Positive
```

The statistic Entity Filter Matches is present at the end of the AppleTalk statistics. It displays the number of NBP Request or Reply packets dropped because of a match against an active entity filter.

Assign entity filter number 2 to port 2 and specify that it is a positive filter by entering:

```
ADD !2 -AppleTalk EntityFilterNum 2 Positive
```

Example To create an entity filter that restricts access to a LaserWriter with the name "MktPrinter" in zone "Mkt," enter:

```
ADD -AppleTalk EntityFilter 1 "MktPrinter:LaserWriter@Mkt"
```

To define that the above entity filter is a positive filter that applies to port 2, enter:

```
ADD !2 -AppleTalk EntityFilterNum 1 Positive
```

For complete information on each of the parameters used in this section and more examples on how to create entity filters, refer to Chapter 4 in *Reference for NETBuilder Family Software*.

Related Information Entity filtering allows you to restrict access across a port to specific named network entities or sets of entities on an AppleTalk network. These resources can include file servers, printers, and communications servers. Access to network entities is based on entity name and (optionally) network number.

Entity filtering operates as a filter on name lookup requests and responses across a port. When a Macintosh user opens the Chooser interface and selects a service icon, name lookups are sent across the internetwork to all networks that are associated with the zone currently selected in the Chooser. Those services that meet the lookup criteria (in this case, those that have the same entity name type in the specified zone) send lookup response packets containing their entity name back to the source of the lookup. Entity filtering prevents responses from being returned by stopping the requests from continuing on or by intercepting the responses. It checks in both directions because wildcards are used in the requests, but not in the responses.

The configuration of entity filters is a two-step process. The first step is to configure filtering criteria by specifying the entity name and, optionally, a network number range qualifier. This information is configured through the EntityFilter parameter. The second step is to associate filtering criteria with a port in addition to the positive or negative filter type attribute. You can use the EntityFilterNum parameter to add an entity filter to a specified port and designate it as a positive or negative filter.

Setting Up Zone Advertisement Filtering

This section provides a procedure on how to set up zone advertisement filtering. A sample topology is provided to illustrate each step of the procedure.

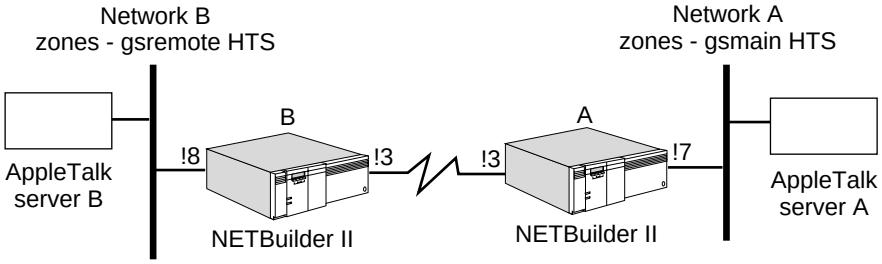


Figure 14-3 Setting Up Zone Advertisement Filtering

Zone advertisement filtering filters specific zones being returned in the ZoneList when a ZIP ZoneList request is received. A Zip ZoneList is created when:

- A station is connected to the AppleTalk network and it is acquiring the available zones.
- A chooser application is acquiring zones to access shared devices.
- An InterPool application is requiring zones to list all devices.

The zone advertisement filter is configured on specific ports, which allows zones to be hidden on some ports but advertised on others. Zones are configured using the EntityFilter parameter in the AppleTalk Service. Only zone-specific filters (“=:=@zone”) can be selected for a zone advertisement filter. This procedure is demonstrated in the following example.

An AppleTalk network consists of two networks connected through a serial port. Each network contains its own private zone (“gsremote” and “gsmain”) and one common zone (“HTS”). The user wants to allow the resources on the common zone to be accessible by both networks but keep the resources on the private zone accessible only by the local network.

Procedure

To configure zone advertisement filtering on NETBuilder II A, follow these steps:

1. Configure entity filter "=:=@HTS" and assign the entity filter number 1 by entering:

   ```
   ADD -AT EntityFilter 1 "=:=@HTS"
   ```

2. Configure entity filter number 1 to port 7 by entering:

   ```
   ADD !7 -AT ZoneAdvFilterNm 1 Negative
   ```

3. Enable zone advertisement filtering on port 7 by entering:

   ```
   SETD !7 -AT CONTrol = ZoneAdvFilter
   ```

   From network A, only zones "HTS" and "gsmain" will be advertised. The zone "gsmain" is advertised because it is the local zone for network A. A Chooser or Interpool will only see those two zones.

   From network B, zones "HTS", "gsmain", and "gsremote" will be advertised. There are no zone advertisement filters configured on NETBuilder II B. A Chooser or Interpool on network B will see all the zones.

4. Prevent gsmain from being advertised to network B port 8 by entering:

   ```
   ADD -AT EntityFilter 1 "=:=@HTS"
   ADD !8 -AT ZoneAdvFilterNum 1 Negative
   SETD !8 -AT CONTtol = ZoneAdvFilter
   ```

   A chooser or Interpool will only see zones "HTS" and "gsremote" The zone "gsremote" was advertised because it is the local zone for network B.

Procedure

To use per-port directional entity filtering to achieve the same effect as zone advertisement filtering, follow these steps:

1. Configure per-port entity filtering on NETBuilder II A by entering:

   ```
   ADD -AT EntityFilter 1 "=:=@HTS"
   ```

2. Configure entity filter number 1 to port 7 by entering:

   ```
   ADD !7 -AT EntityFilterNum 1 Negative ClientIn
   ```

   The negative value specifies that only NBP Requests ("=:=@HTS") entering port 7 will be allowed.

3. Enable entity filtering on port 7 by entering:

   ```
   SETD !7 AT CONTrol = EntityFilter
   ```

   The Zip ZoneList request will return all the zones. Therefore, the Chooser or Interpool will see zones "HTS," "gsmain," and "gsremote." When the Chooser or Interpool tries to find devices on "gsremote," the NBP request will be filtered.

4. To prevent a Chooser or Interpool from network B from accessing "gsmain" devices, set the entity filter to filter NBP requests exiting port 7 by entering:

   ```
   DELete !7 -AT EntityFilterNum 1
   ADD !7 -AT EntityFilterNum 1 Negative ClientBoth
   ```

   The ClientBoth parameter applied the filter to both NBP requests entering and exiting port 7. The negative value specifies that only NBP requests ("=:=@HTS") are allowed.

   Both configurations can be done to filter zone advertisements and NBP requests simultaneously.

Changing a Zone List

You may need to change a zone list on an AppleTalk network to add a new subset of devices for service access on a large link or to correct an error introduced during the initial configuration.

To change a zone list, follow these steps:

1. Disable AppleTalk routing on all ports (on all routers) connected to the AppleTalk network using:

   ```
   SETDefault !<port> -AppleTalk CONTrol = NoROute
   ```

2. Reconfigure all seed routers on the AppleTalk network with the same zone list and default zone.

   - To add a zone name to the zone list, use:

     ```
     ADD !<port> -AppleTalk ZONe "<zone-string>" (1–32 char)
     ```

   - To delete a zone name from the zone list, use:

     ```
     DELete !<port> -AppleTalk ZONe "<zone-string>" (1–32 char)
     ```

   - To update the default zone, use:

     ```
     SETDefault !<port> -AppleTalk DefaultZone ="<zone-string>"
      (1–32 char)
     ```

3. After all routers on the extended AppleTalk internetwork have aged out the network from their routing tables, re-enable AppleTalk routing on all ports that you disabled earlier in step 1 using:

   ```
   SETDefault !<port> -AppleTalk CONTrol = ROute
   ```

A 15-minute wait is adequate for large networks. On some networks, re-enabling the routers too soon may result in difficulty in determining the zones for a specific network or finding services because some routers may have different zone lists for the same network.

3Com recommends rebooting all other AppleTalk devices on the modified AppleTalk internetwork, although some devices may adjust more easily. If there are any problems, reboot the router.

How the AppleTalk Router Works

This section discusses AppleTalk routing concepts, including information about using seed routers to provide network numbers and zone names to a connected network.

3Com bridge/routers provide complete AppleTalk Phase 2 routing capability by broadcasting routing information, forwarding packets, and responding to routing-related requests from AppleTalk-based workstations and other routers.

An AppleTalk router identifies information (including network numbers and zone names) for directly connected AppleTalk networks. The router uses network numbers to determine how to forward data to other networks on the AppleTalk internetwork. The router keeps zone information, which divides the internetwork into logical subdivisions, to help users access services through the AppleTalk internetwork.

Each of the ports associated with a physical interface on the system is considered to be connected to a different network. You determine which network ports on the system support AppleTalk routing.

Any grouping of networks connected by AppleTalk routers is known as an AppleTalk internetwork; each network on an internetwork can be on different physical media (for example, Ethernet, token ring, and FDDI).

The router that contains the primary identifying information associated with a physical network is called a *seed router*. A seed router must be the first router to be brought up on a network, preferably before any other AppleTalk devices are booted on the network. If a router is not a seed router for a network, it obtains the identifying information for the network (the network range, associated zone list, and default zone) from a seed router that is attached to the same network. After a router acquires the seed information from the seed router, it also can provide seed information to other routers and end nodes subsequently activated on the same network.

If bridging is enabled, AppleTalk Phase 1 packets are bridged through all active interfaces, regardless of the state of AppleTalk Phase 2 routing.

The identifying information that an AppleTalk Phase 2 router uses to keep track of networks on the internetwork includes:

- A network number range associated with each network.
- A zone list associated with each network.

A network number range is a unique range of contiguous network numbers, for example, 110–120, that identifies a particular AppleTalk network in a Phase 2 internetwork. A LocalTalk network, sometimes referred to as a non-extended network, is always identified by a network range consisting of a single network number (for example, 30–30) and a single associated zone. A network number in AppleTalk Phase 2 can be any number from 1 to 65,279 (0001 to hex FEFF).

The AppleTalk network number is the portion of packet destination addresses that allows the router to identify and route AppleTalk packets to the correct network.

A zone groups AppleTalk devices (nodes) within one or more networks so users can easily locate and access services (for example, printers and file servers). The networks or devices within a zone do not have to be adjacent or share common routers. Typically, they are geographically adjacent for routing efficiency and easy physical access to devices, such as printers.

The number of zone names you associate with a network depends on the size of the internetwork you are planning. If your internetwork is small, a single zone name may be adequate for all networks. If a single Ethernet or token ring network spans a large geographic area or contains large numbers of AppleTalk devices (such as printers or file servers), then use multiple zones to make it manageable for users.

In AppleTalk Phase 2, LocalTalk networks must be associated with a single zone; Ethernet, token ring, FDDI, and SMDS networks can be associated with multiple zones. In AppleTalk Phase 2, a default zone is identified within the zone list for

a network; the default zone is defined by a seed router. Individual nodes on a network are usually automatically configured to be in the default zone, and can be explicitly configured to be in a different zone present in the network's zone list.

AppleTalk routers also use the mapping of zones to networks to support the distributed name database maintained by the AppleTalk Name Binding Protocol (NBP).

The Apple Macintosh Chooser interface provides the most common point of exposure to zones. If two or more zones exist on the connected AppleTalk internetwork, a list of all zones across all networks is presented to the user. When a user selects a zone and service icon in the Macintosh Chooser, the user sees a list of only those services that exist in the zone. For example, instead of selecting from a list of 20 LaserWriter printers connected to an internetwork, a user may see only the two LaserWriter printers that are within the selected zone. This feature makes printer and other service selection both easier and faster.

Macintosh users can determine, through the icons within the Network Control Panel, what zone they will default to; this choice is reflected in the initial zone that appears in the Chooser interface.

To display network-to-zone mapping information, enter:

```
SHow -AppleTalk NetZoneMapping
```

To list all networks that are associated with each zone in the AppleTalk internetwork, enter:

```
SHow -AppleTalk ZoneNetMapping
```

For information on assigning zone names and other zone-related functions, refer to Chapter 4 in *Reference for NETBuilder Family Software*.

Network Entities

A network entity is a named AppleTalk entity, usually a service (such as file service or a printer) associated with an AppleTalk socket on an AppleTalk node.

The entity name is a character string enclosed in quotes and made up of three fields: object, type, and zone. Object and type are separated by a colon; type and zone are separated by the at (@) sign. Up to 32 characters are allowed for each field in the entity name. Entity names are case-insensitive. The following is an example of an entity name:

"AppleShare Server:AFPServer@engineering"

If you are familiar with the Macintosh Chooser, the object name of network entities appears in the upper right corner. The type is a name associated with the icons that appear in the upper left corner, but not necessarily the same as the name under the icon itself. The zone, if more than one zone is defined in the AppleTalk internetwork, will be in a zone list in the bottom left corner.

Within the bridge/router, network entity names perform the following tasks:

- Name router ports for discovery and APING. (For information on APING, refer to Chapter 1 in *Reference for NETBuilder Family Software*.)
- Describe entity filter specifications. (For more information on entity filtering, refer to Chapter 4 in *Reference for NETBuilder Family Software*.)

Object and zone strings are names that appear in the AppleTalk network-aware user interface, primarily with the Macintosh Chooser window, but also in third-party applications. The character set used in these strings is the extended ASCII character set used within the Macintosh. The bridge/router user interface for AppleTalk provides a universal representation of the extended AppleTalk ASCII character set. The extended character set permits the use of foreign characters in configured strings (for example, zone names) that are seen by AppleTalk end systems. Foreign language characters can also be entered as input to query functions (for example, ANameLookup command), and names containing such characters can be displayed without loss of information.

To enter these characters, key an escape character followed by a two-digit hex code for the desired character. For example, to enter an ñ, you first enter the escape character, which is a backslash (\ ), followed by the two-character hex code for the desired character as listed in Table 14-1. In this case, the hex code is 96 (an ASCII value (decimal) of 150). To specify a zone string such as "mañana," you would enter the eight-character string "ma\96ana."

The AppleTalk Service displays the string in the same format in which it is obtained from another AppleTalk device. On a Macintosh, the example would appear as "mañana" in the Chooser, assuming that multiple zones are defined within the AppleTalk internetwork.

Table 14-1 Macintosh Extended Character Set

ASCII Value	Hex Equivalent	Macintosh Character	ASCII Value	Hex Equivalent	Macintosh Character
128	80	Ä	159	9F	ü
129	81	Å	160	A0	†
130	82	Ç	161	A1	°
131	83	É	162	A2	¢
132	84	Ñ	163	A3	£
133	85	Ö	164	A4	§
134	86	Ü	165	A5	•
135	87	á	166	A6	¶
136	88	à	167	A7	ß
137	89	â	168	A8	®
138	8A	ä	169	A9	©
139	8B	ã	170	AA	™
140	8C	å	171	AB	´
141	8D	ç	172	AC	¨
142	8E	é	173	AD	≠
143	8F	è	174	AE	Æ
144	90	ê	175	AF	Ø
145	91	ë	176	B0	∞
146	92	í	177	B1	±
147	93	ì	178	B2	≤
148	94	î	179	B3	≥
149	95	ï	180	B4	¥

(continued)

Table 14-1 Macintosh Extended Character Set (continued)

ASCII Value	Hex Equivalent	Macintosh Character	ASCII Value	Hex Equivalent	Macintosh Character
150	96	ñ	181	B5	µ
151	97	ó	182	B6	∂
152	98	ò	183	B7	∑
153	99	ô	184	B8	∏
154	9A	ö	185	B9	π
155	9B	õ	186	BA	∫
156	9C	ú	187	BB	ª
157	9D	ù	188	BC	º
158	9E	û	189	BD	Ω
190	BE	æ	223	DF	fl
191	BF	ø	224	E0	‡
192	C0	¿	225	E1	·
193	C1	¡	226	E2	‚
194	C2	¬	227	E3	„
195	C3	√	228	E4	‰
196	C4	ƒ	229	E5	Â
197	C5	≈	230	E6	Ê
198	C6	∆	231	E7	Á
199	C7	«	232	E8	Ë
200	C8	»	233	E9	È
201	C9	…	234	EA	Í
202	CA		235	EB	Î
203	CB	À	236	EC	Ï
204	CC	Ã	237	ED	Ì
205	CD	Õ	238	EE	Ó
206	CE	Œ	239	EF	Ô
207	CF	œ	240	F0	
208	D0	–	241	F1	Ò
209	D1	—	242	F2	Ú
210	D2	“	243	F3	Û
211	D3	”	244	F4	Ù
212	D4	‘	245	F5	ı
213	D5	’	246	F6	ˆ
214	D6	÷	247	F7	˜
215	D7	◊	248	F8	¯
216	D8	ÿ	249	F9	˘
217	D9	Ÿ	250	FA	˙
218	DA	⁄	251	FB	˚
219	DB	¤	252	FC	¸
220	DC	‹	253	FD	˝
221	DD	›	254	FE	˛
222	DE	fi	255	FF	ˇ

Port Startup Operations

After you set up and check the router according to the instructions in the previous sections, it is ready to do some packet routing. The following actions occur when the AppleTalk router (with CONTrol set to ROute and AppleTalk) starts up on a port connected to an AppleTalk network:

- The router acquires a provisional AppleTalk node address for the port using AppleTalk Address Resolution Protocol (AARP) until the final network range for the connected network is known. (Frame Relay, X.25, and PPP must be statically configured with a final address.)
- If the CONTrol parameter is set to SeedingAllowed, and the seed information is configured using the NetRange, ZONe, and DefaultZone parameters, the following applies:
  - Using AARP, the router dynamically acquires a final AppleTalk node address with the network number taken from the configured network range. If the value of the StartupNET parameter is within the configured network range, the values for the StartupNODe (if nonzero) and StartupNET parameters are used as first attempt values in the process. If these values are tried but are already in use by another node, then an attempt is made to use the last address acquired from the previous startup, provided that it is in the proper network range. If this also fails, then the router finds a unique address in the configured network range.
  - If SeedCheck is enabled, and locally configured seed information is different from that seen for any other router on the network during the first twenty seconds of port activity, then the port is disabled. Information describing conflicting configurations is saved. You can display the information that describes configuration conflicts using the SHow -AppleTalk DIAGnostics command.
  - If NoSeedCheck is enabled, the router uses the locally configured seed information. If a difference in seed information between the local configuration and any other router on the connected network is detected, the last occurrence of conflicting information detected is saved. A difference in seed information does not disable the port in this case. You can display the conflicting information using the SHow -AppleTalk DIAGnostics command.
- If the CONTrol parameter is set to NoSeedingAllowed, or you do not have sufficient seed information configured, then the router does not seed, but waits for a seed router to appear.
  - If a seed router appears on the connected network, the router obtains the seed information from that router and proceeds.

    The router performs dynamic node address acquisition using AARP by selecting a network number from the network range given in the seed information. If the value of the StartupNET parameter is within the configured network range, then the values for the StartupNODe (if nonzero) and StartupNET parameters are used as first attempt values in the process. If these values are tried but are already in use by another node, then an attempt is made to use the last address acquired. If this also fails, then the router finds any unique address in the configured network range.
  - If no other seed router is detected on the connected network, the router remains in this listening state indefinitely.

After seed information is established for at least one of the active ports, the router begins to construct a routing table, which contains next router and distance information for all reachable networks and zone lists for each network. The tables are constructed from routing information (RTMP) packets received periodically from other routers. As new routes are discovered from these packets, the receiving router will ask the sending router for zone list information for each new network.

The maintenance of zone list information by the router allows the router to support access by AppleTalk end systems to named network entities. Routers supply AppleTalk end systems with the list of zones to assist in the location of end services. AppleTalk routers also support the discovery of named entities by using zone-to-network associations present in the routing tables.

Network AppleTalk Operations

This section provides an overview of AppleTalk operations on the network, particularly the routing function.

An AppleTalk network is usually configured on each port where AppleTalk packets are received and sent. The port can be a local area port, such as Ethernet, FDDI, or token ring. SMDS is supported over extended distances in an almost identical manner to that of the local area networks. AppleTalk can also be routed over a backbone network not configured as an AppleTalk network. This routing is usually done with a serial line port for a wide area network, such as a PPP, X.25, SMDS, or Frame Relay link. PPP, X.25, and Frame Relay links can also be set up as AppleTalk networks, but more configuration is required.

A router must check its routing table to determine where to route a packet. If the destination node is on a directly connected network, the router sends the packet directly to the destination node. If the network identified in the destination address is not directly connected, the packet is forwarded to the next router in the route to the destination network as maintained in the routing table.

For an example and description of the AppleTalk routing table, refer to the AllRoutes parameter in Chapter 4 in *Reference for NETBuilder Family Software*.

Figure 14-4 is an example of an AppleTalk internetwork. The upper router depicted in the Engineering Zone is a seed router on port 1. The following zone information should be configured on the indicated routers to provide the pictured zone boundaries:

Finance Zone router

port 1: Zonelist: Finance

port 2: PortZone: Finance

Marketing Zone router

port 1: can take defaults from a seed router

PortZone: Marketing

port 2: Zonelist: Marketing

Engineering Zone upper router

port 1: Zonelist: Finance, Engineering, Marketing

port 1: Default Zone: Marketing (presumably most end nodes are in Marketing)

port 1: PortZone: Engineering

Engineering Zone lower router port 1: Zonelist: Engineering

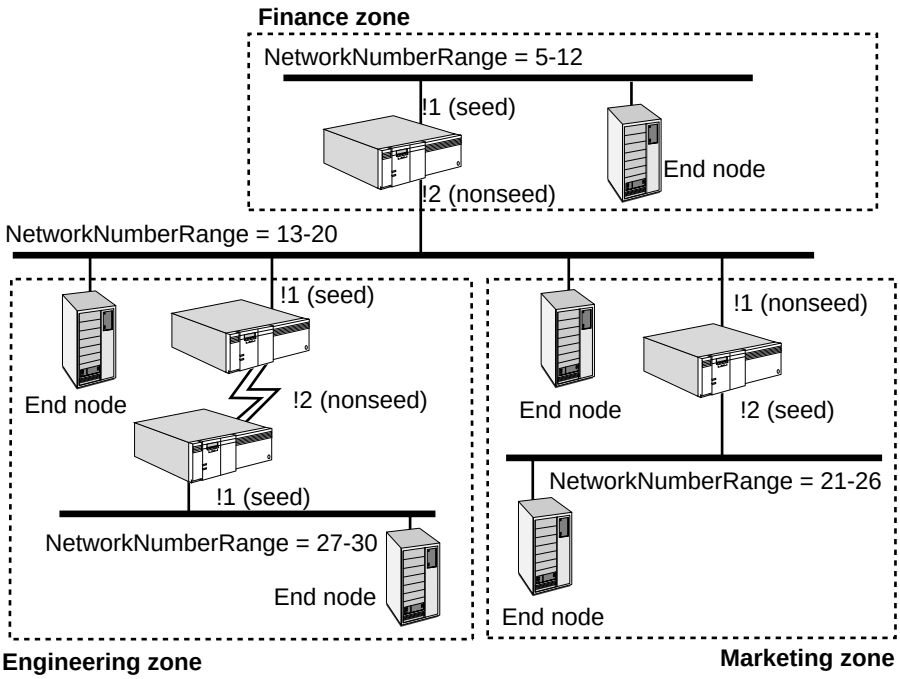


Figure 14-4 AppleTalk Network

Split Horizon

The AppleTalk router uses the split horizon routing method. This routing method helps reduce network traffic by not broadcasting route information for a network out the same interface over which the network's route was learned.

For Frame Relay and X.25 ports, split horizon decisions are made at the next router link level instead of at the port level. This feature allows support for nonmeshed topologies by allowing a router to use a Frame Relay or X.25 port as a virtual hub, sending route information to each router out the port learned from all other routers out of the same port. If the decisions were made at the port level, as is the case for AppleTalk on LANs and SMDS, no routing information learned from any router out of the port would be sent to any router out of the same port.

AppleTalk over PPP

A PPP link routing AppleTalk is normally configured as a non-AppleTalk data link because PPP does not support AppleTalk Address Resolution Protocol (AARP). The two sides of the link may choose the same network and node address if the link is configured as an AppleTalk data link. In this case, AppleTalk routes are not updated properly on both sides of the link. If you decide to configure a PPP link as an AppleTalk data link, enter unique startup network but different unique startup node numbers on the PPP port of both routers using:

```
SETDefault !<port> -AppleTalk StartupNET = <number> (0–65279)
SETDefault !<port> -AppleTalk StartupNODe = <number> (0–253)
```

Filtering on Frame Relay Ports

To apply filtering to or from specific neighbors out of the same Frame Relay port, you must use the virtual port feature. For more information on virtual ports, refer to Chapter 1.

Routing Table

Access the AppleTalk routing table using:

```
SHow !<port> -AppleTalk AllRoutes
```

For a sample display and explanation of a routing table, refer to Chapter 4 in *Reference for NETBuilder Family Software*.

The RTMP establishes and maintains the AppleTalk routing tables. Routing table entries identify the shortest possible path (measured in hop counts) to the network by identifying the next route to which packets should be sent.

AppleTalk always selects the route that requires the fewest hops. When packets are forwarded, a hop count field is incremented. Packets with a hop count of 15 or more are not forwarded to avoid indefinite looping.

15 CONFIGURING DECNET ROUTING

This chapter describes the procedures for configuring your system to perform DECnet Phase IV routing, route filtering, Phase IV to Phase V transition, and internetworking. It also describes how the router works and gives guidelines for operating, managing, and troubleshooting it.

For conceptual information, refer to "How the DECnet Router Works" on page 15-7. If you need help with terminology, refer to "DECnet Phase V and Phase IV Terms" on page 15-18.

Setting Up a Basic DECnet Router

The procedures in this section describe how to route DECnet packets within a DECnet network. Depending on your network requirements, you can use the default values of the parameters in the DECnet Service, or you may want to further configure the router according to "Customizing the Configuration" on page 15-5.

DECnet routing supports multiple independent DECnet networks attached to the router. It also allows internetwork routing either among all nodes on the selected networks, or between specific nodes on selected networks through user-defined address translations.

Unless otherwise noted, each command in the following procedures can be used whether you are configuring a Level 1 (intra-area) or Level 2 (interarea) router.

Configuring for Local Area Networks and Point-to-Point Links

Use this procedure to configure basic DECnet routing over LAN ports and Point-to-Point Protocol (PPP) links.

Prerequisites

Log on to the system with Network Manager privilege and set up the ports and paths of your bridge/router according to the procedure in Chapter 1.

Procedure

To configure the bridge/router to perform DECnet routing, follow these steps:

1 Set the DECnet address for this router using:

```
SETDefault -DECnet ADDRess = None | <area number>.<node
 number>(1–63).(1–1023) [<network> (0–15)]
```

The area number is a decimal number in the range of 1 to 63. The node number is a decimal number in the range of 1 to the value specified for the MaxNodeNumber parameter. The node number must be unique within an area number. For example, if a node with the address 3.1 already exists, do not set the address for this router to 3.1.

The value entered for the area number should not exceed the value configured for the MaxAReaNumber parameter. The default for the area number is 63. The value entered for the node number should not exceed the value entered for the MaxNodeNumber parameter. The default for the maximum node number is 255. For more information on the MaxAReaNumber and MaxNodeNumber parameters, refer to Chapter 17 in *Reference for NETBuilder Family Software*.

Enable bridging and all routing protocols before enabling DECnet.

2 Enable DECnet routing on a particular port using:

```
SETDefault !<port> -DECnet CONTrol = ROute
```

Repeat this step for other ports, including serial line ports.

If DECnet routing is enabled on a serial line port, the system at the other end of the serial line also must be routing (not bridging) DECnet traffic. DECnet assumes that serial lines are point-to-point links. Bridging DECnet packets on the other end of the serial line confuses the router, since it assumes that the address of the system on the other end of the serial line keeps changing.

When DECnet routing is enabled, the system address changes from the original media access control (MAC) address to the DECnet-derived address, which is based on its area and node numbers. This address change affects the static routes on other bridge/routers configured to use this bridge/router as the next hop.

CAUTION: *If you enable DECnet routing on a path where you have reassigned the MAC address using LAN Address Administration (LAA), you may affect the DECnet address for that path. For more information on LAN Address Administration and how it may affect DECnet addresses, refer to Chapter 28.*

3 Select the desired type of routing by entering one of the following commands.

To enable both intra- and interarea (level 2) routing, enter:

```
SETDefault -DECnet NodeType = Area
```

To enable intra-area (level 1) routing only, enter:

```
SETDefault -DECnet NodeType = RoutingIV
```

4 If any node in the area selected has a higher number than 255, increase the MaxNodeNumber parameter using:

```
SETDefault -DECnet MaxNodeNumber = <value>
```

5 Verify the DECnet configuration by entering:

```
SHow -DECnet CONFiguration
```

The router displays the DECnet configuration information. If the CONTrol parameter is not set to route, or if the address that you just configured is incorrect, repeat steps 1 and 2.

For detailed information on these parameters, refer to Chapter 17 in *Reference for NETBuilder Family Software*.

To complete the configuration for PPP links, refer to Chapter 34.

If DECnet routing is enabled after Internetwork Packet Exchange (IPX) routing, 3Com recommends flushing the existing IPX routing tables of adjacent IPX routers.

Configuring for Wide Area Networks

Routing DECnet over Frame Relay, Asynchronous Transfer Mode (ATM), Asynchronous Transfer Mode data exchange interface (ATM DXI), and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to route DECnet over Frame Relay, ATM, ATM DXI, or X.25 in a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM, ATM DXI, or X.25 cloud. For complete information on configuring DECnet routing over Frame Relay, ATM DXI, and X.25, including a discussion on fully meshed, partially meshed, and nonmeshed topologies and virtual ports, refer to Chapter 42, Chapter 43, Chapter 47, and Chapter 45, respectively. For information on the number of virtual ports supported per platform, refer to "Virtual Ports" on page 1-3.

Routing DECnet over Switched Multimegabit Data Service (SMDS) is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your DECnet router to perform routing over SMDS, refer to Chapter 44.

To configure DECnet routing over PPP or Phone Line Gateway (PLG), refer to Chapter 34. For information on wide area networking using Integrated Services Digital Network (ISDN), refer to Chapter 35.

Verifying the Configuration

Before you use a router to interconnect networks, verify that the routers you configured are recognized by the network and are functional by following these steps:

1. Check the routing table by entering:

   SHow -DECnet AllRoutes

   The routing table displays all DECnet areas and nodes to which a router has access. Check to make sure that the routers and end nodes you configured appear in the routing table.

2. Check the status of the ports previously configured on your router by entering:

   SHow -DECnet STATUS

   The DECnet status table displays the status of the ports for this router. Ports configured with DECnet routing enabled should be in the RUNNING state, which indicates that the port is operational.

   For a description of other status states, refer to Chapter 17 in *Reference for NETBuilder Family Software*.

3. Check the values of the path parameters by entering:

   SHow -PATH CONFiguration

4. Check the current DECnet routing parameters using:

   `SHow !<port> -DECnet CONFiguration`

   If the problem persists after these steps are taken, contact your network supplier or 3Com for assistance.

Getting Statistics

To view statistics, enter:

```
SHow -SYS STATistics -DECnet
```

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information on these parameters, refer to Chapter 58 in *Reference for NETBuilder Family Software*. For information on interpreting the statistics displays, refer to Appendix H.

Troubleshooting the Configuration

If you are unable to make connections to nodes within the local area or nodes in other areas after setting up the router, review the following troubleshooting procedure. Using this procedure can correct problems in making single-hop (involving one router) and multiple-hop (involving more than one router) connections. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier for assistance.

To troubleshoot the DECnet configuration, follow these steps:

1. Check that all cables on all routers in a specific path in the routing table are properly connected and that the routers are properly installed.

   For installation instructions, refer to the installation guide for your bridge/router.

2. Check that the state of the port is Up by entering:

   ```
   SHow -PORT CONFiguration
   ```

   If the state of the port is not Up, check that you have correctly completed the basic installation described in Chapter 1 in *New Installation for NETBuilder II Software.*

3. Check the status of the CONTrol parameter by entering:

   ```
   SHow -DECnet CONTrol
   ```

   The router displays the current values for the CONTrol parameter. If the CONTrol parameter for a port is set to NoROute, enable the DECnet router using:

   ```
   SETDefault !<port> -DECnet CONTrol = ROute
   ```

   Enable bridging and all routing protocols before enabling DECnet routing.

4. Check the status of the ports on your router by entering:

   ```
   SHow -DECnet STATUS
   ```

   The DECnet status table displays the status of the ports for this router. Ports configured with DECnet routing enabled should be in the RUNNING state, which indicates that the port is operational. If a port is in the DOWN state:

   - Check the port and the associated path configuration to see if they are enabled.
   - Enable the port and/or path if necessary.
   - Check the cables along the associated path to ensure that they are properly connected.

   For a description of other status states, refer to Chapter 17 in *Reference for NETBuilder Family Software*.

5 Check whether the node you are trying to reach is in the DECnet routing table by entering:

```
SHow -DECnet AllRoutes
```

The DECnet router displays the DECnet routing table entries. From the table entry, you can determine the path being used. Examine the entries to make sure a route in the table is taking the appropriate path.

Customizing the Configuration

This section provides additional procedures you can use to configure your DECnet router. For details on parameters, refer to Chapter 17 in *Reference for NETBuilder Family Software*.

Controlling Routing Information

The DECnet route filters allow you to control the routing information that the router advertises to or accepts from adjacent routers on a specified port. You can also control the list of adjacent routers on a specific port to send to or listen for routing information.

For a brief explanation of the route filtering parameters, refer to "Related Information" on page 15-6. For more information about the DECnet route filtering parameters, refer to Chapter 17 in *Reference for NETBuilder Family Software*.

To add a DECnet address, a list of addresses, or a range of addresses to the route list in the procedure that follows, use the ADD command. To exclude specific routes, use the ADD command with the tilde (~) prefix before each DECnet address to be excluded. For details on specifying lists and ranges of DECnet addresses, refer to Chapter 17 in *Reference for NETBuilder Family Software*.

Procedure

To define route filtering, follow these steps:

All the parameters in this procedure are port-specific.

1 Specify the routes advertised in routing updates using:

```
ADD !<port> -DECnet AdvertisePolicy <DECnet address>
```

To exclude a specific address, use:

```
ADD !<port> -DECnet AdvertisePolicy ~<DECnet address>
```

2 Specify the routes that are accepted from the routing updates of an adjacent router using:

```
ADD !<port> -DECnet ReceivePolicy <DECnet address>
```

3 Add a DECnet address to the list of adjacent routers that receive routing updates from this router using:

```
ADD !<port> -DECnet AdvToNeighbor <DECnet address>
```

4 Specify a list of trusted adjacent routers to listen for router hellos and routing updates using:

```
ADD !<port> -DECnet RcvFromNeighbor <comma-separated list of
 DECnet addresses>
```

5 Enable DECnet route filtering using:

```
SETDefault !<port> -DECnet PolicyControl = (AdvertisePolicy,
 ReceivePolicy)
```

Related Information

There are four route filtering parameters:

- AdvertisePolicy allows you to specify the routes that are advertised to adjacent routers in routing updates.
- ReceivePolicy allows you to specify the routes that are accepted from adjacent routers and cached in the routing tables.
- AdvToNeighbor allows you to specify the adjacent routers where routing updates may be sent.
- RcvFromNeighbor allows you to specify from which adjacent routers to accept hellos and routing updates.

When all four routing policies are configured and enabled, the following route filtering occurs:

- Before a routing update is transmitted onto the outbound port, the local routing information is filtered by the AdvertisePolicy parameter. This filtered information then is sent to the set of adjacent routers specified by the AdvToNeighbor parameter.
- When a router receives a routing update, only routing updates reported by the set of adjacent routers specified by the RcvFromNeighbor parameter are accepted. These routing updates then are filtered by the ReceivePolicy parameter before the reported routes are cached in the local routing database.

To enable and disable route filtering, use the PolicyControl parameter.

Setting the Priority

The PRIOrity parameter changes the priority of the router on the LAN. The router with the highest priority is elected as the designated router on the attached LAN. If multiple routers on the LAN have the highest priority, the router with the highest node ID is elected as the designated router.

To set the router priority, use:

```
SETDefault !<port> -DECnet PRIOrity = <number> (1–127)
```

Setting the Cost

The COST parameter allows you to change the route cost associated with the attached network. For DECnet Phase IV routing, packets are forwarded to the destination using the least-cost route.

To specify the cost associated with a network, use:

```
SETDefault !<port> -DECnet COST = <number> (1–25)
```

Enabling and Disabling Triggered Routing Updates

The CONTrol parameter allows you to choose triggered or complete routing updates as well as enabling and disabling routing.

Triggered updates occur whenever the routing table changes. Complete routing updates occur at intervals determined by the setting of the RoutingTime

parameter (refer to "Setting the Routing Time"). Complete updates are always sent at regular intervals, regardless of the Trigger/NoTrigger setting.

To select triggered routing updates, use:

```
SETDefault !<port> -DECnet CONTrol = Trigger
```

To deselect triggered routing updates, use:

```
SETDefault !<port> -DECnet CONTrol = NoTrigger
```

Setting the Routing Time

The RoutingTime parameter allows you to specify the timer interval (in seconds) at which the router sends complete routing updates to adjacent router nodes.

To set the routing time, use:

```
SETDefault !<port> -DECnet RoutingTime = <seconds>(5–65535)
```

Setting the Hello Messages Time

The HelloTime parameter sets the frequency at which the router sends hello messages to adjacent nodes. The value of the HelloTime parameter also determines the value of time-to-live (TTL) as seen by its adjacent nodes. The TTL of an adjacent node is based on its HelloTime parameter value. The formula used to calculate time-to-live is given in "Related Information."

Procedure

To set the HelloTime parameter, use:

```
SETDefault !<port> -DECnet HelloTime = <seconds>(5–8191)
```

Related Information

The following formula is used to calculate TTL:

TTL = K * <value of HelloTime parameter>

where:

K = 2 if the adjacent node is on a serial line

or

K = 3 if the adjacent node is on a LAN

For example, if the value of the HelloTime parameter configured for the adjacent node on a LAN is 30 seconds, then the TTL for the adjacent router is 90 seconds. If the local node does not receive a hello message from the adjacent node before the TTL counts down from 90 to 0 seconds, the adjacent node is declared down.

How the DECnet Router Works

This section provides information on how the DECnet router works.

DECnet Network

A DECnet network is configured on each port where DECnet packets are received and sent. The port can be any LAN or WAN port.

A DECnet network consist of nodes that do and do not route packets. Nodes that do not route packets (a host such as a VAX station, for example) are called end nodes. Nodes that route packets are called routers. One router per LAN

(Ethernet, token ring, Fiber Distributed Data Interface (FDDI)) has the additional role of routing packets for the end nodes on that LAN. These nodes are called designated routers. The PRIOrity parameter can be used to force a particular router to be the designated router on the LAN. For more information on this parameter, refer to "Customizing the Configuration" on page 15-5.

Routing nodes must be configured (with the NodeType parameter) as either Level 1 or Level 2 routers. A Level 1 router routes packets within a local area only. A Level 2 router routes packets both within a local area (intra-area) and between areas (inter-area).

Figure 15-1 is an example of a DECnet network. This DECnet network is composed of four LANs, which are separated into three areas.

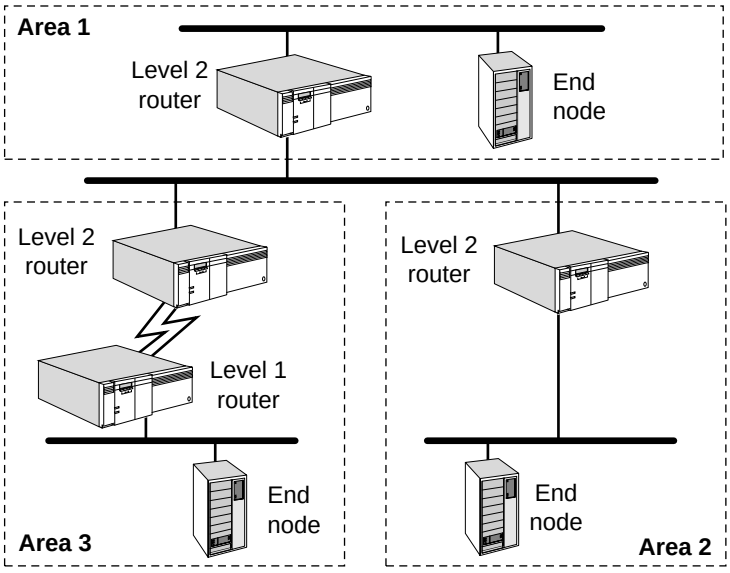


Figure 15-1 DECnet Network

A router must check its routing table to determine where to route a packet.

- If the destination node is within a local area, the router can forward the packet directly to the destination node or to the next-hop router, if appropriate, which sends it to the destination node.
- If the destination node is in another area, the router sends the packet to the nearest Level 2 router in the local area, which sends it to a Level 2 router in the other area. The Level 2 router in the other area then forwards the packet to the destination node in the same way that a Level 1 router does.

Routing Tables

Display the DECnet routing table using the AllRoutes parameter.

A Level 2 router displays two types of routing tables: the DECnet Level 1 (intra-area) routing table and the DECnet Level 2 (interarea) routing table. The DECnet Level 1 routing table displays information on nodes located within the local area that the router can reach. The DECnet Level 2 routing table displays information on other areas that the router can reach. A Level 1 router displays the DECnet Level 1 routing table only.

Each entry in the DECnet Level 1 routing table includes the following types of information that determine how a packet is routed:

- Reachable intra-area destination (node and port)

  The DECnet node address and port number of reachable intra-area destinations.

- Next hop

  The DECnet address of the next router to which a packet is forwarded on its way toward its destination.

- Cost

  The cost value associated with using the indicated intra-area route. In a DECnet network, packets are routed to their destination using the route with the smallest total cost. The COST parameter configures the cost value for each port. The route cost indicates the total cost of traversing one or more network interfaces to reach the intra-area destination.

- Number of hops between router and destination

  The number of hops is equal to the number of routers traversed to reach the destination node.

- BlkSize

  The maximum packet size that can be sent to that end node.

- Priority

  The priority of the router on the LAN. The priority determines which router on the LAN will be the designated router. The designated router is the router with the highest priority. If two or more routers have the highest priority, the router with the highest node ID becomes the designated router.

- TTL

  Indicates the time-to-live in seconds before the route is removed from the routing table. The HelloTime parameter configuration of the adjacent router or end node controls the TTL. For details on this parameter, refer to "Customizing the Configuration" on page 15-5.

Each entry in the DECnet Level 2 routing table includes the following types of information, which determine how a packet is routed:

- Reachable interarea destination (area and port)

  The DECnet area number and port number of reachable interarea destinations.

- Next hop

  The DECnet address of the next router to which a packet is forwarded for routing to its area destination.

- Cost

  The cost value associated with using the indicated interarea route. In a DECnet network, packets are routed to their destination using the route with the smallest total cost. The COST parameter configures the cost value for each port. The route cost indicates the total cost of traversing one or more network interfaces to reach the interarea destination.

- Number of hops between router and destination

  The number of hops is equal to the number of area routers traversed to reach the destination node.

- TTL

  Indicates the time-to-live in seconds before the area route is removed from the routing table. The value of the adjacent router's HelloTime parameter controls the TTL. For details on this parameter, refer to "Customizing the Configuration" on page 15-5.

The DECnet Level 2 routing table also summarizes the number of reachable areas, nodes within the local area, adjacent routers, and adjacent end nodes.

When the router learns multiple routes for a node or area, the least-cost route is always used to reach the node or area. For information on how the router makes the routing decision, refer to "Cost-effective Routing" on page 15-11.

Learning Routes

A router learns routes through routing update messages. These messages update the routing tables with all known destinations and their associated costs and numbers of hops.

Routing update messages are propagated throughout the network in the following manner:

- A node sends a routing update to an adjacent node (a node that is one logical hop away).
- When this adjacent node receives the routing update, it compares the information in the routing update with the information in its routing table.
- If the information in the routing update results in route changes in the routing table and the triggered update option is selected, a routing update with the new route information is generated and sent to the adjacent routers.
- Routing information changes are propagated to all router nodes on the network in this manner.

Level 1 routers send and receive messages to and from all adjacent nodes within the same area. Level 2 routers send and receive messages to and from all adjacent nodes within the same area as well as to and from adjacent Level 2 routers in other areas.

Complete routing updates are sent at user-configured time intervals. The frequency at which routing updates are sent is configured with the RoutingTime parameter. For more information on this parameter, refer to "Customizing the Configuration" on page 15-5.

However, if you have selected triggered routing updates and a router detects a change in the topology of your network (for example, a node is not operating), a routing update immediately reports to the adjacent routers that this node is unreachable. Refer to "Enabling and Disabling Triggered Routing Updates" on page 15-6.

Network Reachability and Split Horizon

A node is considered *reachable* when the computed cost and number of hops it takes to reach is less than the maximum cost and the maximum number of hops you configured for a router. To determine which nodes are reachable, check the routing table for each router.

The values that you set for the MaxCost, MaxHops, MaxAReaCost, and MaxAReaHops parameters determine the maximum cost and number of hops allowed for a node before the node is deemed unreachable.

The DECnet router avoids routing loops using *split horizon*. Split horizon prevents routing loops that may occur when a node includes information on other nodes learned from the same interface on which the routing update is sent. A DECnet router automatically uses split horizon with poison reverse by marking a route as unreachable in a routing update sent on the same interface from which the route was learned. Split horizon occurs automatically and requires no configuration.

Figure 15-2 illustrates how split horizon is used in DECnet routing. In this configuration, router A sends a routing update on port 1 that includes the following information:

- Router A is 0 hops away and has a cost of 0 (since this is information it is reporting on itself).
- Router B is 1 hop away and has a cost of 10.
- Router C is 2 hops away and has a cost of 20.

The routing message that router A sends on port 2 includes the following information:

- Router A is 0 hops away and has a cost of 0 (since this is information it is reporting on itself).
- Router B is 31 hops away and has a cost of 1023 (unreachable).
- Router C is 31 hops away and has a cost of 1023 (unreachable).

Split horizon prevents a router from advertising networks to any router it learned of those networks from. In this example, router A does not advertise to router B the route to router C. If the connection from router B to router C fails, split horizon prevents router B from sending packets bound for router C to router A.

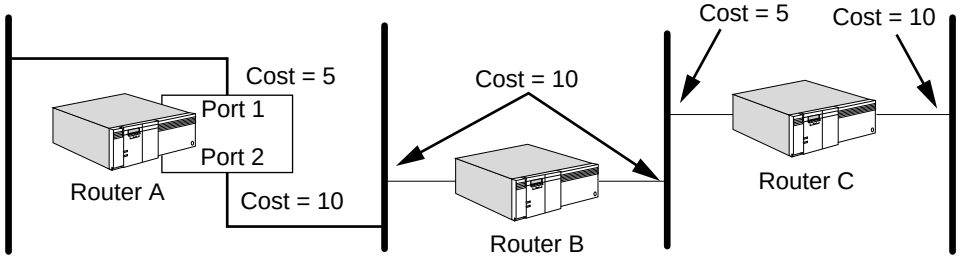


Figure 15-2 DECnet Routing Using Split Horizon

Cost-effective Routing

The DECnet router supports cost-effective routing, which means that the router selects the route with the lowest cost. The lowest-cost route is not necessarily the shortest (the route with the fewest hops). For example, imagine that two routes to another area exist. Route A requires three hops and has an associated

cost of 30. Route B requires four hops and has an associated cost of 25. Route B would be selected because it incurs the least cost (it is the most cost-effective route), although it requires more hops.

If a router has two routes with the same cost associated with each route, the router forwards packets to the router with the higher node ID.

Only the most cost-effective route appears in the routing tables.

Routing Phase IV Traffic over DOD Lines

For DECnet Phase IV environments where traffic is routed over dial-on-demand lines, routing updates and periodic hellos are suppressed once the router adjacency is established, and the routing database is synchronized. This allows the demand circuit to be brought down when it is not carrying traffic.

Address Translation Gateway Support

The Address Translation Gateway feature provides internetwork routing support and address translation for DECnet networks.

Internetwork Routing Support

A DECnet router can support one or more independent DECnet Phase IV networks attached to its LAN or WAN interfaces. Connectivity between the attached DECnet Phase IV networks is achieved through address translation.

Address Translation

Address translation allows connectivity between specific DECnet nodes on different networks that otherwise cannot communicate because of address conflicts between networks. Defining address translation for specific nodes on the selected networks resolves duplicate addresses and permits internetwork communication.

Address Translation Configuration Example

In the following sample configuration, a NETBuilder II bridge/router is attached to DECnet Phase IV networks.

The router is connected to network 0 through Ethernet port 1 as a Level 1 intra-area router with an address of 1.2. The router is also connected to network 2 through Ethernet port 3 as a Level 2 area router with an address of 5.1. Both networks are independent until the internetwork routing function is enabled through the user-defined address translation map.

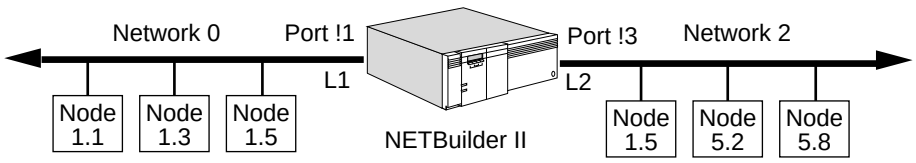


Figure 15-3 DECnet Address Translation Configuration

The following diagram illustrates the address conversion in a packet exchange between node 1.5 on network 0 and node 5.8 on network 2.

```
SA=1.5 / DA=1.8
---------------------->
                        SA=3.1 / DA=5.8
                        ---------------------->
                        SA=5.8 / DA =3.1
                        <----------------------
SA=1.8 / DA=1.5
<----------------------
```

The following diagram illustrates the address conversion in a packet exchange between node 1.1 on network 0 and node 1.5 on network 2.

```
SA=1.1 / DA=1.9
---------------------->
                        SA=3.2 / DA=1.5
                        ------------------->
                        SA=1.5 / DA =3.2
                        <-------------------
SA=1.9 / DA=1.1
<----------------------
```

Without the above address map, node 1.1 on network 0 cannot communicate with node 1.5 on network 2, because of address conflicts between the networks.

To configure the sample address translation configuration, follow these steps:

1 Configure DECnet routing for network 0 by entering:

```
SETDefault -DECnet ADDRess = 1.2
SETDefault -DECnet NodeType = RoutingIV
SETDefault -DECnet MaxNodeNumber = 512
SETDefault !1 -DECnet CONTrol = ROute
```

2 Configure DECnet routing for network 2 by entering:

```
SETDefault -DECnet ADDRess = 5.1 2
SETDefault -DECnet NodeType = Area 2
SETDefault -DECnet MaxAReaNumber = 7 2
SETDefault !3 -DECnet NETwork = 2
SETDefault !3 -DECnet CONTrol = ROute
```

3 Configure address translations between network 0 and network 2 by entering the following commands.

Map virtual node 1.9 on network 0 to real node 1.5 on network 2 by entering:

```
ADD -DECnet AddressMap 1.9@0 1.5@2
```

Map virtual node 3.2 on network 2 to real node1.1.on network 0 by entering:

```
ADD -DECnet AddressMap 3.2@2 1.1@0
```

Map virtual node 1.8 on network 0 to real node 5.8 on network 2 by entering:

```
ADD -DECnet AddressMap 1.8@0 5.8@2
```

Map virtual node 3.1 on network 2 to real node 1.5 on network 0 by entering:

```
ADD -DECnet AddressMap 3.1@2 1.5@0
```

The above address translation map allows nodes 1.1 and 1.5 on network 0 to communicate with nodes 1.5 and 5.8 on network 2.

4 To enable the configured address map to allow internetwork routing, enter:

```
SETDefault -DECnet InterNetRoute = AddressMap
```

Because the router does Level 1 intra-area routing on network 0, node 1.5 on network 0 cannot communicate with node 5.8 on network 2 without an address map. By defining node 5.8 as the virtual node 1.8 on network 0, node 1.5 can access node 5.8 by connecting to the virtual node 1.8 on network 0.

A packet received from network 0 and destined for the virtual address 1.8 will result in the conversion of the real address 5.8. The next hop to 5.8 is

determined by a lookup in the routing table for network 2. The source address is translated to its virtual address on network 2 and the packet is forwarded.

Virtual addresses 1.8 and 1.9 are advertised to network 0 as reachable nodes with zero cost/hop. The virtual area 3 is also advertised to network 2 as a reachable area with zero cost/hop.

The user-defined virtual address must not already exist in the associated network.

Because only configured virtual addresses are advertised to their associated network, only nodes that exist in the address translation map on both networks can communicate directly. In the sample configuration, node 1.3 on network 0 cannot communicate with any nodes on network 2. Node 5.2 also cannot access any nodes on network 0.

Internetwork Boundary Routing

Internetwork Boundary Routing software architecture allows connectivity between DECnet nodes in a Boundary Routing environment where each of the remote networks resides in a different DECnet area.

In this sample configuration (Figure 15-4), the central router is connected to network 0 on Ethernet port 1 as an area router with the address 10.1. The router is also connected to remote networks 1 through 4 through PPP links.

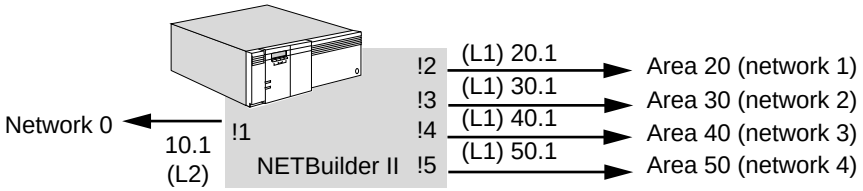


Figure 15-4 DECnet Internetwork Boundary Routing Configuration

Each of these networks exists as an independent network until the internetwork boundary is enabled.

To enable internetwork Boundary Routing, enter:

```
SETDefault -DECnet InterNetRoute = 0, 1, 2, 3, 4
```

All nodes on networks 0, 1, 2, 3, and 4 can now connect to each other. The router advertises areas 20, 30, 40 and 50 as reachable areas to network 0. Packets received from the remote networks that are destined to nodes on one of the networks configured for internetwork Boundary Routing are forwarded to that network if the destination is reachable.

Phase IV to Phase V Transition Support

The DECnet Phase V gateway provides coexistence and interoperability of DECnet Phase IV and Phase V (Open Systems Interconnect) nodes in a DECnet network. For more information, refer to "DECnet Phase V and Phase IV Terms" on page 15-18.

The following features permit interoperability: Phase IV to Phase V Translation and DECnet area to pseudo areas translation.

Phase IV to Phase V Translation

The DEC-compatible Phase IV to Phase V translation algorithm on addressing, data packet, and route advertisements is supported by the 3Com Phase IV to Phase V Transition Support feature. The translation allows Phase IV hosts to exist in Phase V networks and Phase V hosts to exist in Phase IV networks. The Phase IV hosts can communicate only with Phase V hosts that have Phase IV-compatible addresses. A Phase IV-compatible address is a Phase V address that is within the Phase IV addressing limits.

In Figure 15-5, Phase IV and Phase V end systems can communicate with each other using Phase IV routing, Phase V routing, or a combination of Phase IV and Phase V routing. The gateway provides the common routing path that enables these end systems to communicate in the same area or in different areas.

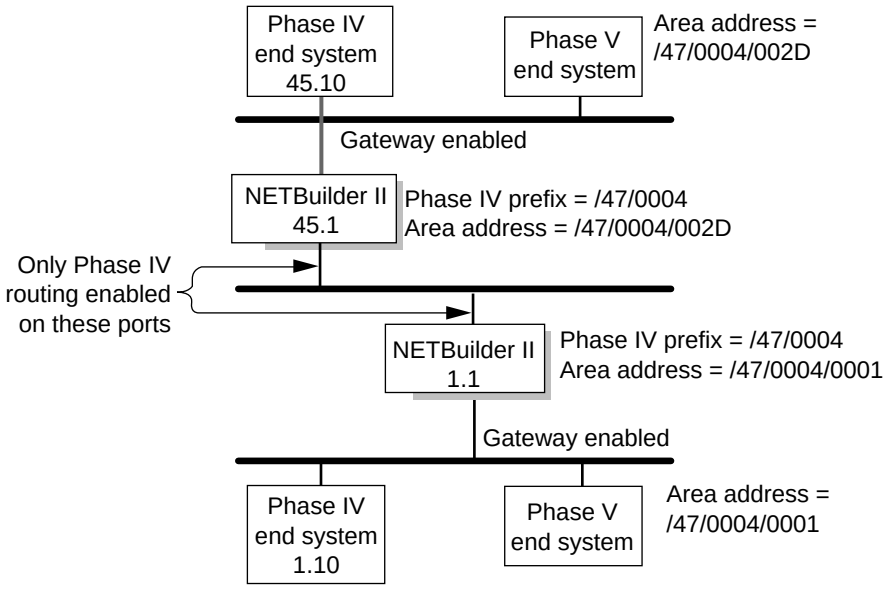


Figure 15-5 DECnet Phase IV to Phase V Translation

The 3Com implementation supports Phase IV and Phase V routing protocols in a single DECnet area. Supporting both protocols in a single area allows Phase V support to be added to a Phase IV area without modifying the existing Phase IV support. The 3Com router translates the routing information between the Phase IV and the Phase V routing environments. In Phase IV routing updates, 3Com routers advertise reachability to Phase V hosts that have a Phase IV-compatible address. 3Com routers also advertise reachable Phase IV hosts in Phase V Link State advertisements.

DECnet Area to Pseudo Areas Translation

A DECnet Phase V area that is Phase IV-compatible can be subdivided into multiple pseudo areas with a smaller address space. The pseudo areas allow a unique OSI area address to be assigned to each DECnet site within the common DECnet area (see Figure 15-6). This permits intersite communication through Level 2 routing (with static prefix routes) in a GOSIP-compliant OSI backbone network.

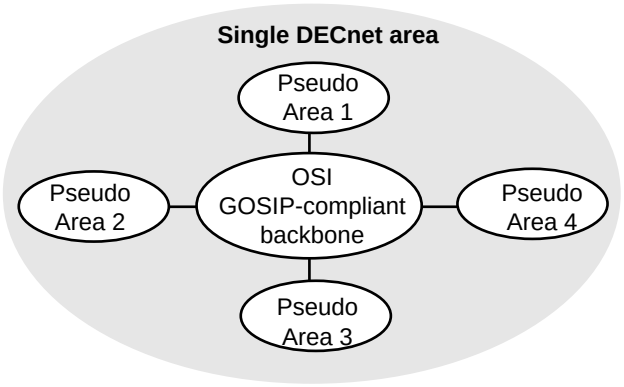


Figure 15-6 DECnet Pseudo Areas

When multiple DECnet sites share a single DECnet area, and connectivity between Phase IV and Phase V hosts must be maintained, to add Phase V routing support you need to configure all sites to reside in the same OSI area with the following area address:

<common Phase IV NSAP Prefix/common DECnet area ID>.

Sites that are connected to a GOSIP-compliant OSI backbone network require routing domain boundaries to restrict routing information exchanges. The result is the partitioning of the common OSI area into disjoint subareas.

Because Phase V nodes currently supports multihoming to only three area addresses, a loss of connectivity in the partitioned area may result. In this case, a pseudo area can be assigned to each site to work around the routing problem in the partitioned OSI area. The pseudo area address of a site, formed by concatenating the common pseudo area prefix and the pseudo area ID of the site, is unique in the common OSI area. This pseudo area address allows intra-area traffic of a site destined for another site to be routed across the backbone's routing domain boundary to the destination site using the backbone's Level 2 interarea routing.

When a packet is forwarded to a remote site, at each site the router maps the destination network service access point (NSAP) address that is within the address space of the common OSI area into its corresponding pseudo area address for intersite routing. When a packet is received from the OSI backbone, a destination pseudo area address is converted into its corresponding NSAP address for intrasite routing. The pseudo area addresses are used strictly for routing intra-area traffic across a partitioned area. The remote pseudo area addresses must be configured on a 3Com router as reachable NSAP address prefixes, using the PrefixRoute parameter in the ISIS Service. For more information about NSAP addressing, refer to Appendix E.

Pseudo Area Configuration

In Figure 15-7, sites 1 and 2 share the same DECnet area 45. Both sites are configured to support two pseudo areas. Site 1 is configured in pseudo area 90. In this area, node addresses 45.1 through 45.511 are mapped to addresses 90.1 through 90.511. The Phase IV end systems and Phase V end systems with a Phase IV-compatible address at both sites can communicate through the OSI GOSIP-compliant backbone.

Site 2 is configured in pseudo area 91 in which node addresses 45.512 through 45.1023 are mapped to addresses 91.1 through 91.511. The Phase IV and Phase V end systems with a Phase IV-compatible address at both sites can communicate through the OSI GOSIP backbone.

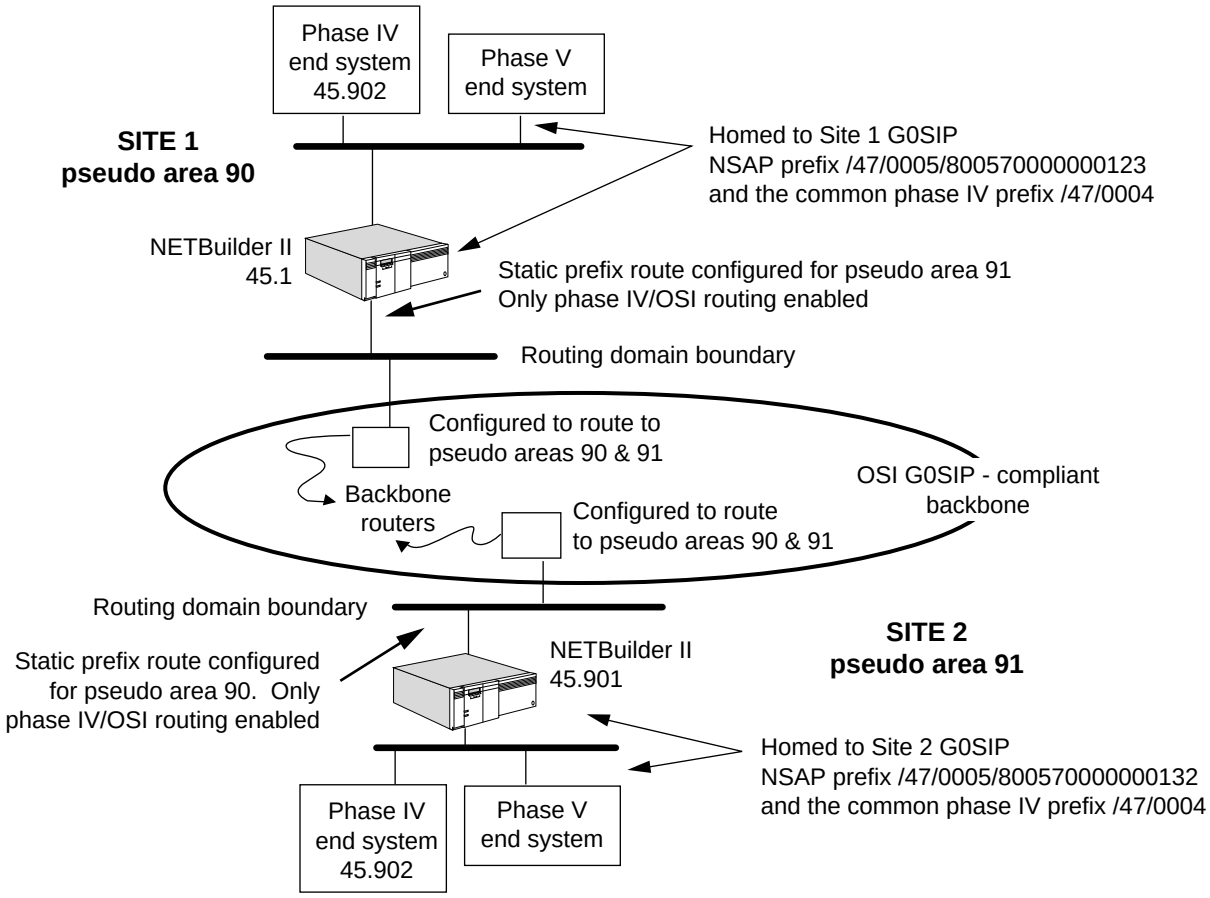


Figure 15-7 DECnet Pseudo Area Configuration

Phase IV to Phase V Transition Configuration Example

To configure Phase IV to Phase V transition on NETBuilder 45.1 based on the example in Figure 15-7, follow these steps:

1. Configure DECnet Phase IV routing by specifying the DECnet address to be used by the router. Enter:

   ```
   SETDefault -DECnet ADDRess = 45.1
   ```

2. Specify the node type. In the following command, the node type is Area. Enter:

   ```
   SETDefault -DECnet NodeType = Area
   ```

3. Enable DECnet routing on ports 1 and 2 by entering:

   ```
   SETDefault !1 -DECnet CONTrol = ROute
   SETDefault !2 -DECnet CONTrol = ROute
   ```

4. Configure Phase V OSI routing.

   The area ID field of the local OSI area address must match the local DECnet area number.

   When the DECnet gateway function is enabled, the area address, formed by concatenating the IVPrefix and the area number of the local DECnet Phase IV

address, must match one of the area addresses configured for the OSI router. Enter the following commands to set the OSI intermediate system area address.

a To specify the area address, enter:

```
ADD -ISIS AreaAddress /47/0004/002D
```

The NSAP address is specified in hexadecimal format. The area id %002D in the NSAP address matches the local DECnet decimal area number 45.

b To specify the intermediate system as a Level 2, enter:

```
SETDefault -ISIS Mode = Level2
```

c To enable the CLNP routing function, enter:

```
SETDefault -CLNP CONTrol = Route
```

The DECnet Phase IV address is specified in decimal format while the OSI area address is specified in hexadecimal format.

5 Configure Phase IV to Phase V translation.

The Phase IV NSAP prefix must match the area prefix of an existing OSI area address configured for the OSI router.

a To specify the common Phase IV NSAP Prefix, enter:

```
SETDefault -DECnet IVPrefix = /47/0004
```

b To enable the DECnet Phase IV to Phase V translation, enter:

```
SETDefault -DECnet GatewayControl = GateWay
```

6 Configure the pseudo area mapping.

In this example, the local pseudo area is 90 and the remote pseudo area is 91.

```
SETDefault -DECnet PseudoAreaPrefix = /47/0005/8000570000000123
SETDefault -DECnet MaxPseudoAreas = 2
```

The following information is displayed:

```
Local Pseudo Area address: /47/0005/80005700000001230090
Route Pseudo Area address: /47/0005/80005700000001230091
```

The MaxPseudoAreas configuration must be identical for all communicating pseudo areas.

7 Configure a static prefix route for the remote pseudo area 91 by entering:

```
SETDefault !1 -ISIS PrefixRoute /47/0005/80005700000001230091
 %080001020304
```

8 Enable the pseudo area translation by entering:

```
SETDefault -DECnet GatewayControl = PseudoArea
```

DECnet Phase V and Phase IV Terms

This section describes DECnet-specific terms:

DECnet Phase V	OSI-compatible. Phase V routing conforms to the ISO's CLNP, ES-IS, and IS-IS protocols. In addition, Phase V nodes are backward-compatible with Phase IV nodes. A Phase V node determines the packet format to use with an adjacent node based on the type of hello message received from that node.

Phase IV NSAP Prefix

The common NSAP Prefix. This prefix must be used in Phase V routing environments to allow communication between Phase IV and Phase V systems in a routing domain. A 3Com router serving as a DECnet gateway concatenates the configured Phase IV NSAP Prefix and its own Phase IV AreaID to form the Phase IV OSI area address for advertising reachable Phase IV nodes in Phase V areas.

Phase IV-compatible NSAP Address

DECnet Phase V nodes can communicate with DECnet Phase IV nodes through a 3Com router serving as a DECnet gateway when the Phase V node is configured with a Phase IV-compatible NSAP address. A Phase IV-compatible NSAP address is defined as: <NsapPrefix/AreaID/StationID/Selector>.

A Phase IV-compatible NSAP address assures that the address can be translated from Phase V to Phase IV and back again without change. A Phase IV-compatible NSAP address must conform to the following rules defined by DEC:

- The NSAP Prefix must match the Phase IV NSAP Prefix specified for the 3Com router with the IVPrefix parameter.
- The 2-octet AreaID has a value within the range of 0 to 63 and matches the low order 6 bits of the 6-octet StationID.
- The high order 32 bits of the StationID must match the DECnet architectural constant *AA-00-04-00* (hexadecimal).

Phase IV-compatible Area Address

Reachability information of Phase V nodes that are configured with a Phase IV-compatible NSAP address is advertised by the 3Com router to adjacent Phase IV routers.

A reachable Phase IV-compatible area address, <NsapPrefix/AreaID>, is advertised by 3Com routers as a reachable Phase IV area if:

The NSAP Prefix matches the Phase IV NSAP Prefix specified for the 3Com router with the IVPrefix parameter.

The 2-octet AreaID has a value within 0 to 63.

16 OSI ROUTING

This chapter describes how to configure, customize, and troubleshoot Open Systems Interconnection (OSI) routers.

For conceptual information, refer to "How the OSI Router Works" on page 16-9.

Setting Up a Basic OSI Router

The procedures in this section describes the minimum steps required to enable your system to route OSI packets. Depending on your network requirements, you may want to further configure the router according to later sections in this chapter.

Configuring for Local Area Networks and Point-to-Point Protocol Links

Use this procedure to configure basic OSI routing for LAN ports and Point-to-Point Protocol (PPP) links.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Set up the ports and paths of your bridge/router according to Chapter 1 and log on to the system with Network Manager privilege.
- It is assumed that you are familiar with the protocols supported by the router. Refer to ISO 8473 for information on connectionless mode network service, ISO 9542 for information on the End System-to-Intermediate System (ES-IS) Protocol, ISO 10589 for information on the Intermediate System-to-Intermediate System (IS-IS) routing Protocol, and ISO 8348, Addendum 2, for NSAP addressing.
- If you are using DECnet routing with OSI routing, you must configure DECnet routing before OSI routing. Configuring DECnet routing changes the Ethernet address of the router, and OSI routing protocols will not recognize the new Ethernet address.
- To configure the OSI router, you must set some parameters in the ISIS and CLNP Services.

Procedures

To configure the bridge/router to perform basic OSI routing, follow these steps:

1 Determine the area address of the router.

3Com bridge/routers are shipped with the default area address of /49/0053.

a To display the current area address of the router, enter:

```
SHowDefault -ISIS AreaAddress
```

b To configure the area address for the router, use:

```
ADD -ISIS AreaAddress <NSAP address>
```

For example, if you want to reset the area address of a router to /47/0004/00351100, enter:

```
ADD -ISIS AreaAddress /47/0004/00351100
```

Guidelines exist for setting area addresses. For more information on this topic, refer to "Area Addresses" on page 16-10.

c After changing or adding additional area addresses, delete any old area addresses.

For example, to delete the default area address, enter:

```
DELete -ISIS AreaAddress /49/0053
```

You can configure up to three area addresses. Multiple area addresses are normally used when transitioning your network from one configuration to another. For example, multiple area addresses can be used if you are introducing a new area address to replace an old one, you are merging two areas into one, or you are separating one area into two areas.

2 Determine whether a router is to perform as a Level 2 router; if necessary, configure it to perform as a Level 2 router.

The default routing type is Level 1(routing within an area or intra-area routing) only.

To configure the router to perform Level 2 routing (routing between areas or interarea routing), enter:

```
SETDefault -ISIS MODE = Level2
```

A router that is configured as a Level 2 router performs both intra-area and interarea routing.

3 Determine which ports are to be used for ISIS routing.

ISIS routing is enabled by default on all ports. To disable ISIS routing on a particular port on which you do not want ISIS routing to occur, use:

```
SETDefault !<port> -ISIS CONTrol = Disable
```

4 Enable the Connectionless Network Protocol (CLNP) routing function by entering:

```
SETDefault -CLNP CONTrol = Route
```

Enabling the routing function immediately starts the operations of both ES-IS and IS-IS routing protocols. The router becomes an IS, and it starts sending intermediate system hello (ISH) packets to the attached networks. Conversely, if the routing function is disabled, operations of both ES-IS and IS-IS routing protocols immediately stop.

5 If you have end systems that do not support the ES-IS Protocol, and the router needs to route packets to them, configure static routes for them by using:

```
ADD !<port> -CLNP ES <NSAP address> <SNPA>
```

For example, the following command adds a subnetwork point of attachment (SNPA) end system address for port 3:

```
ADD !3 -CLNP ES /47/0004/0035110008000200369101 %080002A01459
```

The port referenced in the command is the one where the end system (ES) is reachable. <NSAPaddress> is the NSAP address of the destination ES; <SNPA> is the MAC address of the ES, or may be the MAC address of another router through which the ES is reachable.

6 Configure an interdomain route using:

```
ADD !<port> -ISIS PrefixRoute
```

This step applies to Level 2 routers at a routing domain boundary only.

For more information on how to set up interdomain routing, refer to "Setting Up Interdomain Routing" on page 16-18.

To complete the configuration for PPP links, refer to Chapter 34.

Configuring for Wide Area Networks

You can configure the OSI router to perform routing over wide area network ports using Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), Switched Multimegabit Data Service (SMDS), X.25, and Integrated Services Digital Network (ISDN). To configure your OSI router to perform routing over Frame Relay, ATM DXI, SMDS, or X.25, refer to Chapter 42, Chapter 43, Chapter 44, or Chapter 45. For information on wide area networking using ISDN, refer to Chapter 35.

Verifying the Configuration

This section describes how to verify the router configuration, check with OPING, examine statistics, and check its overall status. Before you use the router for interconnecting networks, check to see whether it can route packets properly. Send packets from one network to another to see if they are properly forwarded.

To verify the router configuration, follow these steps:

1 Check whether all the ESs on the directly attached networks are included in the End System Table by entering:

```
SHow -CLNP ES
```

Check the network attachment for any ESs that are not included in the table. For more information on this table, refer to "End System Table" on page 16-17.

2 Check whether all the intermediate systems (ISs) on the directly attached networks are included in the Intermediate System Table by entering:

```
SHow -CLNP IS
```

Check the network attachment for any ISs that are not included in the table. For more information on this table, refer to "Intermediate System Table" on page 16-17.

3 Check whether all ISs on the directly attached networks have established an adjacency with this router by entering:

```
SHow -ISIS ADJacencies
```

Compare the entries in the displayed adjacency table with the entries in the Intermediate System Table. All Level 2 ISs should be adjacent with each other, with adjacency type L2ONLY. All Level 1 ISs with area addresses in common should be adjacent with each other, with adjacency type L1ONLY. (Level 1 ISs with different area addresses do not establish adjacencies with each other.) Neighboring routers configured with different Hello passwords are not adjacent.

4 Check the Level 1, Level 2, and Interdomain Routing Tables.

a To display the Level 1 Routing Table, enter:

```
SHow -ISIS L1Route
```

The Level 1 Routing Table summarizes all reachable systems, both ESs and ISs, within the area. Check to make sure this table displays all ESs and ISs within the area.

b To display the Level 2 Routing Table, enter:

```
SHow -ISIS L2Route
```

The Level 2 Routing Table applies only to routers that are configured as Level 2 routers. This table summarizes all reachable areas within the routing domain. Check to make sure that all areas are included.

c To display the Interdomain Routing Table, enter:

```
SHow -ISIS PrefixRoute
```

The Interdomain Routing Table applies only to routers configured as Level 2. This table summarizes all reachable routing domains outside of this routing domain.

5 Examine the configuration of ports by entering:

```
SHow -PORT CONFiguration
```

6 Examine the configuration of paths by entering:

```
SHow -PATH CONFiguration
```

7 Examine the CLNP configuration and the ES and IS tables by entering:

```
SHow -CLNP CONFiguration
```

8 Examine the ESIS configuration by entering:

```
SHow -ESIS CONFiguration
```

9 Examine the ISIS configuration and adjacency table by entering:

```
SHow -ISIS CONFiguration
```

10 Examine the Level 1 Routing Table by entering:

```
SHow -ISIS L1Route
```

11 Examine the Level 2 Routing Table by entering:

```
SHow -ISIS L2Route
```

12 Show a collective listing of all routing domains that can be reached from a particular routing domain by entering:

```
SHow -ISIS PrefixRoute
```

Checking Packet-Forwarding Process

After you have configured your routers for OSI, check to see if they can forward packets properly.

To check to see if your routers are configured properly, follow these steps:

1 Select one router in your network and attach a terminal to its console port.

2 Use the OPING command to verify proper routing to each of the other routers:

For example, to send an echo request message to a router having the NSAP address /47/0004/00351300080000200182400, enter:

```
OPING /47/0004/00351300080000200182400
```

You may receive one of the messages in Table 16-1.

Table 16-1 OPING Command Messages

Message	Meaning
`Pinging ... destination is alive`	Successfully reached destination: bidirection verified.
`dest unreachable according to local routing table`	The local router has no route.
`Pinging... received Error Report PDU code 128`	The local router has either a default or a Level2 route, but the path to the destination is not complete.

3 If an error report protocol data unit (PDU) code is received, use the OTraceRoute command to determine where the route fails.

For example, to trace the path to the destination /47/0004/0035130008000200182400, enter:

```
OTraceRoute /47/0004/0035130008000200182400
```

You will receive this message:

```
TTL                         Next_Hop_Address
1                           /47/0004/00351100080002033ad200
2                           /47/0004/0035150008000203892300
Destination Unreachable
```

This message indicates that the router /47/0004/0035150008000203892300, the last router attempting to reach the destination, did not have a route and returned an error response.

4 Access the last router to respond by entering the TELnet command.

In the example in step 3, the last router to respond is router 2, which has the NSAP address of /47/0004/0035150008000203892300. Using the example, enter:

```
TELnet /47/0004/0035150008000203892300
```

You will receive the following message:

```
N-selector changed to 06, trying /47/0004/0035150008000203892306
Connecting ... connected
Escape character is "^]'
NetLogin:
```

This message indicates that you have successfully connected to the last router attempting to route to the destination.

5 Find the next-hop router in the path toward the destination in the Level 2 Routing Table (AreaAddress /47/0004/00351300) by entering:

```
SHow -ISIS L2Route
```

You will receive this message:

```
Time since last table update: 179 sec. Update count: 13381
----AreaAddress----  -----Metric---  -----Port----  -------IS------
/47/0004/00351000    40             1              080002033AD2
/47/0004/00351100    20             1              080002033AD2
/47/0004/00351300    20             1              080002033AD2
/47/0004/00351500    0              -              080002033CC9
```

6 Find the network entity title (NET) of the next-hop router in the path toward the destination in the Intermediate System Table by matching the SystemID of the next-hop router in the Level 2 table with the SystemID portion of the Intermediate System Table.

Using the example in step 5, enter:

```
SHow -CLNP IS
```

You will receive the following message:

```
Intermediate System NetworkEntityTitle    SNPA
/47/0004/0003513000800020033CC900         %0800020303D6
```

7 After you have identified the router that cannot forward your packet, use TELnet to access it, and check the Level 1 Routing Table by entering:

```
SHow -ISIS L1Route
```

You will receive this message:

```
Time since last table update: 133 sec. Update count: 8213
----SystemID----    -----Metric-----  -----Port----  -----SNPA/IS-----
#080002033CC9       0                 -              -
*080002013C37       20                1              IS 080002013C27
Total 0 ES routes, 2 IS routes
```

The SystemID is not in the Level 1 Routing Table, and this table has all of the ESs and ISs for the Area/47/0004/00351300. There is only one other router in this area, and it is not the one you want to reach. For more information, refer to "Troubleshooting the Configuration" on page 16-6.

Getting Statistics

To examine the statistics of the OSI router, follow these steps:

1 After the router is up and running, examine the CLNP statistics by entering:

```
SHow -SYS STATistics -CLNP
```

2 After the router is up and running, examine the ISIS statistics by entering:

```
SHow -SYS STATistics -ISIS
```

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information on these parameters, refer to Chapter 58 in the *Reference for NETBuilder Family Software*. For information on interpreting the statistics displays, refer to Appendix H.

Troubleshooting the Configuration

OSI routing can be difficult to troubleshoot if there is a problem. This section describes some common misconfiguration problems and the basic tools (OPING, OTraceRoute, and TELnet) to solve them.

Incomplete Level 2 Backbone

Figure 16-1 shows an incomplete Level 2 backbone problem that may occur when a transit area has Level 1 routers disrupting the Level 2 path. Area B has broken the Level 2 backbone. The Level 2 information from Area A cannot be distributed to Area C, and Area A has no Level 2 information from Area C.

The solution to this problem routers B1 and B2 as Level 2 routers with the ISIS parameter mode.

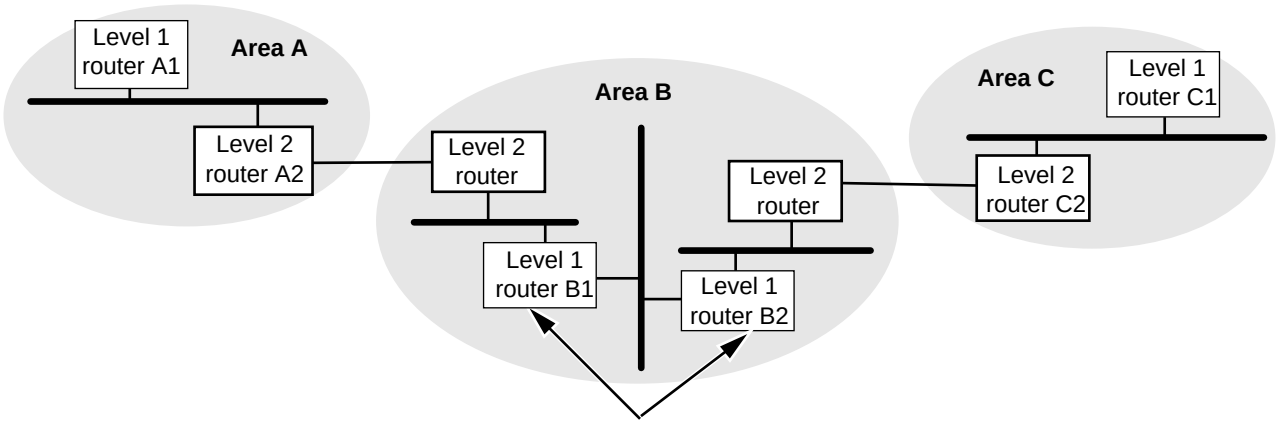


Figure 16-1 Completing the Level 2 Backbone

Partitioned Area

A partitioned area may occur if multiple routers with the same AreaAddress exist on the network and there is no intra-area route between each pair of these routers. Communication within one partition may succeed, but communication outside the partition may exhibit connectivity problems. Packets that originated from a partition and sent to another area may be delivered without any problem, but packets destined to a system within a partitioned area may be forwarded to the wrong partition.

Another symptom of this problem occurs when some return packets are received and others are not received. This situation exists if multiple routes exist to the partitioned area, and some routes route packets to one partition, while other routes route packets to other partitions.

An area may become partitioned when a link goes down within the area, segmenting the area completely, even though both partitions may still be connected through a Level 2 path through the neighboring areas.

If you suspect a partition, examine for consistency the Level 2 Link State Data for the Domain. Each router indicates its set of area addresses in the Link State PDU identified by the SystemID of the router, followed by the value 00:00.

For example, enter:

```
SHow -ISIS LinkStateData 080002033ABB:00:00
```

The following display appears:

```
----------ISIS Level 2 Link State Database, Checksum Sum(0008A143)------------
LSP-ID             sequence   remaining   P bit  H bit  attach     IS type  data      checksum
                   number     lifetime                  bit                 length
080002033Abb:00:00  3231       1110        0      0      1          L2       90        17AE(OK)
Area Addresses ==>/47/0004/00351000
IS neighbors ==> (20 .. .. ..) 080002033CC9:01
IS neighbors ==> (20 .. .. ..) 080002013C37:01
IS neighbors ==> (20 .. .. ..) 080002033ABB:04
IS neighbors ==> (20 .. .. ..) 080002033ABB:05
IS neighbors ==> (20 .. .. ..) 080002033ABB:06
IS neighbors ==> (20 .. .. ..) 080002033ABB:07
IS neighbors ==> (20 .. .. ..) 080002033ABB:08
```

Multiple Area Addresses

If the routers within an area have more than three area addresses configured, that area may become partitioned. If the extra area addresses have different values, then the algorithm for eliminating extra area addresses may arrive at a different set in different parts of the area and eliminate the most important area addresses. (The algorithm is purely numerical and has no other basis for arriving at the set of three area addresses.)

If you are using more than one AreaAddress for a single area, you can avoid partitioning by configuring the same set of area addresses for every router in the area.

Mismatched Passwords

The IS-IS Protocol has three types of passwords: the interface password (HelloPassWord), the area password (L1PassWord), and the domain password (L2PassWord).

If two routers attached to the same network (LAN or point-to-point) do not have the same HelloPassWord, they will not bring up the adjacency. If you use the L1PassWord to protect against unmanaged routers from becoming attached to your network, then all routers in the same area must be configured with the same password. For the L2PassWord, all Level 2 routers, regardless of the area in which they reside, must have the L2PassWord parameter configured to the same string.

If mismatched passwords exist, refer to the IS-IS statistics. These statistics show the port on which mismatched passwords occur and the type of failure (Hello, Level 1 or Level 2). The statistics do not identify the misconfigured system; however, you can find the system by examining the IS-IS Adjacency Table and the Link State Database Table.

Customizing the OSI Router

To change the level of routing and configure passwords, follow these steps:

1. Determine the level of routing to be used on each port that is enabled for ISIS routing.

   Both Level 1 and Level 2 routing are enabled by default. To change level of routing, if necessary, use:

   ```
   SETDefault !<port> -ISIS CONTrol = L2Only
   ```

   The L2Only value under the CONTrol parameter is effective only if the MODE parameter is set to Level2. For complete information on the MODE parameter, refer to Chapter 32 in *Reference for NETBuilder Family Software*.

   For information on Level 1 and Level 2 routing, refer to "Level 1 Routing" on page 16-12 and "Level 2 Routing" on page 16-13.

2. Determine whether the port is connected to a stub or transit network. If there are no other routers on a port (for example, a boundary router port), then configure the port as a stub network using:

   ```
   SETDefault !<port> -ISIS CONTrol = Stub
   ```

3. Configure area passwords for all Level 1 routers in the same area.

To configure an area password for the router, use:

```
SETDefault -ISIS L1PassWord = "<password (1–16 characters)>"
```

The Level 1 password prevents routers in another area from accidentally merging into this area.

4 Configure passwords for all Level 2 routers using:

```
SETDefault L2PassWord = "<password (1–16 characters)>"
```

Configuring a password prevents other routing domains from learning topology information about this routing domain. It also prevents two routing domains from being accidentally merged into one.

How the OSI Router Works

This section describes the concepts involved in OSI routing activities.

OSI Network Topology

An OSI internetwork is divided into multiple routing domains. The IS-IS routing protocol operates within a routing domain; therefore, it is known as an intra-domain routing protocol.

Inside a routing domain, an OSI network is further partitioned into a two-level hierarchy.

The lower level is called an area. A subset of the IS-IS routing protocol, Level 1, operates within an area. Therefore, Level 1 routing is also known as intra-area routing. The Level 1 routing protocol learns the complete topology in its home area; it does not learn the topology outside of its home area, except for area border routers that can reach other areas. For more information on areas and the Level 1 routing protocol, refer to "Areas" on page 16-12.

The upper level is called the Level 2 subdomain, which consists of Level 2 routers that connect the areas that make up an OSI routing domain. Another subset of the IS-IS routing protocol, Level 2, operates within the Level 2 subdomain. The Level 2 routing protocol learns the complete topology of the Level 2 subdomain and all areas that it can reach. However, it does not learn the topology of any specific area.

Figure 16-2 shows the topology of a typical OSI internetwork.

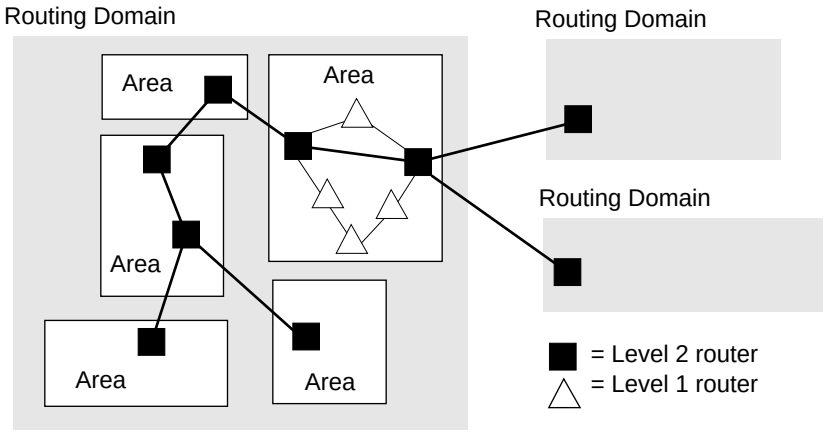


Figure 16-2 Typical OSI Internetwork Topology

Area Addresses

The standard for structure and assignment of NSAP addresses is ISO 8348. It defines three fields: the authority and format identifier (AFI), the initial domain identifier (IDI), and the Domain Specific Part (DSP). This structure is useful for creating procedures for assigning unique network service access point (NSAP) addresses, but it is not useful for intradomain routing purposes.

The standard for intradomain routing, IS 10589, views any NSAP address as containing three parts: an area address, a system ID, and an N-selector. The area address identifies an area within the routing domain. The system ID identifies an ES in the area. The N-selector is used by the ES to distinguish between multiple users of the Connectionless Network Service (CLNS), which on the bridge/router includes ISO Transport Class 4 (TP4) and TCP. This structure of the NSAP address is overlaid on the structure of any standard NSAP address as defined by ISO 8348.

Figure 16-3 shows both structures of the NSAP address.

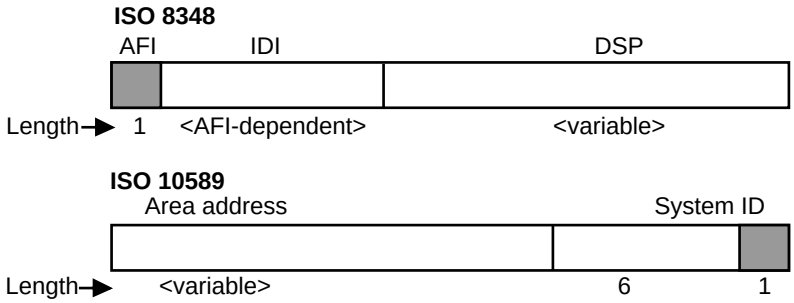


Figure 16-3 NSAP Address Structures

The following are examples of area addresses.

Example 1 Suppose your router has the following NSAP address:

/47/0004/0035110008000201345601

The area address is /47/0004/00351100; the ID is 080002013456; the selector is 01.

Example 2 Suppose your router has the following NSAP address:

/49/005308000201345601

The area address is /49/0053; the ID is 080002013456; the selector is 01.

The following area address guidelines must be considered when you set up your OSI network:

- Each router must have at least one area address before IS-IS routing can take place. You can use the 3Com default area address or configure your own. A router can be configured with a maximum of three area addresses.
- Each area should have a globally unique address associated with it. That is, a given area address should be associated with only one area.
- All systems with a given area address must be located in the same area.

Determining Your Own Area Address. All bridge/routers shipped from 3Com have the default area address of /49/0053. (In this area address, there is no IDI part for AFI value 49.)

For networks that are not going to be interconnected with other routing domains, you can use the default AFI value 49. The DSP prefix, 0053, can be reassigned with a new value for each different area. For networks such as these, there is room for 65,536 area addresses.

However, 3Com recommends that each installation acquire its own NSAP address block from a registration authority and manage the area addresses from that block. For information on registration authorities and how you can obtain registration information, refer to Appendix E.

To set an area address for each router, use the AreaAddress parameter. For complete information on this parameter, refer to Chapter 32 in *Reference for NETBuilder Family Software*.

ID and Selector Values

The ID value is a six-octet field in the NSAP address, as specified in U.S. GOSIP Version 2 DSP format. Because there may be different implementations (which would be incompatible with this implementation) that support different sizes of ID fields, you must ensure that all ISs and ESs use the same ID length within a routing domain.

For all ISs shipped from 3Com, the ID value is automatically extracted from the media access control (MAC) address of the first LAN interface at boot time. You can change this default using the -ISIS SystemID parameter.

The selector is the last octet in the NSAP address. It is used primarily for selecting the transport entity that is to receive a packet. This field is ignored by the IS-IS routing protocol.

Network Entity Title

A router can have multiple area addresses, but it can have only one Network Entity Title (NET). The NET of an IS is computed automatically at boot time. (No user configuration is required.) The NET is computed by taking the area address of the IS and appending the ID value of the IS to it. The selector part is always 00 (refer to Figure 16-2).

To display the NET for a particular IS, enter:

```
SHow -CLNP NetEntityTitle
```

The NET is used primarily for ES-IS and CLNP Protocol operations. Specifically, the NET is used as follows:

- In ISH packets for announcing a router's presence and availability to ESs
- When a router issues CLNP error and redirect protocol data units (PDUs)
- In SourceRoute and RecordRoute options within a CLNP PDU

If there is more than one area address for a router, the NET is computed from the area address made up of the lowest numbers. Since the value of the NET depends on the value of a router's area address, the value of the NET automatically changes to reflect changes to a router's area addresses.

Areas An area is a group of directly interconnected ISs and ESs with the same area address. An area can include anywhere from a single IS up to 100 ISs and up to 2,000 ESs. However, routing in smaller areas runs smoother and more reliably.

3Com recommends that ISs and ESs be grouped into areas based on the following criteria:

- **Departmental function** — for example, manufacturing, engineering, and MIS
- **Administrative or geographic boundaries** — for example, a department, building, campus, or company
- **Level of traffic** — for example, where traffic is heavy and localized, such as a group of workstations and their file servers
- **Reliability of traffic** — for example, where traffic is unreliable and prone to errors, such as a test lab

Level 1 Routing

A router configured as a Level 1 router learns routes from other Level 1 routers within the same area. Each router sends out a hello packet to other routers on directly attached subnets. These hello packets contain the area addresses of the router that is sending the packet. By comparing received hello packets, routers may decide that they belong to the same area. These routers then form adjacencies with each other, or they can reject forming an adjacency if there are no area addresses in common. If different hello passwords are defined, the routers will not become adjacent.

The boundary of an area is learned dynamically.

Figure 16-4 shows a network made up of two areas. In this figure, the area address for a router is enclosed in square brackets ([ ]), and the area address for an area is shown in angle brackets (< >).

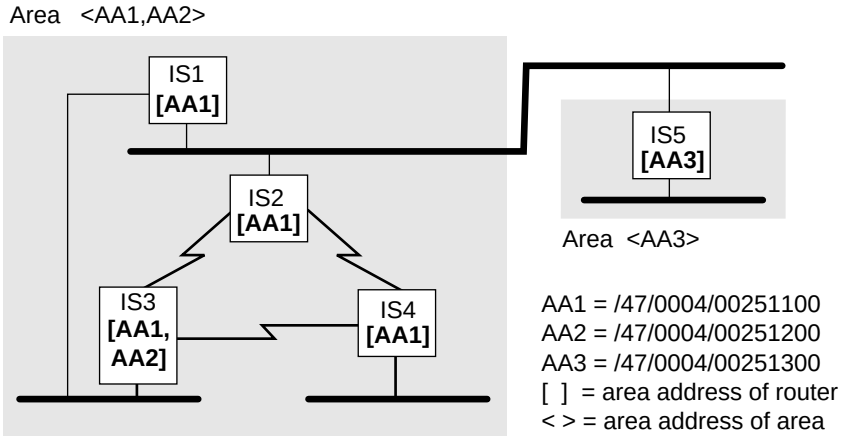


Figure 16-4 Network Made Up of Two Areas

In Figure 16-4, IS1, IS2, IS3, and IS4 form an area, because they all share the common area addresses ([AA1]). IS5 forms an area of its own (<AA3>). All routers belonging to the same area must be directly interconnected through physical paths. From any router, it should be possible to reach any other router in the same area through intra-area routes (by going through other routers belonging to the same area).

Not all directly connected routers belong to the same area. For example, IS1 and IS5 do not share an identical area address; therefore, they form two distinct areas. These two areas reside on the same subnet.

Once adjacencies are formed and areas are determined, the adjacent routers within an area exchange routing information.

Level 1 Routing Table

To display the Level 1 Routing Table, enter:

```
SHow -ISIS L1Route
```

The following display is an example of a Level 1 Routing Table:

```
Time since last table update: 554 sec. Update count: 467
-----System ID----  ----Metric--  -----Port----  -------SNPA/IS-----
*080002A034A3       60            2              IS 080002A014AB
#0800020184A8       0             -              -
*080002A014AB       20            2              IS 080002A014AB
*080002A01123       40            2              IS 080002A014AB
080002000A8F0       80            2              IS 080002A014AB
080002001312F       20            2              IS 080002A014AB
* indicates an IS # indicates the nearest L2 IS
```

Entries in the Level 1 Routing Table include the following types of information:

- System ID

  The Level 1 Routing Table displays all reachable systems within an area. A system can be an ES, an IS (identified by an asterisk [*]), or the closest Level 2 IS (identified by the pound sign [#]).

- Metric

  The Level 1 Routing Table displays the total cost associated with reaching a system within an area.

- Port

  The Level 1 Routing Table displays the port number of the router through which the destination is reachable. A hyphen (-) indicates that the system is the router itself.

- SNPA/IS

  If a destination is directly attached, the Level 1 Routing Table displays the MAC address of the system (identified by SNPA). If a destination is not directly attached, it displays the system ID of the next hop IS, which is one step closer to the destination (identified by IS).

Level 2 Routing

A router configured as a Level 2 router performs the following functions:

- It runs the Level 2 protocol with other Level 2 routers. It learns routes to other areas from other Level 2 routers throughout the Level 2 backbone.
- It continues to run the Level 1 protocol in its home area, and it learns routes from other Level 1 routers.

Since the Level 2 protocol runs in parallel with the Level 1 protocol, they do not interfere with each other. As a result, a Level 2 router continues to serve the intra-area traffic for its home area. A nearby ES should not notice a difference in the behavior of this router.

The primary purpose of a Level 2 router is to interconnect disjointed areas into one single routing domain, thus establishing connectivity between areas.

At least one router from each area is selected and configured as a Level 2 router. For example, in Figure 16-5, if IS2 is chosen to be a Level 2 router for Area <AA1, AA2>, and IS5 is chosen to be a Level 2 router for Area <AA3>, the resulting Level 2 backbone is shown (as indicated by the shadows on the boxes containing IS2 and IS5).

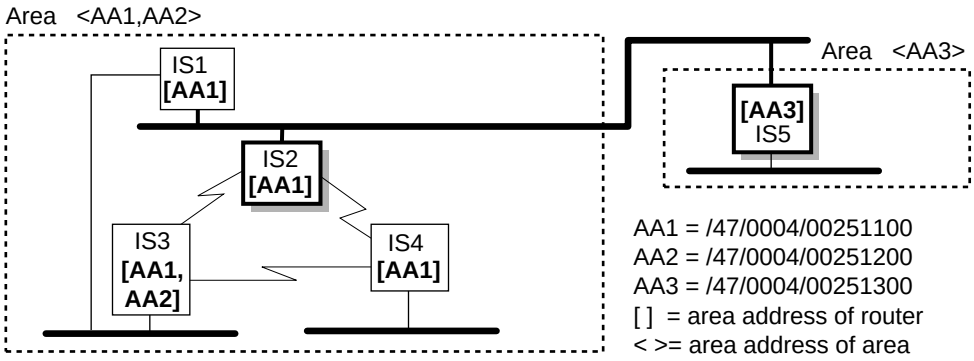


Figure 16-5 Level 2 Backbone with One Level 2 Router in Each Area

Each Level 2 router belongs to its home area. The Level 2 router summarizes the area address(es) of its home area and announces it to all other Level 2 routers on the Level 2 backbone.

In Figure 16-5, IS2 announces that it can reach all hosts with Area Addresses AA1 and AA2. With this information, IS5 knows how to reach those hosts. If IS2 had not been configured as a Level 2 router, there would be no way for IS5 to learn the location of Area <AA1, AA2> even though it is directly attached on the same subnet.

You can configure more than one Level 2 router in each area. Figure 16-6 shows the same topology as in Figure 16-5 except that IS1 is also configured as a Level 2 router (as indicated by the shadows on the boxes containing IS1, IS2, and IS5).

In Figure 16-6, Area <AA1, AA2> now has two Level 2 routers, IS1 and IS2, bordering the Level 2 backbone. If one of these routers fails, the other can continue to serve interarea traffic. From the viewpoint of IS5, if it wants to deliver a PDU to Area <AA1, AA2>, it can select either IS1 or IS2. In fact, it can split the load between the two routers. It does not know which router reaches an ES. The detailed topology information within an area is hidden from the Level 2 backbone. All IS5 knows about Area <AA1, AA2> is that it has two area addresses and both IS1 and IS2 can reach it.

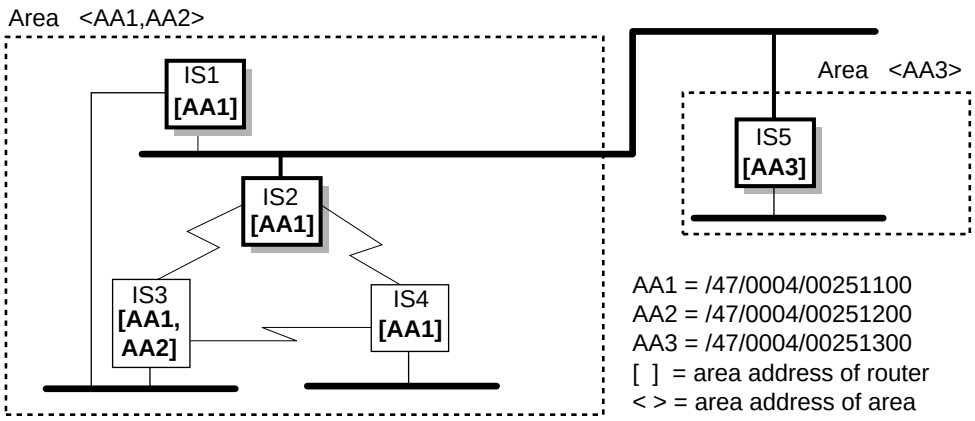


Figure 16-6 Level 2 Backbone with Multiple Level 2 Routers in One Area

All Level 2 routers must be physically interconnected. If a Level 2 router goes down, then the area represented by this router is no longer reachable. A Level 2 backbone should have sufficient redundancy so that the failure of one router or one link does not isolate any area.

Level 2 Routing Table

To display the Level 2 Routing Table, enter:

```
SHow -ISIS L2Route
```

The following display is an example of a Level 2 Routing Table:

```
Time since last table update: 587 sec. Update count: 473
------Area Address----  ---Metric--  ----Port---  -------IS------
/47/0004/00351100       20           1            080002A014AB
/47/0004/00351200       20           2            080002A00949
/47/0004/00351300       0            -            -
/47/0004/00352000       20           2            080002A049B9
```

Entries in the Level 2 Routing Table include the following information:

- Reachable areas
- The Level 2 Routing Table displays all reachable areas within a routing domain.
- Metric
- The Level 2 Routing Table displays the total cost associated with reaching another area within a routing domain. The metric displayed in this table is the total cost of reaching an area border router only. Additional costs may be incurred when traveling from the area border router to the final destination within the area.
- Port
- The Level 2 Routing Table displays the port number of the router through which the next hop IS is reachable. A hyphen indicates it is the router's home area.
- IS

  The Level 2 Routing Table identifies the next IS that would need to be traversed to reach the destination area. The table displays the system ID of the IS, not the MAC address.

Transit and Leaf Areas

A single leaf area is an area that receives traffic only for itself; it needs only one Level 2 router at the point where it is attached to the neighbor area. As shown in Figure 16-7, traffic enters each single leaf area and stops in that area.

In contrast, a transit area is an area that receives traffic for both itself and for other areas; it needs Level 2 routers in order to complete the Level 2 backbone. As shown in Figure 16-7, traffic enters the transit area and can be further routed to the single leaf areas; a transit area interconnects other areas. A leaf area needs only one Level 2 router at the point of attachment to the neighbor area. In order to complete the backbone, a transit area must contain a *path* of Level 2 routers (refer to the path of three Level 2 routers within the Transit Area in Figure 16-1).

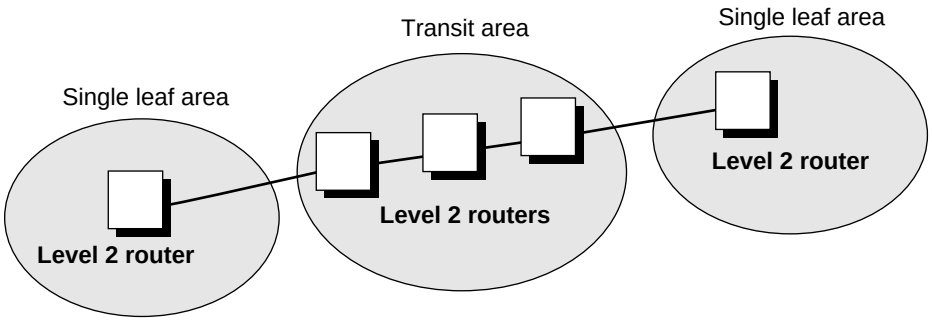


Figure 16-7 Single Leaf and Transit Areas

You must configure the same Level 2 password for each Level 2 router in the same domain.

Metrics and Route Selection

The router running the IS-IS routing protocol selects the path with the lowest total cost to reach its destination. In this case, cost is a user-defined value that measures the capacity of a particular port. A higher value (for example, 50) indicates a higher cost (or a lower capacity). Conversely, a lower value (for example, 10) indicates a lower cost (or a higher capacity).

The total cost to a particular destination is computed by adding the costs of all links toward the destination.

Imagine that there are two routes to a particular destination. Route 1 has a total cost of 100 associated with it; Route 2 has a total cost of 115 associated with it. The router running the IS-IS Protocol will select Route 1, because it has the lowest total cost associated with it.

By default, the cost on all ports has been set to 20, regardless of the underlying network type or speed. These cost values should be adjusted according to your particular situation. For example, a 10 Mbps LAN is preferable to a 64 kbps serial line. In this case, you can set a low cost for the LAN and a higher cost for the serial line.

The L1DefaultMetric and L2DefaultMetric parameters allow you to define the cost associated with using a particular port. For complete information on these parameters, refer to Chapter 32 in *Reference for NETBuilder Family Software*.

Multipath Routing and Load Splitting

Your topology may contain multiple paths with equal minimum costs toward the same destination.

The OSI router supports multipath routing. It can compute up to four paths toward any destination.

Specifically, for intra-area routing, a Level 1 router can compute multiple paths toward all ESs within the area and toward the closest Level 2 router. If there are multiple Level 2 routers with the same minimum cost, one is randomly selected. For interarea routing, a Level 2 router computes multiple paths toward any area. For interdomain routing, a Level 2 router computes multiple paths toward any domain border router that advertises the longest matching address prefix.

After computing multiple paths toward a destination, the router can then perform load splitting. The router splits the traffic load between the paths on a round-robin basis. Load splitting helps prevent some network segments from being heavily congested, while others are underutilized.

End System Table

An End System Table consists of both dynamic and static entries. The router learns the presence of an ES on the network from the end system hello (ESH) packets. You can also modify the table by adding or deleting ESs.

The following display is example of the End System Table displayed by the SHow -CLNP ES command:

```
EndSystem                                SNPA             Interface
/47/0004/001E000108000200E10E01          %08000200E10E    2
```

Intermediate System Table

An Intermediate System Table consists of only dynamic entries. The router learns the presence of an IS on the network from the ISH packets. To ensure that the router properly learns the ISs on the network, it is recommended that you not change the default MulticastES and MulticastIS parameter values.

The following is an example of the Intermediate System Table displayed by the SHow -CLNP IS command:

```
Intermediate System Network Entity Title  SNPA            Interface
/47/0004/001E000108000200999900           %080002009999   2
/49/0053080002A00B7900                    %080002A00B79   1
```

User Configurations

Table 16-2 shows how to change the way the router learns about the network through the ES-IS Protocol. It includes only the parameters that have not been discussed in previous sections. For more information on parameters available in the ESIS Service, refer to Chapter 21 in *Reference for NETBuilder Family Software*.

Table 16-2 Configuring the ESIS Parameters

Parameter	Result
CONTrol:	
ChecKSum | NoChecKSum	Determines whether checksum is used for the ISH PDUs and ESH PDUs.
FastConfig | NoFastConfig	Determines how fast the router learns about its neighbors.
UpdateTime	Determines the interval at which the router sends out ISH PDUs.
HoldTime	Determines the value of the hold-time field in the ISH PDUs and specifies how long the recipient of the ISH PDUs remembers them. 3Com recommends that it be set to greater than twice the value of UpdateTime.

Table 16-3 shows how to change the way the router learns about the network through the IS-IS Protocol. It includes only the parameters that have not been discussed in previous sections. For more information on parameters available in the ISIS Service, refer to Chapter 32 in *Reference for NETBuilder Family Software*.

Table 16-3 Configuring the ISIS Parameters

Parameter	Result
CsnpTime	Sets frequency at which Complete Sequence Numbers PDUs are transmitted.
DISHelloTime	Sets frequency at which hello packets are transmitted by a designated IS.
HelloTime	Sets frequency at which hello packets are transmitted by an IS.
L1BufferSize	Determines maximum size of Level 1 routing packets sent by an IS.
L2BufferSize	Determines maximum size of Level 2 routing packets sent by an IS.
L1Multicast	Sets the multicast address that all Level 1 ISs on an Ethernet should transmit hello and routing packets to.
L2Multicast	Sets the multicast address that all Level 2 ISs on an Ethernet should transmit hello and routing packets to.
LspBroadcastTime	Sets maximum frequency at which routing packets are transmitted on a broadcast network.
LspMAxTime	Sets the maximum interval between regenerations of Link State PDUs.
LspMInTime	Sets minimum interval between event-driven regenerations of Link State PDUs.
LspRtxTime	Sets interval between retransmissions of an update on a point-to-point link.
PsnpTime	Sets frequency at which Partial Sequence Numbers PDUs are transmitted.

Table 16-4 shows parameters in the CLNP Service that allow you to customize the configuration of your OSI router. It includes only the parameters that have not been discussed in previous sections. For more information on parameters available in the CLNP Service, refer to Chapter 16 in *Reference for NETBuilder Family Software*.

Table 16-4 Configuring the CLNP Parameters

Parameter	Result
RDgeneration	Determines the frequency at which redirect packets (RD PDUs) are originated by the router.
ERgeneration	Determines the frequency at which error packets (ER PDUs) are originated by the router.

Setting Up Interdomain Routing

A routing domain is usually a single administrative domain, such as a company or a university that runs a single compatible intradomain routing protocol (such as IS-IS). It is composed of one or more (up to several hundred) areas. The areas within the routing domain are interconnected by Level 2 routers. These Level 2 routers make up the Level 2 subdomain, or backbone, within this particular routing domain.

Prerequisites

Because interdomain routers are not dynamically learned, you must set up static routes between neighboring Level 2 routers in each routing domain. These routes then are distributed to the other Level 2 routers in the domain through the IS-IS Protocol.

Configure all area addresses with the same initial string of digits, or routing domain Identifier. This string of digits can be used by another domain to create a route to the domain it identifies. A domain built with more than one format of NSAP address results in multiple entries for the various formats. Interdomain routing is based on the longest matching prefix. If a packet contains no matching prefixes in a destination NSAP address, a zero-length default route that matches all NSAP addresses may be used.

Procedure

For an example of setting up static routes between routing domains, see Figure 16-8.

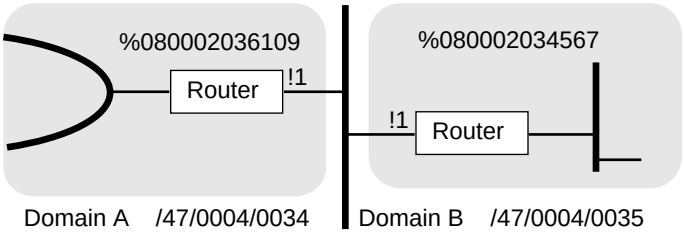


Figure 16-8 Two Routing Domains on a Common Ethernet

To set up interdomain routing, follow these steps:

1. Isolate the routing domains on each router on the common subnet of the adjoining domains by using the ISIS HelloPassWord commands:

   For example, on the border router of domain A, enter:

   ```
   SETDefault !1 -ISIS HelloPassWord = “Domain-A”
   ```

   On the border router of domain B, enter:

   ```
   SETDefault !1 -ISIS HelloPassWord = “Domain-B”
   ```

2. Configure the PrefixRoute for domain A on the border router from domain B.

   For example, on the border router of domain A, enter:

   ```
   ADD !1 -ISIS PrefixRoute /47/0004/0035 %080002034567
   ```

3. Configure the PrefixRoute for domain B on the border router from domain A.

   For example, on the border router of domain B, enter:

   ```
   ADD !1 -ISIS PrefixRoute /47/0004/0034 %080002036109
   ```

4. Configure a default route on the border router of domain A to forward to domain B if a matching prefix route cannot be found.

   For example, on the border router of domain A, enter:

   ```
   ADD !1 -ISIS PrefixRoute Default %080002034567
   ```

Related Information

Figure 16-9 illustrates a sample scenario for configuring address prefixes for routing toward other routing domains. Suppose port 1 on IS1 in Routing domain 1 interfaces port 1 on IS2 in Routing domain 2. To set up a static route from routing domain 1 to routing domain 2, enter the following command on IS1 in routing domain 1:

```
ADD !1 -ISIS PrefixRoute /47/0004/0045 %0800002013456
```

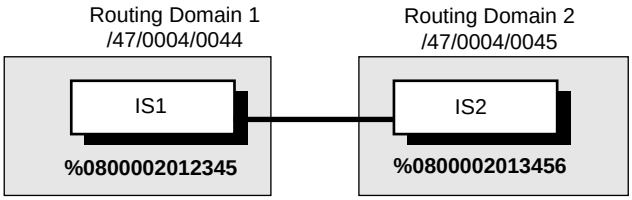


Figure 16-9 Setting Up Interdomain Routing

Conversely, to set up a static route from routing domain 2 to routing domain 1, enter the following command on port 1 of IS2 in routing domain 2:

```
ADD !1 -ISIS PrefixRoute /47/0004/0044 %0800002012345
```

In the above example, note the following:

- Both IS1 and IS2 must be Level 2 routers.
- /47/0004/0044 is the address prefix that summarizes routing domain 1; /47/0004/0045 summarizes routing domain 2. All systems in each domain should have addresses falling under their respective prefixes.
- The link between IS1 and IS2 is a LAN link; therefore, a remote MAC address was specified. You must specify a remote DTE for X.25, an SMDS address for SMDS, or a DLCI number for Frame Relay links.
- If routing domain 2 is a national or regional backbone (meaning that it serves to interconnect many routing domains), it is more appropriate to specify a default route. To set up a default route, for example, on port 1 of IS2, enter:

  ```
  ADD !1 -ISIS PrefixRoute Default %080002013456
  ```

After you specify the static routes, this information is propagated throughout the Level 2 backbone within the routing domain. All Level 2 routers learn the set of reachable address prefixes and which router can be used to reach that address. If two routers can reach the same address prefix (for example, there are two domain border routers connecting the same external domain), a router selects the domain border router that is closest to it. In this situation, you may also want to configure these routers to perform load splitting. For more information on load splitting, refer to "Multipath Routing and Load Splitting" on page 16-17.

Address Prefixes. An address prefix is some number of leading digits of a full NSAP address. It can be as few as two digits or as long as a full NSAP address, whatever is required to uniquely identify another routing domain.

An address prefix points a packet that is being routed between routing domains toward the desired routing domain.

Example Suppose that the Acme Company has been assigned the following NSAP address prefix by the appropriate authority:

/47/0004/0025XXXX

The XXXX field is left for you (the network manager) to assign. You can assume that all hosts with NSAP address prefix /47/0004/0025 reside within the Acme Company and that all hosts within the company have that identical prefix.

Assuming that the Acme Company is a single routing domain, then the routing domain can be categorized by address prefix /47/0004/0025.

An address prefix has the following characteristics:

- It can contain an odd number of digits (semi-octets). Examples include /47/0, /47/000, and /47/0004/0. All ranges of AFI values, including both binary and decimal syntaxes, are supported. However, AFI values 50 and 51 are not supported.
- Longer address prefixes take higher precedence over shorter ones. For example, an NSAP address may match multiple address prefixes, such as shown here:
  - /47
  - /47/0004
  - /47/0004/0035
- Since /47/0004/0035 is the longest address, it is chosen.
- Default routes always have the lowest precedence. They match to all NSAP addresses.

Sometimes a routing domain cannot be assigned a single address prefix. (The routing domain may have been allocated multiple NSAP addresses from different authorities.) Therefore, you must set the -ISIS PrefixRoute parameter for each NSAP address type. The following example illustrates how to set the PrefixRoute parameter for different NSAP address types.

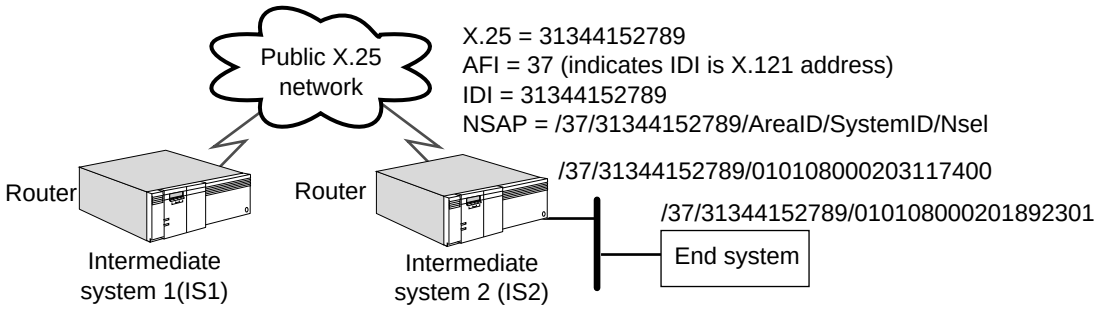
Example Suppose that the AAA Company has merged with the BBB Company to form the CCC Company. Both the AAA and BBB Companies already had large OSI networks. The AAA Company's network used an NSAP addressing scheme based on the U.S. GOSIP version 2 format, while the BBB Company's network used an ANSI-based addressing scheme. If another party (such as a company) needs to communicate with the CCC Company, it must configure its domain border routers with the two address prefixes originally used by the AAA and BBB companies. The following are examples of commands that you can enter on a domain border router to configure it to communicate with the CCC Company over an X.25 PDN:

```
ADD !3 -ISIS PrefixRoute /39/840/543621 #030354321982608
ADD !3 -ISIS PrefixRoute /47/0005/016A9F72 #031354321982608
```

Static Interdomain Routing. If your OSI network is attached to a public X.25 or SMDS network, the address of the public network that identifies your router as a node on the public network also identifies you as an addressing authority according to the standard ISO 8348. Special AFIs for NSAP addresses are formed using the public network address as the IDI, and you can structure the DSP in any way, provided that the last seven octets are the system ID and the N-selector.

If this is the case, and multiple sites are interconnected using the same method for their NSAP address assignments (reachable directly over the same public network), then interdomain routing can be accomplished using an algorithm. The algorithm used extracts the public address from the NSAP address and forwards the CLNP packet on the public network using that extracted address as the SNPA of the next hop router.

This form of address and method of determining the next-hop media address across the public network lets you build extremely large OSI networks.



IS1 can route to any system in the domain of IS2 if you enter:

```
ADD !3 -IS PrefixRoute /37 ALGORITHM
```

Figure 16-10 Domain Addressing Based on X.25 Attachment Address

Interdomain Routing Table. There are two forms of the Interdomain Routing Table.

The first form displays a collective listing of all routing domains that can be reached from a particular routing domain. To display this routing table, enter:

```
SHow -ISIS PrefixRoute
```

The following display shows a typical Interdomain Routing Table:

```
-NSAPAddress Prefix-  --Metric--  --Port--  --------IS/SNPA------
/47/0004/00352        20          1         SNPA %0800002A00AB6
/47/0004/0035         20          1         SNPA %0800002A00B92
/47/0004              20          1         SNPA %0800002013C37
Default               20          2         IS %0800002019876
```

Entries in the Interdomain Routing Table include the following information:

- NSAP address prefix

  This routing table displays reachable NSAP address prefixes.

- Metric

  This routing table displays the total cost associated with reaching a particular routing domain. The metric displayed in this table is the total cost of reaching a domain border router only. Additional costs may be incurred when traveling from the domain border router to the final destination.

- Port

- This routing table displays the port number of the router through which the destination routing domain is reachable.

- Next Hop (IS/SNPA)

  If the external domain is directly reachable, this routing table displays the MAC, SMDS, or data terminal equipment (DTE) address (or data link connection identifier (DLCI)) that can be used to reach this domain (identified by SNPA). This information is displayed when the router itself is the domain border router and has been configured with address prefix information. Otherwise, the routing table displays the next hop IS, which is one step closer to the domain border router that has been configured with address prefix information (identified by IS).

The second form displays the static routes you configured on a particular router. To display these static routes, enter:

```
SHowDefault -ISIS PrefixRoute
```

The following display is a table of the static routes:

```
---NSAPAddress Prefix-----  --Port--  -----SNPA----     ---Status--
/47/0004/00352              1         %0800002A00AB6    Active
/47/0004/0035               1         %0800002A00B92    Active
/47/0004                    1         %0800002A13C37    Active
```

Entries in this form of the Interdomain Routing Table include the following types of information:

- NSAP (Network Service Access Point) address prefix

  This routing table displays reachable NSAP address prefixes.

- Port

  This routing table displays the port number of the router through which the destination routing domain is reachable.

- SNPA (Subnetwork Point of Attachment)

  This routing table displays the MAC, SMDS, or DTE address (or DLCI) that can be used to reach an external domain.

- STATUS

  This routing table displays the status of the NSAP address prefix. An "active" status indicates that the address prefix is operational. An "idle" state indicates that the address prefix is not in service. The address prefix may be in the idle state if the port associated with the prefix is down, the router has not been configured to perform Level 2 routing, or the SNPA syntax is rejected by the lower layers (for example, a DTE address may be specified on a Frame Relay port or a MAC address may be specified on an X.25 port).

Integrated IS-IS for IP and Dual IP/OSI Mode

Integrated IS-IS is a protocol that provides integrated OSI-type routing for IP and OSI environments. It is the IP extension added to the original OSI IS-IS Protocol. Integrated IS-IS routing simplifies network topology, reduces network management complexity, and reduces routing traffic overhead.

To configure Integrated IS-IS for IP and dual IP/OSI environments, refer to Chapter 6.

17 CONFIGURING VINES ROUTING

This chapter describes the procedures for configuring your system to perform VINES IP routing. It also describes how the router works and gives guidelines for operating, managing, and troubleshooting the router.

For conceptual information, refer to "How the VINES Router Works" on page 17-5.

Setting Up a Basic VINES Router

VINES network numbers and addresses are not user-configurable in the same way that other routing protocols are. The router automatically assigns its own VINES network address, enabling the router to communicate with VINES servers once VINES routing is enabled. This VINES network address is 32 bits long and consists of two parts. The first part of the network number is a specific vendor code that Banyan Systems has reserved for 3Com. This vendor code, which starts with hex 302 or hex 303, is composed of the 11 most significant bits of the 32-bit network address. The remaining 21 bits of the network number contain the 21 least significant bits of the router MAC address.

Configuring for Local Area Networks and Point-to-Point Protocol Links

The procedure in this section explains how to enable VINES routing and set up parameters on LAN ports and Point-to-Point Protocol (PPP) links where no VINES servers are available.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to Chapter 1.

Procedure

To configure the system to perform basic VINES routing, follow these steps:

1. Enable VINES routing using:

   ```
   SETDefault !<port> -VIP CONTrol = Route
   ```

2. Enable ARP on those ports where no VINES servers are available using:

   ```
   SETDefault !<port> -VIP CONTrol = Arp
   ```

   The specified port will now respond to Address Resolution Protocol (ARP) query and ARP assignment requests.

3. Forward VINES broadcast packets when the nearest VINES server is more than one hop away from a VINES client using:

   ```
   SETDefault !<port> -VIP CONTrol = NoServer
   ```

4 Select the packet encapsulation format for each Ethernet interface using:

```
SETDefault !<port> -VIP HeaderFormat = [Ethernet | Ieee | Snap]
```

and specifying either Ethernet, Ieee, or Snap.

For more information, refer to "HeaderFormat" on page 63-3 in *Reference for NETBuilder Family Software*.

5 Control whether the router forwards broadcast packets over a port where packet charges are enforced using:

```
SETDefault !<port> -VIP CONTrol = PktChrge
```

This value prevents the router from forwarding broadcast packets received from other reachable nodes and servers, unless the class subfield bit in the Transport Control field is set appropriately.

6 Verify the VINES configuration by entering:

```
SHow -VIP CONFiguration
```

The router displays the configuration information for active VINES ports only. If there is no active port, it prompts you to enable VINES routing. To display the default configuration, enter:

```
SHow !* -VIP CONFiguration
```

To complete the procedure for PPP links, refer to Chapter 34.

Configuring for Wide Area Networks

You can configure the VINES router to perform routing over wide area network ports using Point-to-Point Protocol/Phone Line Gateway (PPP/PLG), Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), Switched Multimegabit Data Service (SMDS), X.25, and Integrated Services Digital Network (ISDN).

Routing VINES over Frame Relay, ATM DXI, and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to route VINES over Frame Relay, ATM DXI, or X.25 in a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. For complete information on configuring VINES routing over Frame Relay, ATM DXI, or X.25, including a discussion on fully meshed, partially meshed, and nonmeshed topologies and virtual ports, refer to Chapter 42, Chapter 43, and Chapter 45, respectively. For information on the number of virtual ports supported per platform, refer to "Virtual Ports" on page 1-3.

Routing VINES over SMDS is supported over fully meshed and hierarchical partially meshed topologies (where virtual ports are configured to attach to distinct groups of fully meshed devices). To configure your VINES router to perform routing over SMDS, refer to Chapter 44.

To configure your VINES router to perform routing over PPP or PLG, refer to Chapter 34. For information on wide area networking using ISDN, refer to Chapter 35.

Verifying the Configuration

After you have configured the basic VINES router, check to see whether it can route packets properly. Examine the VINES routing and neighbor tables, and send packets from one network to another to see if they are properly forwarded.

Verifying Procedure

Before you use the router for interconnecting networks, follow these steps to verify the router configuration:

1. Check the router path configuration by entering:

   SHow -PATH CONFiguration

2. Check the router port configuration by entering:

   SHow -PORT CONFiguration

3. Verify the VIP Service configuration by entering:

   SHow -VIP CONFiguration

   This command displays VINES configuration information and other related data.

4. Check the status of each port on the VINES router by entering:

   SHow -VIP STATUS

   The SHow -VIP STATUS command shows the status of each port, either Up or Down.

5. Examine the routing table to see if the destination networks are reachable by entering:

   SHow -VIP AllRoutes

   This command displays all known routes in the VINES Routing Table.

6. Display all known neighbors in the neighbor table by entering:

   SHow -VIP Neighbor

Getting Statistics

To check statistics for the VINES router, enter:

SHow -SYS STATistics -VIP

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information, refer to Chapter 58 in *Reference for NETBuilder Family Software*. For information on interpreting the statistics displays, refer to Appendix H.

Checking Reachability

You can use the VPing command to check if a specific server or router is reachable or alive. If the target server is not reachable, try reaching the intermediate routers and locate the source of the problem.

To ping a VINES server, use:

```
VPing <server address>
```

The following message appears if the target server is alive:

```
Pinging... 2901599 is alive
```

The target server must be running VINES 5.0 or greater or 3Com NETBuilder software version 6.2 or greater.

For more information on the VPing command, refer to Chapter 1 in *Reference for NETBuilder Family Software*.

Troubleshooting the Configuration

If you are unable to make connections to other networks after setting up the router, review the following troubleshooting procedure. If the router continues to operate improperly after you have completed the troubleshooting procedure, contact your network supplier or 3Com for assistance.

Procedure

To troubleshoot the VINES configuration, follow these steps:

1. Check that all cables on all routers in a specific path in the routing table are properly connected and that the routers are properly installed.

   For installation instructions, refer to the installation guide provided with your bridge/router.

2. Check the VIP CONTrol status by entering:

   ```
   SHow -VIP CONTrol
   ```

   The router displays the current values for the CONTrol parameter. If one of these values is set to NoRoute, enable the VINES router using:

   ```
   SETDefault !<port> -VIP CONTrol = Route
   ```

3. Check the VINES network status by entering:

   ```
   SHow -VIP STATUS
   ```

   Look at the status of the networks. All networks should be in the Up state. If any one is in the Down state, check to make sure that all PORT and PATH parameters are configured appropriately.

4. Check whether a specific neighbor is up and running by entering:

   ```
   SHow -VIP Neighbor
   ```

   If a neighbor is up and running on the network, it will appear in the neighbor table.

5. Check whether the network you are trying to reach is in the VINES Routing Table by entering:

   ```
   SHow -VIP AllRoutes
   ```

   The VINES router displays the routing table entries. From the table, you can determine which path is being used. Examine the entries to make sure that a route in the table is taking the appropriate path.

   If the entry in the table has a hop number of 65535 (hex FFFF), the network is unreachable at the present time. Wait several minutes and enter the SHow -VIP AllRoutes command again.

6. Display statistics for the VIP Service by entering:

   ```
   SHow -SYS STATistics -VIP
   ```

   For information on interpreting the statistics displays, refer to Appendix H.

Customizing the VINES Router

You can customize the VINES router configuration by assigning a name to the local VINES router and assigning symbolic names to neighbors in your VINES network for tracking purposes.

To assign a name to the VINES router network number, use the SETDefault -VIP RtrName command. You can rename the router to any string up to 16 characters, but the name must be unique in the VINES network.

For example, to assign the name "3Com.Engr" to the router, enter:

```
SETDefault -VIP RtrName = "3Com.Engr"
```

This router name is used when the router responds to VINES Security Service requests, which enforce network security and authentication. The service uses the router name for user ID authentication to determine whether the client from where the user is logging in is a physical neighbor and should be permitted access to the network. The router name is also used when the router responds to service statistics requests from clients invoking the WHATZ command.

To assign symbolic names to other VINES servers on your network, enter the ADD -VIP SymbolicNames command. Adding symbolic names to VINES servers can help you keep track of other VINES servers when you display the VINES neighbor and routing tables. You assign the symbolic name to the VINES network number (which must be entered in hexadecimal form).

For example, to assign the name "Finance.2ndFloor" to a VINES server with the network number 002c465f (hexadecimal), enter:

```
ADD -VIP SymbolicNames 002c465f "Finance.2ndFloor"
```

You can assign up to 128 symbolic names, and each symbolic name can be up to 15 characters long. Symbolic names are only displayed when you specifically request the symbolic name option with the SHow -VIP AllRoutes and SHow -VIP Neighbor commands. For more information, refer to "RtrName" on page 63-5 and "SymbolicNames" on page 63-6 in *Reference for NETBuilder Family Software*

Names configured with the RtrName and SymbolicNames parameters have no relationship to Banyan VINES StreetTalk names, and will not be advertised. The NETBuilder bridge/router does not support StreetTalk name server requests.

How the VINES Router Works

VINES networks are configured automatically on each port, and the configuration is transparent to the user. The port can be a local Ethernet port or a serial line port for a wide area network, such as a point-to-point link or an X.25 link.

Figure 17-1 is an example showing a wide area router connecting two local Ethernet networks (Santa Clara) to two wide area networks (Los Angeles and Santa Barbara).

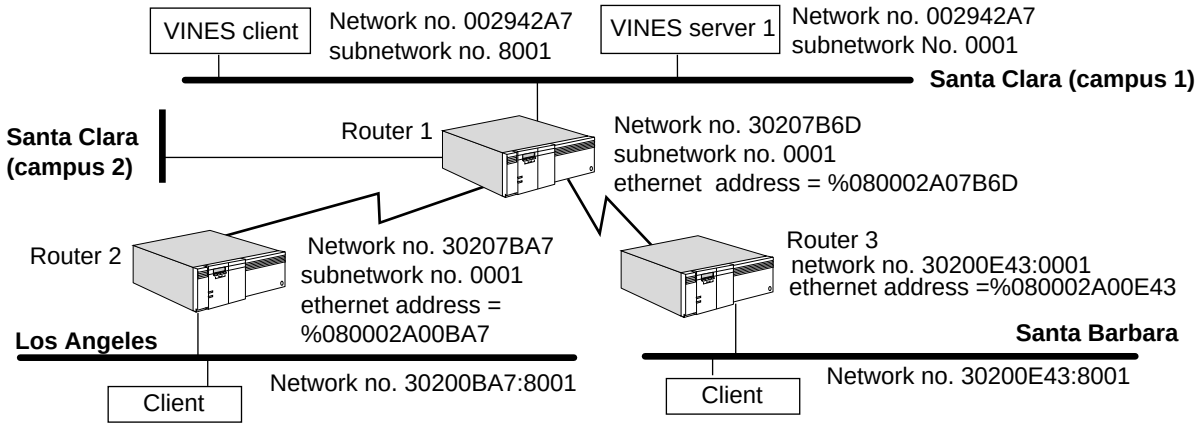


Figure 17-1 Wide Area Router Connecting Four VINES Networks

All 3Com router network numbers start with hex 302 or hex 303. As shown in Figure 17-1, all routers and servers have unique serial numbers, which are the same as the network numbers. Their subnetwork numbers are always 0001. These servers and routers assign unique network numbers and subnetwork numbers to the client nodes. Client subnetwork numbers can be any number from hex 8000 through hex FFFE. One physical network can have as many logical network numbers as the servers and routers (See Figure 17-1).

A router must check its routing table to determine where to route a packet. If the destination is one of the neighbors, the router can send it directly to the neighbor. If the destination is not a neighbor, the router must route the packet to another router (called a "gateway") that is closer to the destination. The route to a remote network can be dynamically learned through routing protocols, such as the Routing Table Protocol (RTP) for VINES.

Routing Tables

Two tables are used in VINES routing: the VINES Routing Table and the VINES Neighbor Table.

VINES Routing Table

This table displays all known routes in the routing table. To display the VINES Routing Table, enter the SHow -VIP ALLRoutes command.

The following display is an example of the default routing table:

```
---------------------------VINES Routing Table---------------------
Port  NET         Gateway     Metric  Port  NET      Gateway  Metric
5     807600533   807600533   45      1     2903035  2903035  2
Total route(s) displayed:2
```

You can also display the routing table in both hex or symbolic formats. To display the routing table in hex format, enter:

```
SHow -VIP AllRoutes Hex
```

To display the routing table in symbolic format, enter:

```
SHow -VIP AllRoutes Sym
```

The VINES Routing Table provides the following information:

- Port
- The port number of the router through which the destination is available.
- NET
- This is a logical network number learned dynamically through its neighboring routers or servers.
- Gateway
- The VIP address of the gateway to which a router must send a packet before the packet can be routed to the destination.
- Metric
- The metric for a particular interface. The metric is automatically calculated, and is based on baud rate.

- Status

  Indicates the status of the route as follows:

  Up Route is up and usable.

  Dn Route is down and soon to be purged.

  Ch Entry has been recently updated and must be included in the next RTP updates across permanent links.

  Hd1 Route is in the first hold-down period and identifies a network whose unreachable state was recently updated, but not verified.

  Hd2 Route is in the second hold-down period and indicates the unreachable state has been confirmed and it can now be advertised.

  The ROUTE status is only displayed when you display the routing table in symbolic or hex format.

VINES Neighbor Table

This table displays all known neighbors in the neighbor table. To display the VINES Neighbor Table, enter:

```
SHow -VIP Neighbor
```

The following display is an example of the neighbor table:

```
-------------------------VINES Neighbor Table---------------------
Port    NETnumber   Media Address    Metric    HdrFmt      Status
1       2903035     %02608CA1B088    2         ETH         Svr/
5       807600533   PPP              45        PPP         Svr/Perm
```

The VINES Neighbor Table provides the following information:

- Port

  Identifies the port number of the router through which the destination is available.

- NETnumber

  If the neighbor is a service node or a router, it has a unique 32-bit network number. The network number is the serial number of the service node or the router. Each service node or router has 0001 for its subnetwork number. If a neighbor is a client node, it gets its network number and subnetwork number from a service node or a router. Subnetwork numbers range from hex 8000 through hex FFFE.

- Media Address

  While network numbers and subnetwork numbers are the logical network numbers of a node, media address represents the underlying data link layer address, such as Ethernet address, X.25 address, or Frame Relay DLCI.

- Metric

  Indicates the metric (in 200 millisecond increments). For information on recommended metric values, refer to "Metric" on page 63-4 in *Reference for NETBuilder Family Software.*

- Header Format

  Indicates whether Ethernet, IEEE, or SNAP packet encapsulation is being used.

- Status

  Indicates the status of the neighbor as follows:

Svr	Neighbor is a server or a router.
CInt	Neighbor is a client.
Perm	Neighbor is a permanent, will not age out. Any neighbor learned over a serial line is considered permanent.
IP	Neighbor is learned through IP.
Redir	Neighbor is in the process of RTP redirect.

For each destination address, the router supports only one route.

Routing Selection

The VINES router keeps in its routing table only one network number per destination. It does not support backup routes.

Deleting Routes

Because VINES does not allow for static route configuration, there is no DELete command that deletes individual routes one at a time. You can delete all the entries by flushing them.

VINES Routing Table entries and neighbor table entries age out if no updates are received for about 9 minutes, which is six times the value of the user-configured UpdateTime parameter. The default value for the UpdateTime parameter is 90 seconds.

To remove all dynamic routes from the VINES Routing Table, enter:

```
FLush -VIP AllRoutes
```

This command simultaneously removes all entries from the VINES Neighbor Table so that the two tables remain consistent.

Learning Routes

Every time the router learns a route change for a network, or every 90 seconds (by default), it uses broadcast packets to report the following types of information to its neighboring gateways:

- The networks it can reach (4 bytes)
- The metric or cost associated with each network it can reach (2 bytes)

You can configure the UpdateTime parameter in the VINES Service to change the interval at which the router broadcasts routing update packets (RTPs). For more information, refer to "UpdateTime" on page 63-6 in *Reference for NETBuilder Family Software*.

Network Reachability, Split Horizon, and UpdateTime

The types of networks that are considered "reachable" when a router broadcasts its RTP update packets are as follows:

- A directly connected network
- All dynamic routes learned through RTP in the routing table

To prevent endless routing loops caused by including routes in the updates sent back to the same gateway from which the routes were originally learned, a preventive measure known as *split horizon* is used. To achieve split horizon, the router does not include those routes learned from that interface when it

generates RTP update packets to an interface. For example, when a network is learned from a neighbor on port 1, this network will not be included in any updates to port 1 to prevent mutual deception.

VINES servers currently do not support split horizon.

The UpdateTime parameter changes the frequency at which the router sends update packets. The UpdateTime parameter specifies the time interval by which the router sends its routing table updates. For networks that seldom experience topology changes, the interval time can be set higher than the default value to reduce the amount of network traffic. For networks that often experience topology changes, this value can be set lower than the default value.

The lower you set the UpdateTime value, the more data traffic will be generated on the network; increased traffic can degrade network performance.

Banyan VINES Client/Server Support

The 3Com VINES router supports a subset of the VINES Protocol suite, such as VINES Internet Protocol (VINES IP), the RTP, Address Resolution Protocol (ARP), and the Internet Control Protocol (ICP). When the VINES router receives broadcast packets, it pays special attention to ICP packets by selectively propagating VINES StreetTalk packets (for the VINES Directory Service), Time Synch Service packets, and VINES Security Service. However, the VINES router does not participate in any other VINES Directory Service.

3Com VINES routers are preassigned with a unique 32-bit network number and a subnetwork number of 0x0001. However, a client must obtain its VINES Internet address from its router or server using the VINES ARP. After a client boots up, it broadcasts an ARP Query Request seeking a response from a server or a router. Any neighbor server or router with the ARP Service enabled responds with an ARP Query Response. Two different versions of VINES ARP are available: sequenced ARP and non-sequenced ARP. All VINES servers and clients running Banyan VINES software previous to version 5.50 use non-sequenced ARP, while servers and clients running VINES software version 5.50 and later use sequenced ARP. For the two types to interoperate, nodes that support sequenced ARP also support non-sequenced ARP. For example, a client node that runs VINES 5.50 can use a VINES 5.0 server if no VINES 5.50 servers are available, and a server that runs VINES 5.50 can provide an ARP Service to a VINES 5.0 client node.

This version of the 3Com VINES router does not support sequenced ARP. The 3Com VINES router uses the RTP to exchange routing information with servers or routers, and to maintain the topology information in the routing table. When routing data packets, the 3Com VINES router makes routing decisions based on the routing database. If the final destination of a packet is a neighbor, the router will send the packet to the neighbor directly. Otherwise, it will send the packet to the next router toward the final destination. Each RTP update packet contains a list of all the networks known to the router and metric for each network.

Two versions of RTP are available: sequenced RTP and non-sequenced RTP. All VINES servers and clients running Banyan VINES software previous to version 5.50 use non-sequenced RTP, while servers and clients running VINES software version 5.50 and later use sequenced RTP. For interoperability, routers that

support sequenced RTP also support non-sequenced RTP for backward compatibility. This version of the 3Com VINES router does not support sequenced RTP.

The 3Com VINES router provides support for RTP Redirect. When a unicast packet has to be forwarded on the same port on which it was received and the RTP Redirect bit is set, 3Com routers generate an RTP Redirect packet to inform the last forwarding router or server of a better path to the given destination. The advantage of RTP Redirect is that an unnecessary extra hop can be reduced.

18 CONFIGURING XNS ROUTING

This chapter describes the procedures for configuring your system to perform Xerox Network Systems (XNS) routing. It also describes how the router works and gives guidelines for operating, managing, and troubleshooting it.

For conceptual information, refer to "How the XNS Router Works" on page 18-8.

Setting Up a Basic XNS Router

The procedure in this section describes the minimum steps required to enable your system to route XNS packets. Depending on your network requirements, you can use the default values of the parameters, or you can further configure the router according to later sections in this chapter.

The parameters in the IDP and RIPXNS Services enable XNS routing functions.

Configuring for Local Area Networks and Point-to-Point Protocol Links

When setting up the basic XNS router, you first configure the router for LAN ports and Point-to Point Protocol (PPP) links.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Set up the ports and paths of your bridge/router according to Chapter 1.

Procedure

To set up the router for XNS routing, follow these steps:

1 Enable XNS routing by entering:

```
SETDefault -IDP CONTrol = Route
```

In addition, you can configure the -IDP CONTrol parameter to provide error checking with the Checksum | NoChecksum value. Checksum provides a high degree of reliability in detecting bad data sent over the network. If Checksum is enabled, a router verifies the IDP checksum of a packet before it forwards the packet. The cost of this service, however, is lower network performance. The default value is NoChecksum.

2 Configure XNS network numbers on each port connected (local interface or serial line interface) using:

```
SETDefault !<port> -IDP NETnumber = &<number>(0–FFFFFFFE)
```

Valid network numbers consist of up to eight hexadecimal digits in the range &0 to &FFFFFFFE. The network number &FFFFFFFF is reserved. Use network number &0 to delete a previously assigned network number. You do not have to specify leading zeros in the network number.

Repeat this step for the other port(s). Each enabled port on a router must be assigned a different network number.

3 Verify the XNS configuration by entering:

```
SHow -IDP CONFiguration
```

The router displays the IDP configuration information. If the CONTrol parameter is not set to route, or the NETnumbers are incorrect, repeat steps 1 and 2.

4 Begin routing table information exchanges with other routers that interface with a port using:

```
SETDefault !<port> -RIPXNS CONTrol = Enabled
```

5 Repeat step 4 for each port being used for XNS routing.

After you have completed this procedure, dynamic XNS routing begins over the configured ports. To complete the configuration for PPP links, refer to Chapter 34.

For more information on dynamic and static routes, refer to "Customizing the XNS Router" on page 18-4.

Configuring for Wide Area Networks

XNS routing over Frame Relay, Asynchronous Transfer Mode data exchange interface (ATM DXI), and X.25 is supported over fully meshed, partially meshed, and nonmeshed topologies. If you plan to route XNS over a partially meshed or nonmeshed topology, you must create a virtual port for each remote network that is attached to a Frame Relay, ATM DXI, or X.25 cloud. For complete information on configuring XNS routing over Frame Relay, ATM DXI, or X.25, including a discussion of fully meshed, partially meshed, and nonmeshed topologies and virtual ports, refer to Chapter 42, Chapter 43, and Chapter 45, respectively. For information on the number of virtual ports supported per platform, refer to "Virtual Ports" on page 1-3.

Routing XNS over Switched Multimegabit Data Service (SMDS) is supported over fully meshed and nonmeshed topologies (nonmeshed topologies require virtual ports). In addition, SMDS virtual ports are supported and can be used for traffic separation and various filtering of by assigning groups of nodes to different virtual ports. For more information, refer to Chapter 44.

To configure your XNS router to perform routing over PPP or phone line gateway (PLG), refer to Chapter 34. For more information on wide area networking using Integrated Services Digital Network (ISDN), refer to Chapter 35.

Verifying the Configuration

After you have configured the basic XNS router, you should verify the configuration to see if you can reach other XNS hosts.

Before you use the router for interconnecting networks, verify the router configuration by following these steps:

1 Check the router path configuration by entering:

```
SHow -PATH CONFiguration
```

2 Check the router port configuration by entering:

```
SHow -PORT CONFiguration
```

3 Examine the IDP Service configuration by entering:

```
SHow -IDP CONFiguration
```

This command displays configuration information specific to the IDP Service parameters for each port that you have configured with a network number.

4 Examine the RIPXNS Service configuration by entering:

```
SHow -RIPXNS CONFiguration
```

This command displays configuration information specific to the RIPXNS Service parameters for each port that you have configured with a network number.

5 Check the state of all networks assigned to the ports of a router by entering:

```
SHow -IDP NETnumber
```

This command displays the network number assigned to each port on this router and the state that each network is in. All networks should be in the UP state. If any one is in the DOWN state, check to make sure that all PORT and PATH parameters are configured correctly.

6 Check the XNS Routing Table to see if all the networks are reachable by entering:

```
SHow -IDP AllRoutes Long
```

This command displays all known routes, both dynamic and static, in the XNS Routing Table.

7 Make a connection from a host on one attached network to a host on another network to see if packets can be routed across the router.

You can also test the connectivity between routers by using the REMote command.

Figure 18-1 shows four Ethernet networks connected by routers A, B, and C.

To check the connectivity between router A and router B, on router A enter:

```
REMote &3141%080002001234
```

To check the connectivity between router A and router C, on router A enter:

```
REMote &3142%08000200abcd
```

After you enter the REMote command, the remote prompt (Remote:) appears. At the Remote prompt, enter any command available on the device to which you remote (Routers B or C); for example, SHow -SYS VERSion or SHow -SYS ADDRess. A response from Routers B or C indicates successful communication between respective routers.

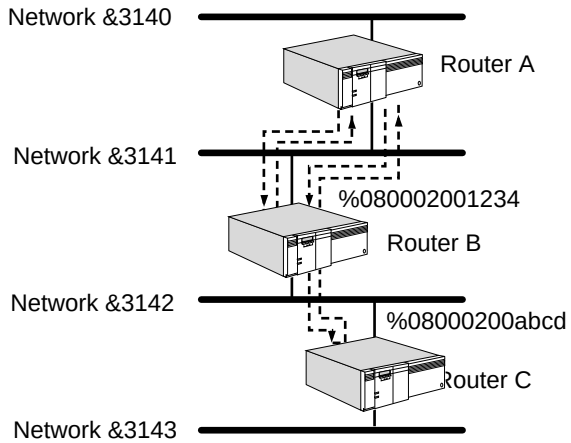


Figure 18-1 Checking Connectivity between Routers

Getting Statistics

To display statistics for the IDP Service, enter:

```
SHow -SYS STATistics -IDP
```

To display statistics for the RIPXNS Service, enter:

```
SHow -SYS STATistics -RIPXNS
```

You can collect statistics for a specific time period by using the SampleTime and STATistics parameters. For more information on these parameters, refer to Chapter 58 in *Reference for NETBuilder Family Software*. For information on interpreting the statistics displays, refer to Appendix H.

Troubleshooting the Configuration

If you are unable to make connections to other networks after setting up the router, review the following troubleshooting procedure. This procedure can help correct problems in making single-hop (involving one router) and multiple-hop (involving more than one router) connections.

To troubleshoot the basic XNS router configuration, follow these steps:

1. Check that all cables on all routers in a specific path in the routing table are properly connected and that the routers are properly installed.

   For installation instructions, refer to the installation guide provided with your bridge/router.

2. Check the -IDP NETnumber and the network status by entering:

   ```
   SHow -IDP NETnumber
   ```

   Look at the status of the networks. All configured networks should be in the UP state. If any one is in the DOWN state, check that all PORT and PATH parameters are correctly configured.

   Look at the current network configuration. If no network is configured on the specific port, use the SETDefault -IDP NETnumber command to add a proper network number to that port.

3. Check the values of -RIPXNS CONTrol parameter by entering:

   ```
   SHow -RIPXNS CONTrol
   ```

   The router displays the current values for the CONTrol parameter.

4. Check whether the network you are trying to reach is in the XNS Routing Table by entering:

   ```
   SHow -IDP AllRoutes
   ```

   To verify single route reachability, you can specify a network number and enter:

   ```
   SHow -IDP AllRoutes <NETnumber>
   ```

   For more information on checking the routing table, refer to "Displaying Routing Information" on page 18-8.

Customizing the XNS Router

After you set up and check the router according to instructions in the previous sections, you are ready to customize the XNS router by configuring specific routes, which includes the following steps:

- Determining network routes dynamically and statically
- Making routing decisions (that is, determining whether a packet destination is on an attached network or a reachable remote network and determining how to reach the destination if multiple routes are available)

This section describes these router activities and explains how you can influence the router's routing decisions under different circumstances.

Local and Wide Area Network Configuration

An XNS network is configured on each port where XNS packets are received and sent. Figure 18-2 is an example showing a wide area router connecting two local Ethernet networks (Santa Clara) to two wide area networks (Los Angeles and Santa Barbara).

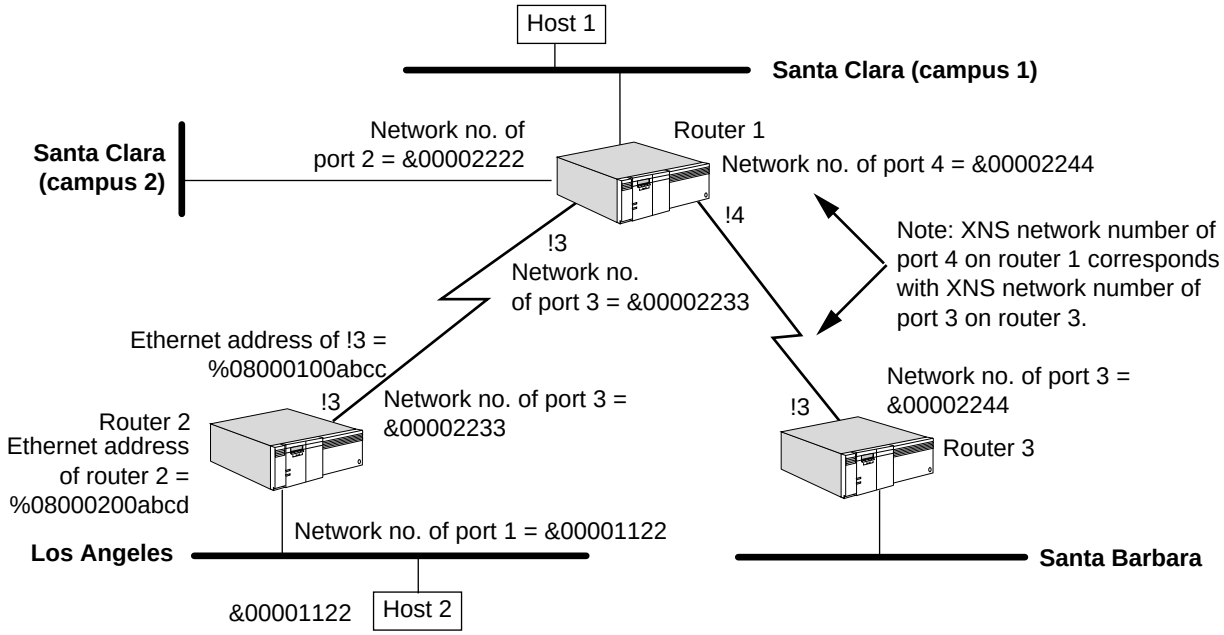


Figure 18-2 Wide Area Router Connecting Four XNS Networks

Any physically attached network, Ethernet or serial line, is considered a directly connected network or "local" network. If more than one serial line is assigned to one port, that port is considered a single directly connected XNS network.

A router must check its routing table to determine where to route a packet. If the destination is on an attached network, the router can send it directly to the network. But if the destination is not directly connected, the router must route the packet to another router (called a *gateway*) that is closer to the destination. The route to a remote network can be statically configured or dynamically learned through routing protocols, such as the Routing Information Protocol (RIP) for XNS.

Defining Routes

The following sections describe the two types of routes (static and dynamic) and how to define them.

Static Routes

A static route is a user-defined route by which a remote network can be reached. To define a static route, enter the ADD -IDP ROUte command and specify the appropriate route information. For more information on setting the ROUte parameter, refer to Chapter 27 in the *NETBuilder Family Bridge/Router Reference Guide*.

For example, on router 1 in Figure 18-2, you can add a static route for the Los Angeles network as follows:

```
ADD -IDP ROUte &1122 &2233%08000100abcc
```

To display the table of static routes, enter:

```
SHow -IDP ROUte
```

Once a static route is configured for a specific destination network, no dynamic routes will be added for that destination network.

You must configure the router with a network number (refer to "Displaying Routing Information" on page 18-8) before the router will accept static routes.

Dynamic Routes

Dynamic routes are routes that are learned dynamically through RIP. RIP allows the periodic exchange of routing table information with other XNS routers. Gateways use this information to route packets to other networks. For more information on this protocol, refer to "Learning Routes" on page 18-8.

Enhancing the Performance of the XNS Router

This section describes ways that you can enhance the performance of the XNS router.

Configuring for RIP Updates

You can change the way the router broadcasts update packets using parameters in the RIPXNS Service (refer to Table 18-1).

Table 18-1 Configuring the XNS Router for RIP Updates Using RIPXNS Parameters

Parameter	Result
UpdateTime	Changes the frequency at which the router sends update packets.
CONTrol parameter options:	
Enabled | Disabled	Determines whether router sends update packets.
Trigger | NoTrigger	Determines whether a route change for a network triggers an update packet from the router.
Poison | NoPoison	Determines how router handles entries learned from another router.
OldNbrMap | NewNbrMap	Permits neighbor address mapping for any bridge/router software versions. If your software version is earlier than 5.0, use option OldNbrMap. If your version is 5.0 or later, use option NewNbrMap (this is the default).
GlobBcast | NoGlobBcast	Determines whether XNS global broadcast packets are forwarded to all interfaces except the incoming port.

The RIPXNS parameters are automatically configured to their default values when you configure the -IDP CONTrol parameter for routing. In some cases, however, you may want to change the default configuration.

To modify the RIPXNS parameters, refer to the following parameter descriptions:

- CONTrol

  The -RIPXNS CONTrol parameter determines on a per-port basis how the router sends the routing table information to the network. The following are the default values for the RIPXNS parameters:

  CONTrol = (Enabled, Trigger, NoPoison, NewNbrMap, GlobBcast)

  The impact of setting the -RIPXNS CONTrol parameter to Enabled depends on the setting of the -IDP CONTrol parameter. Table 18-2 shows the relationship of the -IDP CONTrol parameter to the -RIPXNS CONTrol parameter.

Table 18-2 CONTrol Parameters in IDP and RIPXNS

CONTrol Setting in IDP	CONTrol Setting in RIPXNS	Effect
Route	Enabled	Packet routing starts. Enables routing table updates based on packets received from other gateways. Routing table update packets are generated and sent to other networks. Allows normal routing performance.
NoRoute	Enabled	Packet routing stops. Allows routing table updates based on the packets received. Routing table update packets are not generated and sent to other networks. Allows normal routing performance when packet routing resumes.
Route	Disabled	Packet routing starts. Packets are routed according to static routes only. Routing table updates received are ignored. Routing table updates are not generated and sent to other networks. Reduces the amount of network data traffic and allows network administrator control over packet routing.
NoRoute	Disabled	Packet routing stops. Routing table updates stop (no packets are received or generated).

  Setting the -RIPXNS CONTrol parameter to Trigger causes the router to send an update packet when the network topology changes. The advantage is that triggered updates immediately allow the network to know a potentially better route to a particular network. Setting the -RIPXNS CONTrol parameter to NoTrigger reduces the amount of data packets broadcast over the network, and normal update packets are sent only at the time interval specified by the UpdateTime parameter.

  Setting the -RIPXNS CONTrol parameter to Poison causes the router to set the number of hops for a specific table entry to 16 when it sends routing table updates. It does this to prevent routing loops in which two gateways are trying continually to update each other with the same information. The poisoned information (specified by a hop count of 16) remains in the router's update packet, adding to the data traffic on the network.

  Setting the -RIPXNS CONTrol parameter to NoPoison prevents the router from sending poisoned routing information in an update packet, thus reducing the amount of data traffic over the network.

- UpdateTime

  The -RIPXNS UpdateTime parameter specifies the time interval by which the router sends its routing table updates. For networks that seldom experience topology changes, the interval time can be set higher than the default value

to reduce the amount of network traffic. For networks that often experience topology changes, this value can be set lower than the default value.

The lower you set the UpdateTime value, the more data traffic is generated on the network. Increased traffic can degrade network performance.

Configuring for Error Checking

In addition to routing configuration changes available through the RIPXNS Service parameters, you can configure the -IDP CONTrol parameter to provide error checking through the Checksum | NoChecksum value. Checksum provides a high degree of reliability in detecting bad data sent over the network. If Checksum is enabled, a router verifies the IDP checksum of a packet before it forwards the packet. The cost of this service, however, is lower network performance. The default value is NoChecksum.

To configure the router to provide error checking, enter:

```
SETDefault -IDP CONTrol = Checksum
```

How the XNS Router Works

This section provides general information about XNS routing.

Learning Routes

Normally, every 30 seconds (by default) or every time it learns a route change for a network, the router uses broadcast packets to report to its neighbors the following types of information:

- The networks it can reach
- The number of hops associated with each network it can reach

You can configure some router parameters to determine how the router sends out the updates by completing the following tasks:

- Changing the frequency of broadcast traffic.
- Configuring the router so that it does not send or receive update and request packets.
- Configuring the router not to send out a trigger update response when it learns a route change for a network.

Displaying Routing Information

The routing table provides information that determines how a packet is routed. The long form of the routing table displays only the most efficient route.

To display the long form, enter:

```
SHow -IDP AllRoutes Long
```

The following is a typical example of the long form of the routing table:

```
----------------------XNS Routing Table----------------------------
Port     NETnumber            Gateway                         Hops
1        &00003145*           &00003140%080002015980          2
1        &00003147            &00003140%080002015982          5
1        &00003149*           &00003140%080002015980          7
Total route(s) displayed:     3
```

Asterisks in the display indicate static routes.

Depending on the AllRoutes option selected, the routing table can include the following information, which determines how a packet is routed:

- Port number

  This is the port associated with the attached network.

- Network number

  The router maintains valid routes to remote networks. A network route is used to reach all hosts on the network. If you have a large routing table, you can specify a network number to verify its reachability by using the SHow -IDP AllRoutes <NETnumber> syntax.

- Gateway address

  This is the XNS address of the gateway to which a router must send the packet before the packet can be routed to the destination. For more information on gateway addresses, refer to "Static Routes" on page 18-5.

- Number of hops between router and destination

  The numbers of hops is equal to the number of gateways traversed. The XNS router selects the most efficient path for information. The most efficient path is the path that requires the fewest hops to reach a destination. In cases where two paths require the same number of hops, the router selects the first entry in the routing table.

For each destination address, the router can support up to two routes (that is, two gateways). These routes, either learned or configured, are stored in the routing table. The router selects the most efficient route to reach a destination. For information on how the router makes routing decisions, refer to "Learning Routes" on page 18-8.

To display the short form of the routing table, enter:

```
SHow -IDP AllRoutes
```

The short form, which is the default, only displays network numbers and hop counts.

Deleting Routes

Routes in the routing table are deleted differently depending on whether they are static or dynamic routes:

- A static route can be removed using the DELete -IDP ROUte command.

  For example, to delete the Ethernet static route configured in "Static Routes" on page 18-5, enter:

  ```
  DELete -IDP ROUte &1122
  ```

- A dynamic route learned through RIP is deleted when the router's internal timer (approximately three times the value of the -RIPXNS UpdateTime parameter) expires.

  For example, if the UpdateTime parameter is set to 30 seconds, the route is deleted if no RIP updates are received for the route within 90 seconds.

To remove all dynamic routes, enter:

```
FLush -IDP AllRoutes
```

Network Reachability and Split Horizon

The types of networks that are considered *reachable* when a router broadcasts its RIP update packets are as follows:

- All directly connected networks
- All static routes
- Dynamic routes learned through RIP and currently in the routing table

Some networks, though accessible, are not reported by the router. For example, in Figure 18-3, router B broadcasts an update packet on network &2222. The packet does not include network &1111, because this network is learned from the same port on which the packet is broadcast. This process is known as *split horizon*.

Split horizon prevents routing loops caused by including routes in the updates sent to the port from which the routes were originally learned.

When no poison reverse is used, the router omits this type of route from routing updates sent to the same port.

With poison reverse, the router includes this type of route in its report, but the number of hops associated with that network is 16. For example, with poison reverse, router A includes networks &1111 and &3333 in its report sent to router B, but specifies that the number of hops for network &3333 is 16, while the number of hops for network &1111 is 1. Because RIP considers any network with a hop number higher than 15 unreachable, router B, upon receipt of the report, knows that packets destined for network &3333 should never be routed to router A. Through this same process, router A will know network &1111 is unreachable through router B.

Split horizon guarantees that if router B's connection with network &3333 fails, it will not send packets to router A, under the assumption that router A can reach the destination network (&3333), because it cannot.

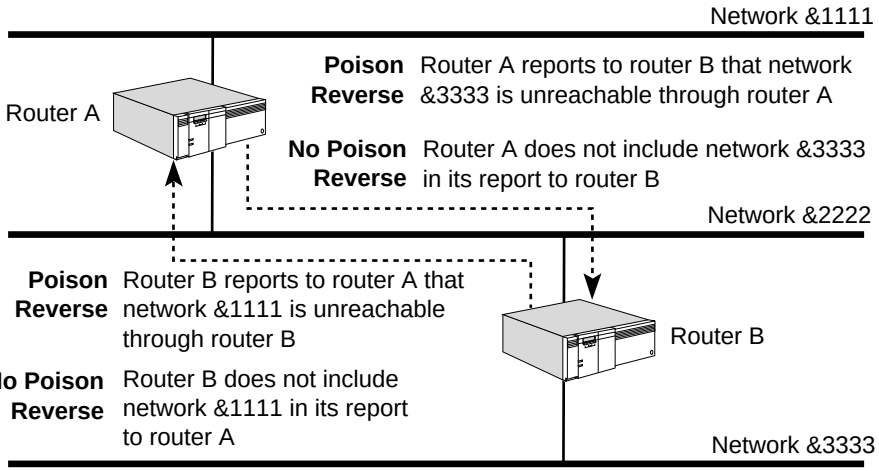


Figure 18-3 XNS Routing Using Split Horizon

19 CONFIGURING THE ROUTER DISCOVERY PROTOCOL

This chapter describes how to configure your system to use Internet Control Message Protocol (ICMP) Router Discovery Protocol (RDP) messages to dynamically generate a list of active neighbor router addresses. RDP is an extension of ICMP, and assists hosts in discovering neighboring routers. It is also a requirement per RFC 1812 for IP V4 routers on all connected networks that support either IP multicasting or IP broadcast addressing.

This chapter also describes how RDP works and gives guidelines for operating, managing, and troubleshooting it.

For conceptual information, refer to "How RDP Works" on page 19-4.

Setting Up RDP

The procedure in this section describes the steps required to enable your system for RDP. Depending on your network requirements, you can use the default values of the parameters in the various services, or you can configure the router with custom values.

Prerequisites

Before beginning this procedure, complete the following tasks:

- Log on to the system with Network Manager privilege.
- Configure your ports and paths as described in Chapter 1.
- Configure IP routing as described in Chapter 19 or IP multicast routing as described in Chapter 9 of this guide.

Procedures

To set up RDP, see Figure 19-1 and perform these procedures.

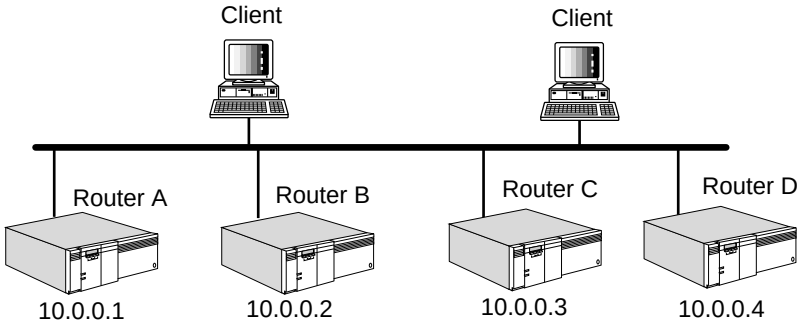


Figure 19-1 Routers Participating in RDP

Defining Participating Routers

To create the list of routers that will participate in RDP, follow these steps:

1 Define the list of routers that will participate in the router discovery process using:

```
ADD RouterList <IP address>[NoAdvertise][<preference
  level>|Infinity]
```

Enter the IP address for each system in router mode to be advertised. The length of this list is limited only by available system memory. For example, to list the routers in Figure 19-1 for participation in RDP, enter:

```
ADD -RDP RouterList 10.0.0.1
ADD -RDP RouterList 10.0.0.2
ADD -RDP RouterList 10.0.0.3
ADD -RDP RouterList 10.0.0.4
```

a You can also enter the IP address for each system in router mode, and assign a preference level to specify the default routers that will learn from router advertisement messages using:

```
ADD RouterList <IP address> <preference level>
```

The <preference level> option indicates the preferences for selecting the default router. For example, to designate a router as the default, enter:

```
ADD -RDP RouterList 10.0.0.1 100
```

The preference level option is a 32-bit, signed, twos-complement integer, which allows you to enter a definitive number to specify the selection criteria for the default router. The higher the value assigned, the more preference the router address has.

b To indicate that an address *not* be used as a default router address, enter:

```
ADD -RDP RouterList 10.0.0.4 Infinity
```

The Infinity option indicates a minimum value (0x80000000) that prevents it from being picked up by hosts as a default router.

c If you do not want a router to participate in the discovery process, enter its IP address with the NoAdvertise option. For example:

```
ADD -RDP RouterList 10.0.0.6 NoAdvertise
```

Configuring the Timers

To configure the RDP timers, follow these steps:

1 Specify a value for the lifetime field in router advertisement messages using:

```
SETDefault !<port> -RDP LifeTime = <seconds>(4-9000) | Default
```

The default is 30 minutes (1800 seconds), and is valid only on routers in router mode.

2 Specify a value for the maximum interval between two router advertisement messages using:

```
SETDefault !<port> -RDP MAxInterval = <seconds>(4-1800) | Default
```

The default is 10 minutes (600 seconds), which must be less than the value of the LifeTime parameter, and is valid only on routers in router mode.

3 Specify a value for the minimum interval allowed between two router advertisement messages using:

```
SETDefault !<port> -RDP MInInterval = <seconds>(3-1800) | Default
```

The default is 75 percent of the MAxInterval value (nine minutes or 450 seconds), which must be less than the value set for MAxInterval, and is valid only on routers in router mode.

For example, the default lifetime of the router advertisement message is 30 minutes, and the default interval between the messages is ten minutes. To change these values to a lifetime value of 12 minutes and an interval of 6 minutes, enter:

```
SETD !1 -RDP LifeTime = 720
SETD !1 -RDP MAxInterval = 360
SETD !1 -RDP MInInterval = 300
```

The value of MInInterval must always be less than that set for MAxInterval.

Enabling and Disabling RDP

To enable or disable RDP, follow these steps:

1 Enable or disable RDP globally using:

```
SETDefault !<port> -RDP CONTrol = ([Auto | Enable | Disable],
  [Multicast | Broadcast])
```

The default is Auto, which enables RDP on local area networks, but not wide area networks.

2 To enable RDP on a wide area network, enter:

```
SETDefault !1 -RDP Control = Enable
```

3 To specify that packets are multicasted, enter:

```
SETDefault !1 -RDP Control = Multicast
```

When the system is set in host mode, the Multicast option sends router solicitation messages out with the IP destination set to the all-routers address (244.0.0.2).

When the router is set in router mode, the Multicast option sends router advertisement messages out with the IP destination address set to the all-host IP address (244.0.0.1). This is the default mode.

4 To specify IP broadcasting when it is enabled, enter:

```
SETDefault !1 -RDP Control = Broadcast
```

Both router solicitation and router advertisement messages are sent out with the IP destination set to the limited-broadcast IP address (255.255.255.255).

Discovering Neighboring RDP Routers

To discover neighboring RDP routers by having the system send out router solicitations, use:

```
DiscRouteRs [!<port> | <source IP>] [Broadcast] [<timeout (1-30
  seconds)>]
```

For systems in host mode (!0), to discover neighboring routers, specify either an outgoing port number or one of the system source IP addresses to be sent with the router solicitations. After the command is entered, the system transmits

router solicitations every second until reaching the time set with the timeout option. During the timeout period, any router advertisements that are received are displayed.

Verifying the RDP Configuration

To verify that RDP is recognized by the IP network, display the values associated with RDP using:

```
SHow [!<port>] -RDP CONFiguration
```

Troubleshooting the RDP Configuration

You can troubleshoot the RDP operation using one or more of these steps:

1. Display the set of interfaces enabled or disabled for RDP using:

   ```
   SHow [!<port>] -RDP CONTrol
   ```

2. Display the router list by entering:

   SHow -RDP RouterList

3. Delete one or all of the interfaces enabled or disabled for RDP using:

   ```
   DELete -RDP RouterList {<IP address>|ALL}
   ```

4. Flush the router list and allow the routes to be relearned by entering:

   FLush -RDP RouterList

   This command only works when the bridge/router is in host mode.

5. Display the value of the router advertisement lifetime field using:

   ```
   SHow [!<port>] -RDP LifeTime
   ```

6. Display the value of the maximum interval between router advertisement messages using:

   ```
   SHow [!<port>] -RDP MAxInterval
   ```

7. Display the value of the minimum interval between router advertisement messages using:

   ```
   SHow [!<port>] -RDP MInInterval
   ```

How RDP Works

RDP is a process defined by RFC 1256 that allows a router to use two messages, router advertisements and router solicitations, to discover the addresses of neighboring routers. The RDP process works only on routers enabled for the process that are in the same subnetwork. The router listens for router advertisements, or can solicit an address by sending a router solicitation message.

The discovery process is dynamic because the list collected contains only the addresses of active routers. Routers that are not actively sending router advertisements are dropped from the list, and are reinserted in the list only when they come back up and begin sending messages again.

RDP is not a routing protocol; it only allows hosts to keep track of neighboring routers, not which router is best to reach a particular destination. This protocol uses two ICMP messages to provide a simple router discovery method that provides a list of router addresses on a multicast link without manual configuration and that is independent of the routing protocol being used.

Although RDP cannot make routing decisions, if a host makes a poor choice for a first-hop router for a destination, it receives an ICMP Redirect message identifying a better route.

RDP Features

The router advertisement message includes a preference level for each advertised router address. When a host system must choose a default router, it is expected to select router addresses with a high preference level. You set this preference level when you specify the list of routers that will participate in the RDP process.

To make sure that hosts ignore routers that go down, the router advertisement message also includes a lifetime field that specifies the maximum length of time that advertised addresses are to be considered valid by hosts. The default advertising rate is every ten minutes, and the default lifetime span is 30 minutes. The defaults minimize the load imposed on the links by the periodic transmission of the messages, but you can change the defaults as needed. For example, you may want to decrease the lifetime value so that you can become aware of routers that go down before the 30-minute period is up.

Other Timer Considerations

When an interface enabled for advertising the router address becomes active, the router begins transmission of periodic router advertisement messages. The interval between the first three messages cannot be greater than 16 seconds. After these first three messages, however, the interval is randomly chosen from the values configured for each interface. A new random interval is chosen for each transmission to reduce the possibility that all routers will transmit packets at the same time.

Periodic advertisements are either multicasted to the all-host address (244.0.0.1) or broadcasted to the limited-broadcast address (255.255.255.255), depending on the system configuration. The router also transmits advertisements in response to host solicitations. When the source IP address is not set to 0, the reply is unicasted to the host; otherwise, it is multicasted to the configured address.

The host transmits no more than three RDP router solicitation messages when an interface becomes active. The first message is transmitted within one second and then retransmitted at 3-second intervals. Transmissions stop when a host receives a valid router advertisement message. Router solicitations can be configured to be either multicast to the all-routers address (244.0.0.2) or broadcast to the limited-broadcast address (255.255.255.255).

RDP Terms

The following terms are used in this chapter to explain RDP:

multicast link	A link over which the IP multicast or IP broadcast services are supported, and can include media such as satellite, point-to-point, and store-and-forward networks such as SMDS.
neighbor	Router with an IP address belonging to the same subnet.
default router	The router address that has the highest preference level. Unless a host has been redirected or is configured to use a specific router address, it must choose a default router address for a particular destination. You can set the preference level to encourage or discourage use of a particular router.

20 CONFIGURING UDP BROADCAST HELPER

This chapter describes the User Datagram Protocol (UDP) Broadcast Helper feature. This feature allows applications in the Transmission Control Protocol/Internet Protocol (TCP/IP) stack to forward broadcast packets through a gateway (router) and to another network segment. The broadcast packets are typically requests from clients for access to servers, which may contain address, configuration, or name information.

A common application for UDP Broadcast Helper is related to the Bootstrap Protocol (BOOTP) and the Dynamic Host Configuration Protocol (DHCP). UDP Broadcast Helper for BOOTP and DHCP assists clients with finding their boot servers when the boot servers are located through a router and on another network segment.

3Com implements the BOOTP and DHCP relay agents in the bridge/router software, allowing existing BOOTP clients to interoperate with DHCP servers. The clients and associated servers do not need to reside on the same IP network or subnet, and changes to the client's initialization software is unnecessary.

This chapter provides information on how to configure UDP Broadcast Helper through the UDPHELP Service and how to verify the configuration. It also provides information on how to configure and customize the configuration.

For conceptual information, refer to "How UDP Broadcast Helper Works" on page 20-9.

Configuring UDP Broadcast Helper

UDP Broadcast Helper allows you to configure up to 32 UDP ports on your bridge/router using the ADD -UDPHELP ActivePorts command.

UDP Broadcast Helper supports several names of well-known services. The names of these services are mapped to specific UDP port numbers. (The name-to-UDP port mappings are also referred to as *built-in names*.) You can configure UDP ports using built-in names. Table 20-1 lists the supported service names, the UDP port numbers they are mapped to, and the mnemonic name for each name-to-UDP port mapping.

Table 20-1 Supported Service Name-to-UDP Port Mappings

UDP Port Description	UDP Port Number (Decimal)	Mnemonic Name
Daytime	13	DAYTIME
Time	37	TIME
Host name server	42	IEN116

(continued)

Table 20-1 Supported Service Name-to-UDP Port Mappings (continued)

UDP Port Description	UDP Port Number (Decimal)	Mnemonic Name
Domain name server	53	DNS
TACACS – database service	65	TACACS
Bootstrap protocol server	67*	BPSERVER
Trivial file transfer	69	TFTP
HOSTS2 name server	81	HOSTS2
NIC host name server	101	NIC
Simple file transfer protocol	115	SFTP
NetBIOS name service	137	NBNAME
NetBIOS datagram service	138	NBDATA
AppleTalk Name Binding	202	ATNBP
AppleTalk zone information	206	ATZIS

* BOOTP and DHCP use the same UDP port numbers: server port (67 decimal) and client port (68 decimal).

The UDP ports and built-in name mappings listed in Table 20-1 are reserved and cannot be changed or reconfigured.

Prerequisites Before beginning this procedure, complete the following tasks:

- Log on to the bridge/router with Network Manager privilege.
- Set up ports and paths according to Chapter 1.
- Set up the client and server LANs and WANs according to Chapter 6.
- Examine your network configuration and determine upon which bridge/routers UDP Broadcast Helper should be enabled.
- To determine what services are available through UDP Broadcast Helper, refer to Table 20-1. The number of services you want to configure determines the number of UDP ports you must configure.
- For each UDP port you intend to use, determine which networks or servers should receive related broadcast packets.
- Determine the IP addresses of the networks and servers that should receive broadcast packets.

Procedure To set up UDP Broadcast Helper, see Figure 20-1 and follow these steps:

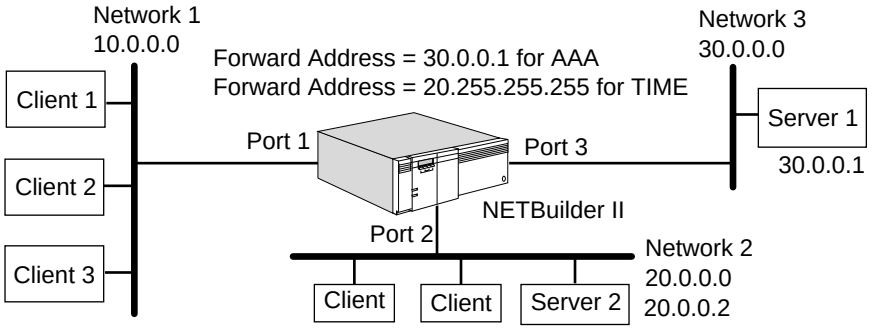


Figure 20-1 Configuring UDP Broadcast Helper

A UDP port is part of an entity address and not related to an interface (port) on the bridge/router. In the command syntax, the UDP port does not need to be preceded by an exclamation point (!).

1 Enable UDP Broadcast Helper by entering:

```
SETDefault -UDPHELP CONTrol = Enable
```

2 Determine which UDP ports your bridge/router will be listening to or helping. Add each of these UDP ports to an active ports list using:

```
ADD -UDPHELP ActivePorts {<UDP port> | <name>}
```

You can specify a UDP port by either UDP port number or name. If you specify a UDP port by name, the name can be either a built-in or a name that you define.

If you want to specify a UDP port by a defined name, you must map the name to a UDP port number first as described in step 3, then add the UDP port to the active ports list as described in this step. To specify a UDP port by a defined name, you must perform step 3 first.

For example, to add UDP port 100, enter:

```
ADD -UDPHELP ActivePorts 100
```

To add a UDP port with the built-in name TIME, enter:

```
ADD -UDPHELP ActivePorts TIME
```

TIME is the name of a service that has a UDP port number mapped to it (refer to Table 20-1). In addition to specifying this UDP port by its built-in name, you can also specify this UDP by the port number mapped to this service. For example, you can enter:

```
ADD -UDPHELP ActivePorts 37
```

To add a UDP port with a name you define, for example, AAA, enter:

```
ADD -UDPHELP ActivePorts AAA
```

3 If you added a UDP port and specified it by port number, you can optionally define a name for the port and map the name to the port number. If you added a UDP port and specified it by a built-in name, skip this step and go on to step 4. If you want to add a UDP port and specify it by a name you defined, you must map the name to a UDP port number.

Use:

```
ADD -UDPHELP Name <name string> <UDP port>
```

For example, to map the defined name AAA to UDP port number 100, enter:

```
ADD -UDPHELP Name AAA 100
```

4 For each UDP port you added to the active ports list, 3Com recommends that you set up a list of networks and servers that should receive UDP broadcast packets.

After you add a UDP port to the active ports list, the bridge/router automatically forwards broadcast packets destined for the UDP port to all interfaces. You do not need to set up a list of networks and servers that should receive UDP broadcast packets. However, 3Com strongly recommends limiting the networks and server that receive UDP broadcast packets to help prevent broadcast storms and loops.

You can use one of the following syntaxes:

```
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address>
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address>
 <subnet mask>
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address>
 <subnet mask> [Ones | Zeroes]
ADD -UDPHELP ForwardAddress <UDP port or name> <list of
 interfaces>
```

You can add up to 32 addresses to the forward address list.

For example, using the network configuration shown in Figure 20-1, add server 1 to a list for UDP port AAA by entering:

```
ADD -UDPHELP ForwardAddress AAA 30.0.0.1
```

The bridge/router forwards broadcast packets destined for UDP port AAA to server 1 only.

To add network 2 to a list for UDP port TIME, enter:

```
ADD -UDPHELP ForwardAddress TIME 20.0.0.0
```

The bridge/router forwards broadcast packets destined for UDP port TIME to all nodes on network 2.

The bridge/router does not rebroadcast packets through X.25, Frame Relay, and SMDS interfaces. You must add the IP address of each server to the list of servers that must receive UDP broadcast packets.

5 To limit the reach of a broadcast packet and the potential duration of broadcast storms, 3Com recommends you specify the default number of seconds that pass before a broadcast packet is discarded. Use:

```
SETDefault -UDPHELP TTLOverride = <seconds>(1–255)
```

Upon receiving a client's request packet, the bridge/router assigns the packet a time-to-live (TTL) value. The bridge/router assigns the lowest TTL value among the following possible sources:

- The TTL value of the incoming request packet minus one
- The TTL value configured by the -UDPHELP TTLOverride parameter
- The TTL value configured by the -IP DefaultTTL parameter

  If the TTL value configured by the -UDPHELP TTLOverride parameter is the lowest, the bridge/router forwards the packet with the TTL value configured by this parameter, which overrides the other TTL values.

For more information on the UDPHELP Service parameters used in this procedure, refer to Chapter 62 in *Reference for NETBuilder Family Software*. For more information on the -IP DefaultTTL parameter, refer to Chapter 29 in *Reference for NETBuilder Family Software*.

Relaying BOOTP and DHCP Traffic

UDP Broadcast Helper allows you to set up BOOTP and DHCP so clients can boot from an unspecified server, which may be located through a router and on another network segment. The bridge/router forwards the BOOTPREQUEST packet and DHCP messages from a booting client to a server that can respond with the client's IP address.

If your network is quickly growing or changing, you may want to use the UDP Broadcast Helper for BOOTP instead of configuring a client to boot from one particular server, and then have to reconfigure the client to boot from another server if the network configuration changes.

By supporting both the BOOTP and DHCP relay agents, the bridge/router software allows existing BOOTP clients to interoperate with DHCP servers. BOOTP and DHCP clients and their associated servers often times do not reside on the same IP network or subnetwork. If the bridge/router software does not provide support for a relay agent, every subnet that has BOOTP and DHCP clients is required to have a BOOTP and DHCP server.

Prerequisites Before beginning this procedure, complete the following tasks:

- Log on to the bridge/router with Network Manager privilege.
- Set up ports and paths according to Chapter 1.
- Set up the client and server LANs and WANs according to Chapter 6.
- Examine your network configuration and determine which bridge/routers UDP Broadcast Helper for BOOTP should be enabled upon.
- Determine which networks or servers should receive BOOTPREQUEST packets.
- If possible, determine the IP addresses of the networks or servers that should receive BOOTPREQUEST packets.

Procedure To configure UDP Broadcast Helper for BOOTP and DHCP, see Figure 20-2 and follow these steps:

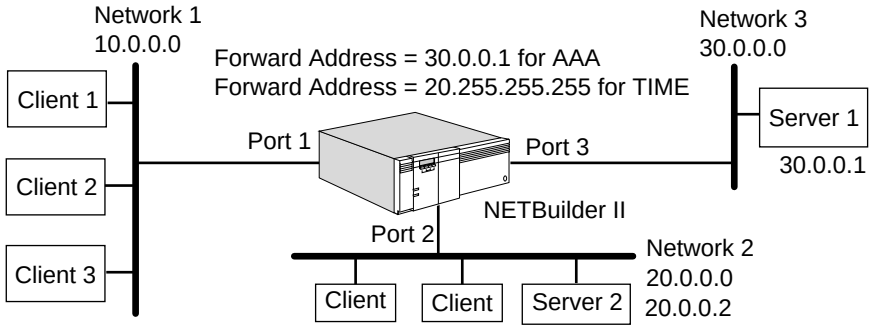


Figure 20-2 Configuring UDP Broadcast Helper for BOOTP

1 Enable UDP Broadcast Helper by entering:

```
SETDefault -UDPHELP CONTrol = Enable
```

2 Add a UDP port for the BOOTP or DHCP server to the active ports list.

You can specify either the built-in name BPSERVER or the UDP port number 67, which is mapped to built-in name BPSERVER. Both BOOTP and DHCP use the same UDP port numbers.

Enter either:

```
ADD -UDPHELP ActivePorts bpserver
```

or

```
ADD -UDPHELP ActivePorts 67
```

3 For UDP port 67 or BPSERVER, 3Com recommends that you set up a list of networks and servers that should receive the BOOTPREQUEST broadcast packets.

If your bridge/router is configured to boot from a server that is accessed through an X.25, Frame Relay, or SMDS interface, you must perform this step. The bridge/router does not rebroadcast BOOTPREQUEST packets over X.25, Frame Relay, or SMDS interfaces.

For an SMDS network, the group address functions as a LAN broadcast.

For X.25 and Frame Relay networks, the router duplicates the packet and forwards it to each configured or dynamically learned neighbor.

You need to configure the ForwardAddress parameter to eliminate unnecessary LAN broadcast packets using one of the following syntaxes:

```
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address>
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address>
 <subnet mask>
ADD -UDPHELP ForwardAddress <UDP port or name> <IP address>
 <subnet mask> [Ones | Zeroes]
ADD -UDPHELP ForwardAddress <UDP port or name> <list of
 interfaces>
```

You can add up to 32 addresses to the forward address list.

If you know the specific IP address of the server (or the network IP address where the servers resides) from which the client should obtain its IP address, add the address to the list.

For example, if the address of the server that responds to the BOOTPREQUEST packets is 10.1.0.1, you can add this address to the list by entering:

```
ADD -UDPHELP ForwardAddress 67 10.1.0.1
```

In the next two examples, you can specify the mnemonic name BPSERVER instead of 67.

To forward BOOTPREQUEST packets to all servers on a specific network, enter:

```
ADD -UDPHELP ForwardAddress 67 10.0.0.0
```

The bridge/router stores address 10.255.255.255 in the list, meaning that all servers (hosts) on network 10 will receive the BOOTPREQUEST packet.

4 Optionally, configure the bridge/router to detect unauthorized BOOTP and DHCP servers using:

```
ADD -UDPHELP AuthDHCPServer <IP address>
```

Specify the addresses of authorized servers. You can add up to 32 servers to the list.

Any BOOTPREPLY or DHCP OFFER packet received with an IP source address that does not match any server's IP address on the list is discarded, a system message is entered, and an SNMP trap is sent. For information about the trap, refer to "AuthDHCPServer" on page 62-2 in *Reference for NETBuilder Family Software*.

For more information on the parameters used in this procedure, refer to Chapter 62 in *Reference for NETBuilder Family Software*.

This completes the basic configuration for UDP Broadcast Helper for BOOTP and DHCP. Information on customizing the configuration of UDP Broadcast Helper for BOOTP is described later in this chapter.

Verifying the Configuration

This section summarizes the commands you need to know to verify UDP Broadcast Helper (including UDP Broadcast Helper for BOOTP) configuration and obtain related statistics.

Checking Parameter Settings

You can check the settings of all parameters associated with UDP Broadcast Helper and UDP Broadcast Helper for BOOTP by entering:

```
SHow -UDPHELP CONFiguration
```

Getting Statistics

You can obtain statistics related to UDP Broadcast Helper and BOOTP by entering:

```
SHow -SYS STATistics -UDPHELP
```

Statistics for UDP Broadcast Helper are displayed. For information on the elements of the display, refer to Appendix H.

Customizing the Configuration for BOOTP

You can customize UDP Broadcast Helper for BOOTP configuration by configuring the BootpMaxHops and BootpThreshold parameters in the UDPHELP Service. The BootpMaxHops parameter limits the number of hops that a BOOTPREQUEST packet can make on a network. The BootpThreshold parameter prioritizes and forwards BOOTPREQUEST packets to a server according to a predetermined plan and determines which clients are booted first.

Limiting the Number of Hops

By configuring the BootpMaxHops parameter and limiting the number of hops, you can control how far a BOOTPREQUEST packet can travel on a network. For example, if your network configuration is similar to that shown in Figure 20-3, you can set the BootpMaxHops value on bridge/router C so that clients in a given area of the network can only boot from a specific server or servers.

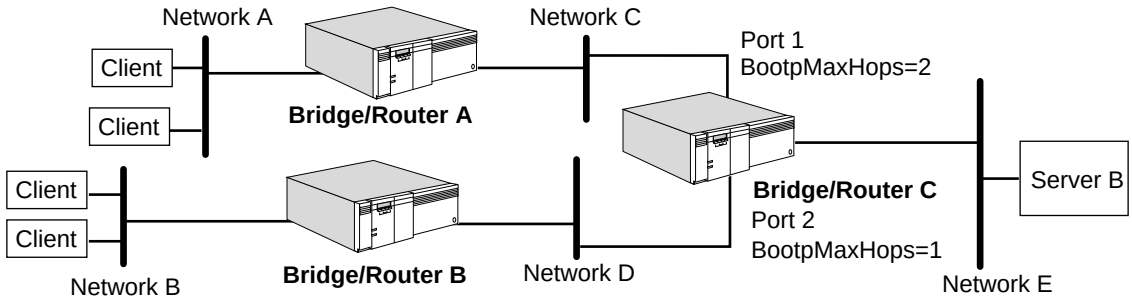


Figure 20-3 Limiting the Number of Hops for BOOTPREQUEST Packets

Prerequisites

Before beginning the procedure, make sure that you have configured UDP Broadcast Helper for BOOTP as described earlier in this chapter.

Procedure

For the following procedure, assume that a client on Network A needs to send BOOTPREQUEST packets to server B on network E. Because you do not know the IP address of server B and you have not configured the ForwardAddress parameter on any of the bridge/routers, each bridge/router will continue to

forward the packet out each of its ports and flood the network with packets. To control this flood of packets, you can configure the BootpMaxHops parameter as follows:

1 On port 1 of bridge/router C, configure the BootpMaxHops parameter to 2 by entering:

```
SETDefault !1 -UDPHELP BootpMaxHops = 2
```

2 On port 2 of bridge/router C, configure the BootpMaxHops parameter to 1 by entering:

```
SETDefault !2 -UDPHELP BootpMaxHops = 1
```

When bridge/router C receives BOOTREQUEST packets from the clients on network A, it forwards the packets to the server on network E. However, bridge/router C receives and discards the BOOTPREQUEST packets from the clients on network B because the BootpMaxHops parameter value is set to 1 on port 2. Bridge/Router C discards the BOOTPREQUEST packets because the packets have already traversed one gateway, which is bridge/router B.

For additional information on the BootpMaxHops parameter, refer to Chapter 62 in *Reference for NETBuilder Family Software*.

Determining Order of Booting

By configuring the BootpThreshold parameter in the UDPHELP Service, you can determine which clients are booted first. For example, if your network configuration is similar to that shown in Figure 20-4, you can set the BootpThreshold value on each bridge/router port so that clients are booted according to a predetermined plan.

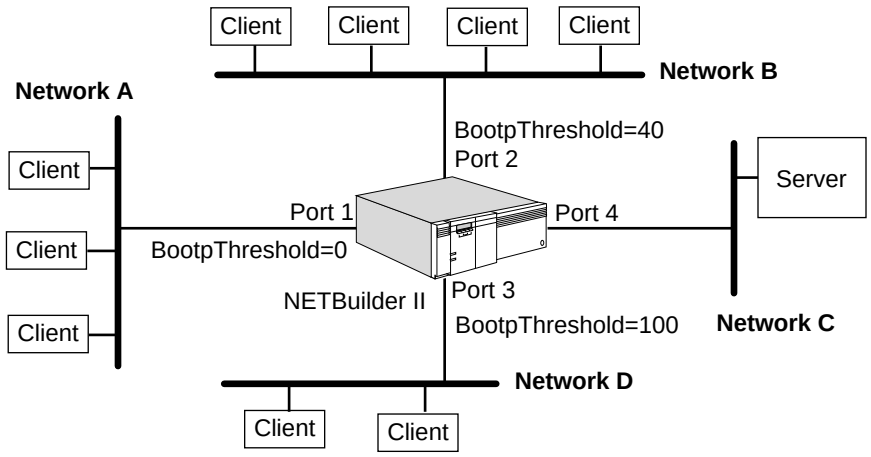


Figure 20-4 Determining Which Clients are Booted First

Prerequisites

Before beginning this procedure, complete the following tasks:

- Configure UDP Broadcast Helper for BOOTP as described earlier in this chapter.
- Determine which clients you want to boot first.

Procedure

For the following procedure, assume that according to your predetermined plan, you want the clients on network A to be booted first, then the clients on network B, and then the clients on network D. You need to set the

BootpThreshold parameter on bridge/router ports 1, 2, and 3 to different values so that the bridge/router will prioritize and forward the BOOTPREQUEST packets to the server in the proper order. To determine which clients are booted first, follow these steps:

1 Set the BootpThreshold value on port 1 to the lowest value of all three ports.

To change the setting, enter:

```
SETDefault !1 -UDPHELP BootpThreshold = 0
```

2 Set the BootpThreshold value on port 2 to the next lowest value of all three ports by entering:

```
SETDefault !2 -UDPHELP BootpThreshold = 40
```

3 Set the BootpThreshold value on port 3 to a value greater than that set for ports 1 and 2 by entering:

```
SETDefault !3 -UDPHELP BootpThreshold = 100
```

When all the clients send out BOOTPREQUEST packets (the Seconds Elapsed Field in the BOOTPREQUEST packet is initially set to 0) at the same time, the bridge/router forwards the packets received on port 1 because the Seconds Elapsed Field and BootpThreshold value match. The bridge/router discards the packets received on port 2 and 3 because the Seconds Elapsed Field in these packets is less than the BootpThreshold value configured for ports 2 and 3.

The clients on networks B and D increase the Seconds Elapsed Field value in the BOOTPREQUEST packets and resend the packets. When the Seconds Elapsed Field value is greater than or equal to the BootpThreshold value on port 2, the bridge/router forwards the packets from the clients on network B to the server on network C. The bridge/router continues to discard the BOOTPREQUEST packets from network D until the Seconds Elapsed Field value is greater than or equal to the BootpThreshold value for port 3.

For additional information on the BootpThreshold parameter, refer to Chapter 62 in *Reference for NETBuilder Family Software*.

How UDP Broadcast Helper Works

When boot servers are located through a router on another network, UDP Broadcast Helper helps BOOTP and DHCP clients to locate the server and retrieve address, configuration, and name information. Without the implementation of this feature, broadcast packets will not be propagated outside of the same network.

Broadcast packets generally do not traverse a router; however, there are some situations in which it is useful to propagate broadcast packets to other networks.

For example, in the topology shown in Figure 20-5, a client on network 129.1.0.0 may require access to a time server on network 129.2.0.0. Normally broadcast requests from the client on network 129.1.0.0 would not be forwarded to servers on network 129.2.0.0; however, you can configure UDP Broadcast Helper to allow the forwarding of broadcast requests to servers on network 129.2.0.0.

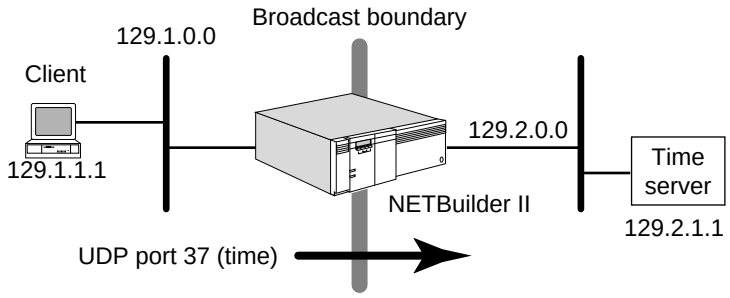


Figure 20-5 Sample UDP Broadcast Helper Topology

UDP applications are identified within a packet by "well-known" port numbers. You can configure the bridge/router to allow broadcast packets to well-known port 37, which is the port number mapped to built-in name TIME for the time service, through to network 129.2.0.0.

BOOTP and DHCP Protocols

The BOOTP Protocol is built on the client-server model and allows a single BOOTP reply to specify many items needed for a client to boot, including the client IP address, the address of a gateway, and the address of a server.

The DHCP Protocol is an extension of the BOOTP Protocol and is also built on the client-server model. DHCP is specifically designed for servers in large network environments that have nomadic users and complex TCP/IP software configurations.

DHCP not only allows a host to automatically allocate reusable IP addresses and additional configuration parameters needed for client operations, it also allows the client/server host to configure host parameters not directly related to the IP Protocol. This feature allows the host to exchange packets with any other host on the Internet. However, DHCP does *not* register newly configured hosts with the Domain Name System and is *not* used to configure routers.

The 3Com implementation of UDP Broadcast Helper feature includes the BOOTP and DHCP relay agent, which allows clients and their associated servers not residing on the same IP network or subnetwork to communicate. Without the relay agent, every subnet that has BOOTP and DHCP clients would be required to have a BOOTP and DHCP server.

Although the BOOTP and DHCP Protocols uses the same UDP port numbers (67 and 68), they have some important differences as follows:

- DHCP allows IP addresses to be "leased" for a fixed length of time.

  Groups of hosts that do not need permanent IP addresses can lease an address from a limited pool of addresses. Also, a host that is only temporarily connected to the network can be assigned an IP address because the addresses can be reused when they are no longer needed by the original host.

- DHCP packet length is longer than BOOTP.

  The additional packet length allows a DHCP server to provide the client with all the IP configuration parameters that it needs to operate.

- DHCP is a more complicated protocol than BOOTP.

  DHCP has seven message types; BOOTP uses only two. In addition, DHCP requires complex state machines.

INDEX

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z