Cisco IOS Novell IPX Configuration Guide

Release 12.2SX

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Cisco IOS Novell IPX Configuration Guide

About Cisco IOS and Cisco IOS XE Software Documentation

Last updated: December 10, 2008

This document describes the objectives, audience, conventions, and organization used in Cisco IOS and Cisco IOS XE software documentation, collectively referred to in this document as Cisco IOS documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

Documentation Objectives

Cisco IOS documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS documentation set is intended for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS commands necessary to perform particular tasks. The Cisco IOS documentation set is also intended for those users experienced with Cisco IOS who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS release.

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section includes the following topics:

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
string	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to *public*, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
italic	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
...	An ellipsis (three consecutive nonbolded periods without spaces) after a syntax element indicates that the element can be repeated.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y | z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS uses the following program code conventions:

Convention	Description
`Courier font`	Courier font is used for information that is displayed on a PC or terminal screen.
`Bold Courier font`	Bold Courier font indicates text that the user must enter.
< >	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

The Cisco IOS documentation set uses the following conventions for reader alerts:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. Included are lists of configuration guides, command references, and supplementary references and resources that make up the documentation set. The following topics are included:

Cisco IOS Documentation Set

Cisco IOS documentation consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and severity 3 (moderate) defects in released Cisco IOS code. Review release notes before other documents to learn whether or not updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
  - Configuration guides—Compilations of documents that provide informational and task-oriented descriptions of Cisco IOS features.
  - Command references—Compilations of command pages that provide detailed information about the commands used in the Cisco IOS features and processes that make up the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the documentation organization and how to access various document types.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Cisco IOS XE, and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books describe Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are provided by technology. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in Table 2 on page xiii.

Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS and Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references are comprehensive, meaning that they include commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

For additional information about configuring and operating specific networking devices, go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS AppleTalk Configuration Guide *Cisco IOS XE AppleTalk Configuration Guide* *Cisco IOS AppleTalk Command Reference*	AppleTalk protocol.
Cisco IOS Asynchronous Transfer Mode Configuration Guide *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Bridging and IBM Networking Configuration Guide *Cisco IOS Bridging Command Reference* *Cisco IOS IBM Networking Command Reference*	• Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). • Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
Cisco IOS Broadband and DSL Configuration Guide *Cisco IOS XE Broadband and DSL Configuration Guide* *Cisco IOS Broadband and DSL Command Reference*	Point-to-Point Protocol (PPP) over ATM (PPPoA) and PPP over Ethernet (PPPoE).
Cisco IOS Carrier Ethernet Configuration Guide *Cisco IOS Carrier Ethernet Command Reference*	Connectivity fault management (CFM), Ethernet Local Management Interface (ELMI), IEEE 802.3ad link bundling, Link Layer Discovery Protocol (LLDP), media endpoint discovery (MED), and operations, administration, and maintenance (OAM).
Cisco IOS Configuration Fundamentals Configuration Guide *Cisco IOS XE Configuration Fundamentals Configuration Guide* *Cisco IOS Configuration Fundamentals Command Reference*	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
Cisco IOS DECnet Configuration Guide *Cisco IOS XE DECnet Configuration Guide* *Cisco IOS DECnet Command Reference*	DECnet protocol.
Cisco IOS Dial Technologies Configuration Guide *Cisco IOS XE Dial Technologies Configuration Guide* *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), large scale dialout, dial-on-demand routing, dialout, modem and resource pooling, ISDN, multilink PPP (MLP), PPP, virtual private dialup network (VPDN).
Cisco IOS Flexible NetFlow Configuration Guide *Cisco IOS Flexible NetFlow Command Reference*	Flexible NetFlow.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS H.323 Configuration Guide	Gatekeeper enhancements for managed voice services, Gatekeeper Transaction Message Protocol, gateway codec order preservation and shutdown control, H.323 dual tone multifrequency relay, H.323 version 2 enhancements, Network Address Translation (NAT) support of H.323 v2 Registration, Admission, and Status (RAS) protocol, tokenless call authorization, and VoIP gateway trunk and carrier-based routing.
Cisco IOS High Availability Configuration Guide *Cisco IOS XE High Availability Configuration Guide* *Cisco IOS High Availability Command Reference*	A variety of High Availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
Cisco IOS Integrated Session Border Controller Command Reference	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
Cisco IOS Intelligent Services Gateway Configuration Guide *Cisco IOS Intelligent Services Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, session state monitoring.
Cisco IOS Interface and Hardware Component Configuration Guide *Cisco IOS XE Interface and Hardware Component Configuration Guide* *Cisco IOS Interface and Hardware Component Command Reference*	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
Cisco IOS IP Addressing Services Configuration Guide *Cisco IOS XE Addressing Services Configuration Guide* *Cisco IOS IP Addressing Services Command Reference*	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
Cisco IOS IP Application Services Configuration Guide *Cisco IOS XE IP Application Services Configuration Guide* *Cisco IOS IP Application Services Command Reference*	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
Cisco IOS IP Mobility Configuration Guide *Cisco IOS IP Mobility Command Reference*	Mobile ad hoc networks (MANet) and Cisco mobile networks.
Cisco IOS IP Multicast Configuration Guide *Cisco IOS XE IP Multicast Configuration Guide* *Cisco IOS IP Multicast Command Reference*	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS IP Routing Protocols Configuration Guide *Cisco IOS XE IP Routing Protocols Configuration Guide* *Cisco IOS IP Routing Protocols Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast, bidirectional forwarding detection (BFD), Enhanced Interior Gateway Routing Protocol (EIGRP), Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), on-demand routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
Cisco IOS IP SLAs Configuration Guide *Cisco IOS XE IP SLAs Configuration Guide* *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
Cisco IOS IP Switching Configuration Guide *Cisco IOS XE IP Switching Configuration Guide* *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
Cisco IOS IPv6 Configuration Guide *Cisco IOS XE IPv6 Configuration Guide* *Cisco IOS IPv6 Command Reference*	For IPv6 features, protocols, and technologies, go to the IPv6 "Start Here" document at the following URL: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.html
Cisco IOS ISO CLNS Configuration Guide *Cisco IOS XE ISO CLNS Configuration Guide* *Cisco IOS ISO CLNS Command Reference*	ISO connectionless network service (CLNS).
Cisco IOS LAN Switching Configuration Guide *Cisco IOS XE LAN Switching Configuration Guide* *Cisco IOS LAN Switching Command Reference*	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
Cisco IOS Mobile Wireless Home Agent Configuration Guide *Cisco IOS Mobile Wireless Home Agent Command Reference*	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*	Cisco IOS radio access network products.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Multiprotocol Label Switching Configuration Guide *Cisco IOS XE Multiprotocol Label Switching Configuration Guide* *Cisco IOS Multiprotocol Label Switching Command Reference*	MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.
Cisco IOS Multi-Topology Routing Configuration Guide *Cisco IOS Multi-Topology Routing Command Reference*	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
Cisco IOS NetFlow Configuration Guide *Cisco IOS XE NetFlow Configuration Guide* *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, export features.
Cisco IOS Network Management Configuration Guide *Cisco IOS XE Network Management Configuration Guide* *Cisco IOS Network Management Command Reference*	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS Software (XSM Configuration).
Cisco IOS Novell IPX Configuration Guide *Cisco IOS XE Novell IPX Configuration Guide* *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.
Cisco IOS Optimized Edge Routing Configuration Guide *Cisco IOS Optimized Edge Routing Command Reference*	Optimized edge routing (OER) monitoring, policy configuration, routing control, logging and reporting, and VPN IPsec/generic routing encapsulation (GRE) tunnel interface optimization.
Cisco IOS Quality of Service Solutions Configuration Guide *Cisco IOS XE Quality of Service Solutions Configuration Guide* *Cisco IOS Quality of Service Solutions Command Reference*	Class-based weighted fair queuing (CBWFQ), custom queuing, distributed traffic shaping (DTS), generic traffic shaping (GTS), IP- to-ATM class of service (CoS), low latency queuing (LLQ), modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), priority queuing, Security Device Manager (SDM), Multilink PPP (MLPPP) for QoS, header compression, AutoQoS, QoS features for voice, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).
Cisco IOS Security Configuration Guide *Cisco IOS XE Security Configuration Guide* *Cisco IOS Security Command Reference*	Access control lists (ACLs), authentication, authorization, and accounting (AAA), firewalls, IP security and encryption, neighbor router authentication, network access security, network data encryption with router authentication, public key infrastructure (PKI), RADIUS, TACACS+, terminal access security, and traffic filters.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Service Selection Gateway Configuration Guide *Cisco IOS Service Selection Gateway Command Reference*	Subscriber authentication, service access, and accounting.
Cisco IOS Software Activation Configuration Guide *Cisco IOS Software Activation Command Reference*	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
Cisco IOS Software Modularity Installation and Configuration Guide *Cisco IOS Software Modularity Command Reference*	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes and patches.
Cisco IOS Terminal Services Configuration Guide *Cisco IOS Terminal Services Command Reference* *Cisco IOS XE Terminal Services Command Reference*	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
Cisco IOS Virtual Switch Command Reference	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). **Note** For information about virtual switch configuration, refer to the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
Cisco IOS Voice Configuration Library *Cisco IOS Voice Command Reference*	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
Cisco IOS VPDN Configuration Guide *Cisco IOS XE VPDN Configuration Guide* *Cisco IOS VPDN Command Reference*	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy, L2TP extended failover, L2TP security VPDN, multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82: tunnel assignment ID, shell-based authentication of VPDN users, tunnel authentication via RADIUS on tunnel terminator.
Cisco IOS Wide-Area Networking Configuration Guide *Cisco IOS XE Wide-Area Networking Configuration Guide* *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay, Layer 2 Tunneling Protocol Version 3 (L2TPv3), Link Access Procedure, Balanced (LAPB), Switched Multimegabit Data Service (SMDS), and X.25.
Cisco IOS Wireless LAN Configuration Guide *Cisco IOS Wireless LAN Command Reference*	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS releases.
Cisco IOS New, Modified, Removed, and Replaced Commands	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
Cisco IOS Software System Messages	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system; be informational only; or may help diagnose problems with communications lines, internal hardware, or the system software.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at the following URL: http://www.cisco.com/go/mibs
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is published monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

Using the Command-Line Interface in Cisco IOS and Cisco IOS XE Software

Last updated: December 10, 2008

This document provides basic information about the command-line interface (CLI) in Cisco IOS and Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

For more information about using the CLI, see the "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About Cisco IOS and Cisco IOS XE Software Documentation" document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note The AUX port on the Route Processor (RP) installed in a Cisco ASR1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 CLI Command Modes

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
User EXEC	Log in.	`Router>`	Issue the **logout** or **exit** command.	• Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the **enable** command.	`Router#`	Issue the **disable** command or the **exit** command to return to user EXEC mode.	• Issue **show** and **debug** commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the **configure terminal** command.	`Router(config)#`	Issue the **exit** command or the **end** command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the **interface** command.	`Router(config-if)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the **line vty** or **line console** command.	`Router(config-line)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon # >` The # symbol represents the line number and increments at each prompt.	Issue the **continue** command.	• Run as the default operating mode when a valid image cannot be loaded. • Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. • Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on the Cisco ASR1000 series router)	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload. • A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode. • The router was accessed using an RP auxiliary port. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.	`Router(diag)#`	If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.	• Inspect various states on the router, including the Cisco IOS state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS software or other processes. • Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or possibly other hardware components. • Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
cont                continue executing a downloaded image
context             display the context of a loaded image
cookie              display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

Note A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 2 describes how to use the Help feature.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the help feature in any command mode.
?	Lists all commands available for a particular command mode.
*partial command***?**	Provides a list of commands that begin with the character string (no space between the command and the question mark).
partial command<**Tab**>	Completes a partial command name (no space between the command and <Tab>).
command **?**	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
command keyword **?**	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help

Help may be requested at any point in a command by entering a question mark '?'.  If
nothing matches, the help list will be empty and you must backup until entering a '?'
shows the available options.

Two styles of help are provided:

1. Full help is available when you are ready to enter a command argument (e.g. 'show ?')
and describes each possible argument.

2. Partial help is provided when an abbreviated argument is entered and you want to know
what arguments match the input (e.g. 'show pr?'.)
```

?

```
Router# ?
Exec commands:
  access-enable        Create a temporary access-List entry
  access-profile       Apply user-profile to interface
  access-template      Create a temporary access-List entry
  alps                 ALPS exec commands
  archive              manage archive files
<snip>
```

***partial command*?**

```
Router(config)# zo?
zone  zone-pair
```

***partial command*<Tab>**

```
Router(config)# we<Tab> webvpn
```

***command* ?**

```
Router(config-if)# pppoe ?
  enable        Enable pppoe
  max-sessions  Maximum PPPOE sessions
```

***command keyword* ?**

```
Router(config-if)# pppoe enable ?
  group  attach a BBA group
  <cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 3 describes these conventions.

Table 3 CLI Syntax Conventions

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
  WORD  domain name
Router(config)# ethernet cfm domain dname ?
  level
Router(config)# ethernet cfm domain dname level ?
  <0-7>  maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
  <cr>
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
Router(config)# logging host ?
  Hostname or A.B.C.D  IP address of the syslog server
  ipv6                 Configure IPv6 syslog server
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, "two words" is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The CLI command history feature saves the commands you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the up arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.

- Press Ctrl-N or the down arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.

  Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

  The CLI command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrrp** as a keyword in addition to **version**. (Command and keyword examples from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or **un**	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias** *mode command-alias original-command*. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** and **default** forms of commands are described in the command pages of command references.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS and Cisco IOS XE software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebug all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.

Caution Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

Three output modifiers are available and are described as follows:

- **begin** *regular expression*—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include** *regular expression*—Displays all lines in which a match of the regular expression is found.
- **exclude** *regular expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression "protocol."

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 5 shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at "^" marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following documents:

- *Cisco IOS Release 12.2SR System Message Guide*
- *Cisco IOS System Messages, Volume 1 of 2* (Cisco IOS Release 12.4)
- *Cisco IOS System Messages, Volume 2 of 2* (Cisco IOS Release 12.4)

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html

  or

  "Using Cisco IOS XE Software" chapter of the *Cisco ASR1000 Series Aggregation Services Routers Software Configuration Guide*:

  http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/using_cli.html
- Cisco Product Support Resources

  http://www.cisco.com/web/psa/products/index.html
- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html
- *White Paper: Cisco IOS Reference Guide*

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_white_paper09186a008018305e.shtml
- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com User ID and password)

  http://www.cisco.com/kobayashi/sw-center/
- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software

  http://www.cisco.com/pcgi-bin/Support/Errordecoder/index.cgi

- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)

  http://tools.cisco.com/Support/CLILookup

- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

  https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl

Novell IPX Overview

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

The Cisco IOS software supports a variety of routing protocols. The *Cisco IOS Novell IPX Configuration Guide* discusses Novell IPX network protocols; it contains these sections:

- Novell IPX
- Configuring Novell IPX
- Novell IPX Configuration Examples

The *Cisco IOS IP Configuration Guide* discusses the following network protocols:

- IP
- IP Routing

This overview chapter provides a high-level description of Novell IPX. For configuration information, see the appropriate section in this publication.

For the latest feature information and caveats, see the release notes for your platform and software release. Additionally, use Cisco Feature Navigator to find information about feature, platform, and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

Novell IPX

This section offers background information and briefly describes the Cisco implementation of Novell IPX.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Background on Novell IPX

Novell Internetwork Packet Exchange (IPX) is derived from the Xerox Network Systems (XNS) Internet Datagram Protocol (IDP). IPX and XNS have the following differences:

- IPX and XNS do not always use the same Ethernet encapsulation format.
- IPX uses the Novell proprietary Service Advertising Protocol (SAP) to advertise special network services. File servers and print servers are examples of services that typically are advertised.
- IPX uses delay (measured in ticks) while XNS uses hop count as the primary metric in determining the best path to a destination.

The Cisco Implementation of Novell IPX

The Cisco implementation of the Novell IPX protocol is certified to provide full IPX routing functionality.

IPX MIB Support

Cisco supports the IPX MIB (currently, read-only access is supported). The IPX Accounting group represents one of the local Cisco-specific IPX variables we support. This group provides access to the active database that is created and maintained if IPX accounting is enabled on a router or access server.

IPX Enhanced IGRP Support

Cisco IOS software supports IPX Enhanced IGRP, which provides the following features:

- Automatic redistribution—IPX Routing Information Protocol (RIP) routes are automatically redistributed into Enhanced IGRP, and Enhanced IGRP routes are automatically redistributed into RIP. If desired, you can turn off redistribution. You also can completely turn off Enhanced IGRP and IPX RIP on the device or on individual interfaces.
- Increased network width—With IPX RIP, the largest possible width of your network is 15 hops. When Enhanced IGRP is enabled, the largest possible width is 224 hops. Because the Enhanced IGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport layer hop counter. Cisco works around this problem by incrementing the transport control field only when an IPX packet has traversed 15 routers, and the next hop to the destination was learned via Enhanced IGRP. When a RIP route is being used as the next hop to the destination, the transport control field is incremented as usual.
- Incremental SAP updates—Complete SAP updates are sent periodically on each interface until an Enhanced IGRP neighbor is found, and thereafter only when changes are made to the SAP table. This procedure works by taking advantage of the Enhanced IGRP reliable transport mechanism, which means that an Enhanced IGRP peer must be present for incremental SAPs to be sent. If no peer exists on a particular interface, periodic SAPs will be sent on that interface until a peer is found. This functionality is automatic on serial interfaces and can be configured on LAN media.

LANE Support

Cisco IOS software supports routing IPX between Ethernet-emulated LANs and Token Ring-emulated LANs. For more information on emulated LANs and routing IPX between them, refer to the "Configuring LAN Emulation" chapter of the *Cisco IOS Switching Services Configuration Guide*.

VLAN Support

Cisco IOS software supports routing IPX between VLANs. Users with Novell NetWare environments can configure any one of the four IPX Ethernet encapsulations to be routed using ISL encapsulation across VLAN boundaries. For more information on VLANs and routing IPX between them over ISL, refer to the “Configuring Routing Between VLANs with ISL Encapsulation” chapter of the *Cisco IOS Switching Services Configuration Guide*.

Multilayer Switching Support

Cisco IOS software supports IPX Multilayer Switching (MLS). For more information on IPX MLS, refer to the “Multilayer Switching” chapter of the *Cisco IOS Switching Services Configuration Guide*.

Configuring Novell IPX

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes how to configure Novell Internetwork Packet Exchange (IPX) and provides configuration examples. For a complete description of the IPX commands in this chapter, refer to the *Cisco IOS Novell IPX Command Reference* publication. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

For the latest feature information and caveats, see the release notes for your platform and software release. Additionally, use Cisco Feature Navigator to find information about feature, platform, and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

IPX Addresses

An IPX network address consists of a network number and a node number expressed in the format *network.node*.

Network Numbers

The network number identifies a physical network. It is a 4-byte (32-bit) quantity that must be unique throughout the entire IPX internetwork. The network number is expressed as hexadecimal digits. The maximum number of digits allowed is eight.

The Cisco IOS software does not require that you enter all eight digits; you can omit leading zeros.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Node Numbers

The node number identifies a node on the network. It is a 48-bit quantity, represented by dotted triplets of four-digit hexadecimal numbers.

If you do not specify a node number for a router to be used on WAN links, the Cisco IOS software uses the hardware MAC address currently assigned to it as its node address. This is the MAC address of the first Ethernet, Token Ring, or FDDI interface card. If there are no valid IEEE interfaces, the Cisco IOS software randomly assigns a node number using a number that is based on the system clock.

IPX Address Example

The following example shows how to configure an IPX network address:

```
4a.0000.0c00.23fe
```

In this example, the network number is 4a (more specifically, it is 0000004a), and the node number is 0000.0c00.23fe. All digits in the address are hexadecimal.

IPX Configuration Task List

To configure IPX routing, perform the tasks in the following sections:

- Configuring IPX Routing (Required)
- Configuring IPX Enhanced IGRP (Optional)
- Configuring IPX and SPX over WANs (Optional)
- Controlling Access to IPX Networks (Optional)
- Tuning IPX Network Performance (Optional)
- Shutting Down an IPX Network (Optional)
- Configuring IPX Accounting (Optional)
- Configuring IPX Between LANs (Optional)
- Configuring IPX Between VLANs (Optional)
- Configuring IPX Multilayer Switching (Optional)
- Monitoring and Maintaining the IPX Network (Optional)

See the “Novell IPX Configuration Examples” section at the end of this chapter for configuration examples.

Configuring IPX Routing

You configure IPX routing by first enabling it on the router and then configuring it on each interface.

Optionally, you can route IPX on some interfaces and transparently bridge it on other interfaces. You can also route IPX traffic between routed interfaces and bridge groups, or route IPX traffic between bridge groups.

To configure IPX routing, perform the tasks in the following sections. The first two tasks are required; the rest are optional.

- Enabling IPX Routing (Required)
- Assigning Network Numbers to Individual Interfaces (Required)
- Enabling Concurrent Routing and Bridging (Optional)
- Configuring Integrated Routing and Bridging (Optional)

IPX Default Routes

In IPX, a *default route* is the network where all packets for which the route to the destination address is unknown are forwarded.

Original Routing Information Protocol (RIP) implementations allowed the use of network -2 (0xFFFFFFFE) as a regular network number in a network. With the inception of NetWare Link Services Protocol (NLSP), network -2 is reserved as the default route for NLSP and RIP. Both NLSP and RIP routers should treat network -2 as a default route. Therefore, you should implement network -2 as the default route regardless of whether you configure NLSP in your IPX network.

By default, Cisco IOS software treats network -2 as the default route. You should ensure that your IPX network does not use network -2 as a regular network. If, for some reason, you must use network -2 as a regular network, you can disable the default behavior. To do so, see the "Adjusting Default Routes" section later in this chapter.

For more background information on how to handle IPX default routes, refer to the Novell *NetWare Link Services Protocol (NLSP) Specification, Revision 1.1* publication.

Enabling IPX Routing

The first step in enabling IPX routing is to enable it on the router. If you do not specify the node number of the router to be used on WAN links, the Cisco IOS software uses the hardware MAC address currently assigned to it as its node address. This is the MAC address of the first Ethernet, Token Ring, or FDDI interface card. If there are no valid IEEE interfaces, the Cisco IOS software randomly assigns a node number using a number that is based on the system clock.

To enable IPX routing, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx routing [node]`	Enables IPX routing.

For an example of how to enable IPX routing, see the "IPX Routing Examples" section at the end of this chapter.

Caution If you plan to use DECnet and IPX routing concurrently on the same interface, you should enable DECnet routing first, then enable IPX routing without specifying the optional MAC node number. If you enable IPX before enabling DECnet routing, routing for IPX will be disrupted because DECnet forces a change in the MAC-level node number.

Assigning Network Numbers to Individual Interfaces

After you have enabled IPX routing, you enable IPX routing on the individual interfaces by assigning network numbers to those interfaces.

You enable IPX routing on interfaces that support a single network or multiple networks.

When you enable IPX routing on an interface, you can also specify an encapsulation (frame type) to use for packets being sent on that network. Table 1 lists the encapsulation types you can use on IEEE interfaces and shows the correspondence between Cisco naming conventions and Novell naming conventions for the encapsulation types.

Table 1 Cisco and Novell IPX Encapsulation Names on IEEE Interfaces

Interface Type	Cisco Name	Novell Name
Ethernet	novell-ether (Cisco IOS default) arpa sap snap	Ethernet_802.3 Ethernet_II Ethernet_802.2 Ethernet_Snap
Token Ring	sap (Cisco IOS default) snap	Token-Ring Token-Ring_Snap
FDDI	snap (Cisco IOS default) sap novell-fddi	Fddi_Snap Fddi_802.2 Fddi_Raw

Note The SNAP encapsulation type is not supported and should not be configured on any IPX interfaces that are attached to a FDDI-Ethernet bridge.

Assigning Network Numbers to Individual Interfaces Task List

The following sections describe how to enable IPX routing on interfaces that support a single network and on those that support multiple networks. To enable IPX routing on an interface, you must perform one of the tasks:

- Assigning Network Numbers to Interfaces That Support a Single Network (Required)
- Assigning Network Numbers to Interfaces That Support Multiple Networks (Required)
- Setting the Encapsulation Type for Subinterfaces (Required)

Assigning Network Numbers to Interfaces That Support a Single Network

A single interface can support a single network or multiple logical networks. For a single network, you can configure any encapsulation type. Of course, it should match the encapsulation type of the servers and clients using that network number.

To assign a network number to an interface that supports a single network, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx network` *network* `[encapsulation` *encapsulation-type*`]`	Enables IPX routing on an interface.

If you specify an encapsulation type, be sure to choose the one that matches the one used by the servers and clients on that network. Novell-ether or ARPA encapsulations cannot be used for FDDI-Ethernet bridged IPX traffic. Use SAP encapsulations on originating and destination IPX interfaces that are attached to the FDDI-Ethernet bridge. See Table 1 for a list of encapsulation types you can use on IEEE interfaces.

For an example of how to enable IPX routing, see the "IPX Routing Examples" section at the end of this chapter.

Assigning Network Numbers to Interfaces That Support Multiple Networks

When assigning network numbers to an interface that supports multiple networks, you must specify a different encapsulation type for each network. Because multiple networks share the physical medium, the Cisco IOS software is allowed to identify the packets that belong to each network. For example, you can configure up to four IPX networks on a single Ethernet cable, because four encapsulation types are supported for Ethernet. Remember, the encapsulation type should match the servers and clients using the same network number. See Table 1 for a list of encapsulation types you can use on IEEE interfaces.

There are two ways to assign network numbers to interfaces that support multiple networks. You can use subinterfaces or primary and secondary networks.

Setting the Encapsulation Type for Subinterfaces

You typically use subinterfaces to assign network numbers to interfaces that support multiple networks.

A *subinterface* is a mechanism that allows a single physical interface to support multiple logical interfaces or networks. That is, several logical interfaces or networks can be associated with a single hardware interface. Each subinterface must use a distinct encapsulation, and the encapsulation must match that of the clients and servers using the same network number.

Note When enabling NLSP and configuring multiple encapsulations on the same physical LAN interface, you must use subinterfaces. You cannot use secondary networks.

Any interface configuration parameters that you specify on an individual subinterface are applied to that subinterface only.

To configure multiple IPX networks on a physical interface using subinterfaces, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number.subinterface-number`	Specifies a subinterface.
Step 2	`Router(config-if)# ipx network network [encapsulation encapsulation-type]`	Enables IPX routing, specifying the first encapsulation type.

Note You cannot configure more than 200 IPX interfaces on a router using the **ipx network** command.

To configure more than one subinterface, repeat these two steps. See Table 1 for a list of encapsulation types you can use on IEEE interfaces.

For examples of configuring multiple IPX networks on an interface, see the "IPX Routing on Multiple Networks Examples" section at the end of this chapter.

Primary and Secondary Networks

When assigning network numbers to interfaces that support multiple networks, you can also configure primary and secondary networks.

The first logical network you configure on an interface is considered the *primary network*. Any additional networks are considered *secondary networks*. Again, each network on an interface must use a distinct encapsulation and it should match that of the clients and servers using the same network number.

Any interface configuration parameters that you specify on this interface are applied to all the logical networks. For example, if you set the routing update timer to 120 seconds, this value is used on all four networks.

To use primary and secondary networks to configure multiple IPX networks on an interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ipx network` *network* `[encapsulation` *encapsulation-type*`]`	Enables IPX routing on the primary network.
Step 2	`Router(config-if)# ipx network` *network* `[encapsulation` *encapsulation-type*`] [secondary]`	Enables IPX routing on a secondary network.

To configure more than one secondary network, repeat these steps as appropriate. See Table 1 for a list of encapsulation types you can use on IEEE interfaces.

Note When enabling NLSP and configuring multiple encapsulations on the same physical LAN interface, you must use subinterfaces. You cannot use secondary networks.

Enabling Concurrent Routing and Bridging

You can route IPX on some interfaces and transparently bridge it on other interfaces simultaneously. To enable this type of routing, you must enable concurrent routing and bridging. To enable concurrent routing and bridging, use the following command in global configuration mode:

Command	Purpose
`Router(config)# bridge crb`	Enables the Cisco IOS software to both route and bridge a given protocol on separate interfaces within a single router.

Configuring Integrated Routing and Bridging

Integrated routing and bridging (IRB) enables a user to route IPX traffic between routed interfaces and bridge groups, or route IPX traffic between bridge groups. Specifically, local or unroutable traffic is bridged among the bridged interfaces in the same bridge group. Routable traffic is routed to other routed interfaces or bridge groups. Using IRB, you can do the following:

- Switch packets from a bridged interface to a routed interface
- Switch packets from a routed interface to a bridged interface
- Switch packets within the same bridge group

For more information about configuring integrated routing and bridging, refer to the "Configuring Transparent Bridging" chapter in the *Cisco IOS Bridging and IBM Networking Configuration Guide*.

Configuring IPX Enhanced IGRP

Enhanced IGRP is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. Enhanced IGRP uses the same distance vector algorithm and distance information as IGRP. However, the convergence properties and the operating efficiency of Enhanced IGRP have improved significantly over IGRP.

The convergence technology is based on research conducted at SRI International and employs an algorithm referred to as the Diffusing Update Algorithm (DUAL). This algorithm guarantees loop-free operation at every instant throughout a route computation, and allows all routers involved in a topology change to synchronize at the same time. Routers that are not affected by topology changes are not involved in recomputations. The convergence time with DUAL rivals that of any other existing routing protocol.

Enhanced IGRP Features

Enhanced IGRP offers the following features:

- Fast convergence—The DUAL algorithm allows routing information to converge as quickly as any currently available routing protocol.
- Partial updates—Enhanced IGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for Enhanced IGRP packets.
- Less CPU usage than IGRP—Full update packets need not be processed each time they are received.
- Neighbor discovery mechanism—This feature is a simple hello mechanism used to learn about neighboring routers. It is protocol-independent.
- Scaling—Enhanced IGRP scales to large networks.

Enhanced IGRP Components

Enhanced IGRP has four basic components discussed in the following sections:

- Neighbor Discovery/Recovery
- Reliable Transport Protocol
- DUAL Finite-State Machine
- Protocol-Dependent Modules

Neighbor Discovery/Recovery

Neighbor discovery/recovery is the process that routers use to dynamically learn of other routers on their directly attached networks. Routers must also discover when their neighbors become unreachable or inoperative. The router achieves neighbor discovery/recovery with low overhead by periodically sending

small hello packets. As long as hello packets are received, a router can determine that a neighbor is alive and functioning. Once this status is determined, the neighboring devices can exchange routing information.

Reliable Transport Protocol

The reliable transport protocol is responsible for guaranteed, ordered delivery of Enhanced IGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some Enhanced IGRP packets must be sent reliably, and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities (such as Ethernet) it is not necessary to send hellos reliably to all neighbors individually. Therefore, Enhanced IGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets (such as updates) require acknowledgment, which is indicated in the packet. The reliable transport has a provision to send multicast packets quickly when there are unacknowledged packets pending. This provision helps ensure that convergence time remains low in the presence of varying speed links.

DUAL Finite-State Machine

The DUAL finite-state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses the distance information (known as a metric) to select efficient, loop-free paths. DUAL selects routes to be inserted into a routing table based on feasible successors. A *successor* is a neighboring router used for packet forwarding that has a least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no feasible successors but there are neighbors advertising the destination, a recomputation must occur. This is the process whereby a new successor is determined. The amount of time it takes to recompute the route affects the convergence time. Recomputation is processor-intensive. It is advantageous to avoid recomputation if it is not necessary. When a topology change occurs, DUAL will test for feasible successors. If there are feasible successors, it will use any it finds in order to avoid unnecessary recomputation.

Protocol-Dependent Modules

The protocol-dependent modules are responsible for network layer protocol-specific tasks. They are also responsible for parsing Enhanced IGRP packets and informing DUAL of the new information received. Enhanced IGRP asks DUAL to make routing decisions, but the results are stored in the IPX routing table. Also, Enhanced IGRP is responsible for redistributing routes learned by other IPX routing protocols.

IPX Enhanced IGRP Configuration Task List

To enable IPX Enhanced IGRP, perform the tasks in the following sections. Only the first task is required; the remaining tasks are optional.

- Enabling IPX Enhanced IGRP (Required)
- Customizing Link Characteristics (Optional)
- Customizing the Exchange of Routing and Service Information (Optional)
- Querying the Backup Server (Optional)

Enabling IPX Enhanced IGRP

To create an IPX Enhanced IGRP routing process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router eigrp autonomous-system-number`	Enables an Enhanced IGRP routing process.
Step 2	`Router(config-if)# network {network-number | all}`	Enables Enhanced IGRP on a network.

To associate multiple networks with an Enhanced IGRP routing process, you can repeat the preceding two steps.

For an example of how to enable Enhanced IGRP, see the "IPX Enhanced IGRP Example" section at the end of this chapter.

Customizing Link Characteristics

You might want to customize the Enhanced IGRP link characteristics. The following sections describe these customization tasks:

- Configuring the Percentage of Link Bandwidth Used by Enhanced IGRP (Optional)
- Configuring Maximum Hop Count (Optional)
- Adjusting the Interval Between Hello Packets and the Hold Time (Optional)

Configuring the Percentage of Link Bandwidth Used by Enhanced IGRP

By default, Enhanced IGRP packets consume a maximum of 50 percent of the link bandwidth, as configured with the **bandwidth** interface subcommand. If a different value is desired, use the **ipx bandwidth-percent** command. This command may be useful if a different level of link utilization is required, or if the configured bandwidth does not match the actual link bandwidth (it may have been configured to influence route metric calculations).

To configure the percentage of bandwidth that may be used by Enhanced IGRP on an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx bandwidth-percent eigrp as-number percent`	Configures the percentage of bandwidth that may be used by Enhanced IGRP on an interface.

For an example of how to configure the percentage of Enhanced IGRP bandwidth, see the "IPX Enhanced IGRP Bandwidth Configuration Example" section at the end of this chapter.

Configuring Maximum Hop Count

Note Although adjusting the maximum hop count is possible, it is not recommended for Enhanced IGRP. We recommend that you use the default value for the maximum hop count of Enhanced IGRP.

By default, IPX packets whose hop count exceeds 15 are discarded. In larger internetworks, this maximum hop count may be insufficient. You can increase the hop count to a maximum of 254 hops for Enhanced IGRP. To modify the maximum hop count, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx maximum-hops` *hop*	Sets the maximum number of hops of an IPX packet reachable by non-RIP routing protocols. Also sets the maximum number of routers that an IPX packet can traverse before being dropped.

Adjusting the Interval Between Hello Packets and the Hold Time

You can adjust the interval between hello packets and the hold time.

Routers periodically send hello packets to each other to dynamically learn of other devices on their directly attached networks. Routers use this information to discover their neighbors and to discover when their neighbors become unreachable or inoperative.

By default, hello packets are sent every 5 seconds. The exception is on low-speed, nonbroadcast multiaccess (NBMA) media, where the default hello interval is 60 seconds. Low speed is considered to be a rate of T1 or slower, as specified with the **bandwidth** interface configuration command. The default hello interval remains 5 seconds for high-speed NBMA networks.

Note For the purposes of Enhanced IGRP, Frame Relay and SMDS networks may or may not be considered to be NBMA. These networks are considered NBMA if the interface has not been configured to use physical multicasting; otherwise they are considered not to be NBMA.

You can configure the hold time on a specified interface for a particular Enhanced IGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is three times the hello interval, or 15 seconds.

On very congested and large networks, 15 seconds may not be sufficient time for all routers to receive hello packets from their neighbors. In this case, you may want to increase the hold time. To increase the hold time, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx hold-time eigrp` *autonomous-system-number seconds*	Sets the hold time.

To change the interval between hello packets, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx hello-interval eigrp` *autonomous-system-number seconds*	Sets the interval between hello packets.

Note Do not adjust the hold time without consulting with Cisco technical support.

Customizing the Exchange of Routing and Service Information

You might want to customize the exchange of routing and service information. The following sections describe these customization tasks:

- Redistributing Routing Information (Optional)
- Disabling Split Horizon (Optional)
- Controlling the Advertising of Routes in Routing Updates (Optional)
- Controlling the Processing of Routing Updates (Optional)
- Controlling SAP Updates (Optional)
- Controlling the Advertising of Services in SAP Updates (Optional)
- Controlling the Processing of SAP Updates (Optional)

Redistributing Routing Information

By default, the Cisco IOS software redistributes IPX RIP routes into Enhanced IGRP, and vice versa.

To disable route redistribution, use the following command in IPX-router configuration mode:

Command	Purpose
`Router(config-ipx-router)# no redistribute {connected | eigrp autonomous-system-number | rip | static}`	Disables redistribution of RIP routes into Enhanced IGRP and Enhanced IGRP routes into RIP.

The Cisco IOS software does not automatically redistribute NLSP routes into Enhanced IGRP routes and vice versa. To configure this type of redistribution, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx router eigrp autonomous-system-number`	From global configuration mode, enables Enhanced IGRP.
Step 2	`Router(config-ipx-router)# redistribute nlsp [tag]`	From IPX-router configuration mode, enables redistribution of NLSP into Enhanced IGRP.
Step 3	`Router(config)# ipx router nlsp [tag]`	Enables NLSP.
Step 4	`Router(config-ipx-router)# redistribute eigrp autonomous-system-number`	From IPX-router configuration mode, enables redistribution of Enhanced IGRP into NLSP.

For an example of how to enable redistribution of Enhanced IGRP and NLSP, see the “Enhanced IGRP and NLSP Route Redistribution Example” section at the end of this chapter.

Disabling Split Horizon

Split horizon controls the sending of Enhanced IGRP update and query packets. If split horizon is enabled on an interface, these packets are not sent for destinations if this interface is the next hop to that destination.

By default, split horizon is enabled on all interfaces.

Split horizon blocks information about routes from being advertised by the Cisco IOS software out any interface from which that information originated. This behavior usually optimizes communication among multiple routers, particularly when links are broken. However, with nonbroadcast networks (such as Frame Relay and SMDS), situations can arise for which this behavior is less than ideal. For these situations, you can disable split horizon.

To disable split horizon, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx split-horizon eigrp` *autonomous-system-number*	Disables split horizon.

Note Split horizon cannot be disabled for RIP or SAP, only for Enhanced IGRP.

Controlling the Advertising of Routes in Routing Updates

To control which devices learn about routes, you can control the advertising of routes in routing updates. To control this advertising, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# distribute-list` *access-list-number* `out` [*interface-name* | *routing-process*]	Controls the advertising of routes in routing updates.

Controlling the Processing of Routing Updates

To control the processing of routes listed in incoming updates, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# distribute-list` *access-list-number* `in` [*interface-name*]	Controls which incoming route updates are processed.

Controlling SAP Updates

If IPX Enhanced IGRP peers are found on an interface, you can configure the Cisco IOS software to send SAP updates either periodically or when a change occurs in the SAP table. When no IPX Enhanced IGRP peer is present on the interface, periodic SAPs are always sent.

On serial lines, by default, if an Enhanced IGRP neighbor is present, the Cisco IOS software sends SAP updates only when the SAP table changes. On Ethernet, Token Ring, and FDDI interfaces, by default, the software sends SAP updates periodically. To reduce the amount of bandwidth required to send SAP

updates, you might want to disable the periodic sending of SAP updates on LAN interfaces. This feature should only be disabled when all nodes out of this interface are Enhanced IGRP peers; otherwise, loss of SAP information on the other nodes will result.

To send SAP updates only when a change occurs in the SAP table, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx sap-incremental eigrp` *autonomous-system-number*	Sends SAP updates only when a change in the SAP table occurs.

To send SAP updates only when a change occurs in the SAP table and to send only the SAP changes, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx sap-incremental eigrp` *autonomous-system-number* **rsup-only**	Sends SAP updates only when a change in the SAP table occurs, and sends only the SAP changes.

When you enable incremental SAP using the **ipx sap-incremental eigrp rsup-only** command, Cisco IOS software disables the exchange of route information via Enhanced IGRP for that interface.

To send periodic SAP updates, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx sap-incremental eigrp` *autonomous-system-number*	Sends SAP updates periodically.

For an example of how to configure SAP updates, see the "Enhanced IGRP SAP Update Examples" section at the end of this chapter.

To disable split horizon for incremental SAP, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx sap-incremental split-horizon`	Disables split horizon for SAP.

Note IPX incremental SAP split horizon is off for WAN interfaces and subinterfaces, and on for LAN interfaces. The global default stays off. The interface setting takes precedence if the interface setting is modified or when both the global and interface settings are unmodified. The global setting is used only when the global setting is modified and the interface setting is unmodified.

Controlling the Advertising of Services in SAP Updates

To control which devices learn about services, you can control the advertising of these services in SAP updates. To control this advertising, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# distribute-sap-list` *access-list-number* **out** [*interface-name* | *routing-process*]	Controls the advertising of services in SAP updates distributed between routing processes.

For a configuration example of controlling the advertisement of SAP updates, see the "Advertisement and Processing of SAP Update Examples" section at the end of this chapter.

Controlling the Processing of SAP Updates

To control the processing of routes listed in incoming updates, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# distribute-sap-list` *access-list-number* **in** [*interface-name*]	Controls which incoming SAP updates are processed.

For a configuration example of controlling the processing of SAP updates, see the "Advertisement and Processing of SAP Update Examples" section at the end of this chapter.

Querying the Backup Server

The backup server table is a table kept for each Enhanced IGRP peer. It lists the IPX servers that have been advertised by that peer. If a server is removed from the main server table at any time and for any reason, the Cisco IOS software examines the backup server table to learn if this just-removed server is known by any of the Enhanced IGRP peers. If it is, the information from that peer is advertised back into the main server table just as if that peer had readvertised the server information to this router. Using this method to allow the router to keep the backup server table consistent with what is advertised by each peer means that only changes to the table must be advertised between Enhanced IGRP routers; full periodic updates need not be sent.

By default, the Cisco IOS software queries its own copy of the backup server table of each Enhanced IGRP neighbor every 60 seconds. To change this interval, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **ipx backup-server-query-interval** *interval*	Specifies the minimum period of time between successive queries of the backup server table of a neighbor.

Configuring IPX and SPX over WANs

You can configure IPX over dial-on-demand routing (DDR), Frame Relay, PPP, SMDS, and X.25 networks. For more information about dial-on-demand routing (DDR) refer to the *Cisco IOS Dial Technologies Configuration Guide.* For more information about Frame Relay, SMDS, and X.25 refer to the *Cisco IOS Wide-Area Networking Configuration Guide*.

When you configure IPX over PPP, address maps are not necessary for this protocol. Also, you can enable IPX header compression over point-to-point links to increase available useful bandwidth of the link and reduce response time for interactive uses of the link.

You can use fast-switching IPX serial interfaces configured for Frame Relay and SMDS, and you can use fast-switching Subnetwork Access Protocol (SNAP)-encapsulated packets over interfaces configured for ATM.

Additionally, you can configure the IPXWAN protocol.

For an example of how to configure IPX over a WAN interface, see the "IPX over a WAN Interface Example" section at the end of this chapter.

Configuring IPX over DDR

IPX sends periodic watchdog keepalive packets from servers to clients after a client session has been idle for approximately 5 minutes. On a DDR link, a call would be made every 5 minutes, regardless of whether there were data packets to send. You can prevent these calls from being made by configuring the Cisco IOS software to respond to the watchdog keepalive packets of a server on behalf of a remote client—sometimes referred to as *spoofing the server*. Spoofing makes a server view a client as always connected, even when it is not, thus reducing the number of available licenses. Users can set the duration of IPX watchdog spoofing and periodically disable it so that Novelle NetWare servers can clean up inactive connections.

When configuring IPX over DDR, you might want to disable the generation of these packets so that a call is not made every 5 minutes. A call made every 5 minutes is not an issue for the other WAN protocols, because they establish dedicated connections rather than establishing connections only as needed.

Use the **ipx watchdog-spoof** command to enable and set the duration of watchdog spoofing. You can specify the number of consecutive hours spoofing is to stay enabled and the number of minutes spoofing is to stay disabled. The server can clean up inactive connections when spoofing is disabled. Be sure that fast switching and autonomous switching are disabled on the serial interface before using this command.

To enable watchdog spoofing, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx watchdog-spoof [enable-time-hours disable-time-minutes]`	Enables and sets the duration of watchdog spoofing.

To keep the serial interface idle when only watchdog packets are being sent, refer to the tasks described in the "Deciding and Preparing to Configure DDR" chapter of the *Cisco IOS Dial Technologies Configuration Guide*. For an example of configuring IPX over DDR, see the "IPX over DDR Example" section at the end of this chapter.

Configuring SPX Spoofing over DDR

Sequenced Packet Exchange (SPX) sends periodic keepalive packets between clients and servers. Similar to IPX watchdog packets, these are keepalive packets that are sent between servers and clients after the data has stopped being transferred. On pay-per-packet or byte networks, these packets can incur large customer telephone connection charges for idle time. You can prevent these calls from being made by configuring the Cisco IOS software to respond to the keepalive packets on behalf of a remote system.

When configuring SPX over DDR, you might want to disable the generation of these packets so that a call has the opportunity to go idle. Disabling the generation of packets may not be an issue for the other WAN protocols, because they establish dedicated connections rather than establishing connections only as needed.

To keep the serial interface idle when only keepalive packets are being sent, refer to the tasks described in the "Deciding and Preparing to Configure DDR" chapter of the *Cisco IOS Dial Technologies Configuration Guide*.

For an example of how to configure SPX spoofing over DDR, see the "IPX over DDR Example" section at the end of this chapter.

Configuring IPX Header Compression

You can configure IPX header compression over point-to-point links. With IPX header compression, a point-to-point link can compress IPX headers only, or the combined IPX and NetWare Core Protocol headers. Currently, point-to-point links must first negotiate IPX header compression via IPXCP or IXPWAN. The Cisco IOS software supports IPX header compression as defined by RFC 1553.

For details on configuring IPX header compression, refer to the "Configuring Medial-Independent PPP and Multilink PPP" chapter in the *Cisco IOS Dial Technologies Configuration Guide*.

Configuring the IPXWAN Protocol

The Cisco IOS software supports the IPXWAN protocol, as defined in RFC 1634. IPXWAN allows a router that is running IPX routing to connect via a serial link to another router, possibly from another manufacturer, that is also routing IPX and using IPXWAN.

IPXWAN is a connection startup protocol. Once a link has been established, IPXWAN incurs little or no overhead.

You can use the IPXWAN protocol over PPP. You can also use it over HDLC; however, the devices at both ends of the serial link must be Cisco routers.

To configure IPXWAN on a serial interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# no ipx network`	Ensures that you have not configured an IPX network number on the interface.
Step 2	`Router(config-if)# encapsulation ppp`	Enables PPP.
Step 3	`Router(config-if)# ipx ipxwan` [*local-node* {*network-number* | **unnumbered**} *local-server-name retry-interval retry-limit*]	Enables IPXWAN.
Step 4	`Router(config-if)# ipx ipxwan error` [**reset** | **resume** | **shutdown**]	Optionally, defines how to handle IPXWAN when a serial link fails.
Step 5	`Router(config-if)# ipx ipxwan static`	Optionally, enables static routing with IPXWAN. Note that the remote site must also use static routing.

Controlling Access to IPX Networks

To control access to IPX networks, first create access lists and then apply them to individual interfaces using filters.

Types of Access Lists

You can create the following IPX access lists to filter various kinds of traffic:

- Standard access list—Restricts traffic based on the source network number. You can further restrict traffic by specifying a destination address and a source and destination address mask. Standard IPX access lists use numbers from 800 to 899 or names to identify them.
- Extended access list—Restricts traffic based on the IPX protocol type. You can further restrict traffic by specifying source and destination addresses and address masks, and source and destination sockets. Extended IPX access lists use numbers from 900 to 999 or names to identify them.
- SAP access list—Restricts traffic based on the IPX SAP type. These lists are used for SAP filters and GNS response filters. Novell SAP access lists use numbers from 1000 to 1099 or names to identify them.
- IPX NetBIOS access list—Restricts IPX NetBIOS traffic based on NetBIOS names, not numbers.

Types of Filters

There are more than 14 different IPX filters that you can define for IPX interfaces. They fall into the following six groups:

- Generic filters—Control which data packets are routed in or out of an interface based on the source and destination addresses and IPX protocol type of the packet.
- Routing table filters—Control which RIP updates are accepted and advertised by the Cisco IOS software, and from which devices the local router accepts RIP updates.
- SAP filters—Control which SAP services the Cisco IOS software accepts and advertises and which GNS response messages it sends out.
- IPX NetBIOS filters—Control incoming and outgoing IPX NetBIOS packets.
- Broadcast filters—Control which broadcast packets are forwarded.

Table 2 summarizes the filters, the access lists they use, and the commands used to define the filters in the first five groups. Use the **show ipx interfaces** command to display the filters defined on an interface.

Table 2 IPX Filters

Filter Type	Access List Used by Filter	Command to Define Filter
Generic filters		
Filters inbound or outbound packets based on the contents of the IPX network header.	Standard or Extended	**ipx access-group** {*access-list-number* | *name*} [**in** | **out**]
Routing table filters		
Controls which networks are added to the routing table.	Standard or Extended	**ipx input-network-filter** {*access-list-number* | *name*}

Table 2 IPX Filters (Continued)

Filter Type	Access List Used by Filter	Command to Define Filter
Controls which networks are advertised in routing updates.	Standard or Extended	**ipx output-network-filter** {*access-list-number* | *name*}
Controls which networks are advertised in the Enhanced IGRP routing updates sent out by the Cisco IOS software.	Standard or Extended	**distribute-list** {*access-list-number* | *name*} **out** [*interface-name* | *routing-process*]
Controls the routers from which updates are accepted.	Standard or Extended	**ipx router-filter** {*access-list-number* | *name*}
SAP filters		
Filters incoming service advertisements.	SAP	**ipx input-sap-filter** {*access-list-number* | *name*}
Filters outgoing service advertisements.	SAP	**ipx output-sap-filter** {*access-list-number* | *name*}
Controls the routers from which SAP updates are accepted.	SAP	**ipx router-sap-filter** {*access-list-number* | *name*}
Filters list of servers in GNS response messages.	SAP	**ipx output-gns-filter** {*access-list-number* | *name*}
IPX NetBIOS filters		
Filters incoming packets by node name.	IPX NetBIOS	**ipx netbios input-access-filter host** *name*
Filters incoming packets by byte pattern.	IPX NetBIOS	**ipx netbios input-access-filter bytes** *name*
Filters outgoing packets by node name.	IPX NetBIOS	**ipx netbios output-access-filter host** *name*
Filters outgoing packets by byte pattern.	IPX NetBIOS	**ipx netbios output-access-filter bytes** *name*
Broadcast filters		
Controls which broadcast packets are forwarded.	Standard or Extended	**ipx helper-list** {*access-list-number* | *name*}

Implementation Considerations

Remember the following information when configuring IPX network access control:

- Access lists entries are scanned in the order you enter them. The first matching entry is used. To improve performance, we recommend that you place the most commonly used entries near the beginning of the access list.
- An implicit *deny everything* entry is defined at the end of an access list unless you include an explicit *permit everything* entry at the end of the list.
- For numbered access lists, all new entries to an existing list are placed at the end of the list. You cannot add an entry to the middle of a list. Consequently, if you have previously included an explicit *permit everything* entry, new entries will never be scanned. The solution is to delete the access list and reenter it with the new entries.

  For named access lists, all new entries to an existing list are placed at the end of the list. You cannot add entries to the middle of a list. However, you can remove specific entries using the **no deny** and **no permit** commands, rather than deleting the entire access list.

- Do not set up conditions that result in packets getting lost. One way you can lose packets is when a device or interface is configured to advertise services on a network that has access lists that deny these packets.

Controlling Access to IPX Networks Task List

To control access to IPX networks, perform the required tasks in the following sections:

- Creating Access Lists (Required)
- Creating Filters (Required)

Creating Access Lists

You can create access lists using numbers or names. You can choose which method you prefer. If you use numbers to identify your access lists, you are limited to 100 access lists per filter type. If you use names to identify your access lists, you can have an unlimited number of access lists per filter type.

The following sections describe how to perform these tasks:

- Creating Access Lists Using Numbers (Optional)
- Creating Access Lists Using Names (Optional)

Creating Access Lists Using Numbers

To create access lists using numbers, use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list` *access-list-number* {**deny** | **permit**} *source-network* [*.source-node* [*source-node-mask*]][*destination-network* [*.destination-node* [*destination-node-mask*]]]	Defines a standard IPX access list using a number. (Generic, routing, and broadcast filters use this type of access list.)
`Router(config)# access-list` *access-list-number* {**deny** | **permit**} *protocol* [*source-network* [*.source-node* [*source-network-mask.source-node-mask*]] *source-socket* [*destination-network* [*.destination-node* [*destination-network-mask.destination-node-mask*] *destination-socket*] [**log**] [**time-range time-range-name**]	Defines an extended IPX access list using a number. (Generic, routing, and broadcast filters use this type of access list.) Use the **log** keyword to get access list logging messages, including violations. Specifies a time range to restrict when the **permit** or **deny** statement is in effect.
`Router(config)# access-list` *access-list-number* {**deny** | **permit**} *network* [*.node*] [*network-mask.node-mask*] [*service-type* [*server-name*]]	Defines a SAP filtering access list using a number. (SAP and GNS response filters use this type of access list.)

Once you have created an access list using numbers, apply it to the appropriate interfaces using filters as described in the "Creating Filters" section later in this chapter. Applying a filter will activate the access list.

Creating Access Lists Using Names

IPX named access lists allow you to identify IPX access lists with an alphanumeric string (a name) rather than a number. Using IPX named access lists allows you to maintain security by using a separate and easily identifiable access list for each user or interface. IPX named access lists also remove the limit of 100 lists per filter type.You can configure an unlimited number of the following types of IPX named access lists:

- Standard
- Extended
- SAP
- NetBIOS

If you identify your access list with a name rather than a number, the mode and command syntax are slightly different.

Implementation Considerations

Consider the following information before configuring IPX named access lists:

- Except for NetBIOS access lists, access lists specified by name are not compatible with releases prior to Cisco IOS Release 11.2(4)F.
- Access list names must be unique across all protocols.
- Except for NetBIOS access lists, numbered access lists are also available.

IPX Named Access List Configuration Task List

To configure IPX named access lists for standard, extended, SAP, NLSP route aggregation (summarization), or NetBIOS access lists, perform one or more of the tasks in the following sections:

- Creating a Named Standard Access List (Optional)
- Creating a Named Extended Access List (Optional)
- Creating a Named SAP Filtering Access List (Optional)
- Creating a NetBIOS Access List (Optional)
- Applying Time Ranges to Access Lists (Optional)

Creating a Named Standard Access List

To create a named standard access list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx access-list standard` *name*	Defines a standard IPX access list using a name. (Generic, routing, and broadcast filters use this type of access list.)
Step 2	`Router(config-access-list)# {deny | permit}` *source-network* `[.`*source-node* `[`*source-node-mask*`]] [`*destination-network* `[.`*destination-node* `[`*destination-node-mask*`]]]`	In access-list configuration mode, specifies one or more conditions allowed or denied. This determines whether the packet is passed or dropped.
Step 3	`Router(config)# exit`	Exits access-list configuration mode.

For an example of creating a named standard access list, see the "Standard Named Access List Example" section at the end of this chapter.

Creating a Named Extended Access List

To create a named extended access list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx access-list extended name`	Defines an extended IPX access list using a name. (Generic, routing, and broadcast filters use this type of access list.)
Step 2	`Router(config-access-list)# {deny | permit} protocol [source-network] [[[.source-node] source-node-mask] | [.source-node source-network-mask.source-node-mask]] [source-socket] [destination-network] [[[.destination-node] destination-node-mask] | [.destination-node destination-network-mask.destination-nodemask]] [destination-socket] [log] [time-range time-range-name]`	In access-list configuration mode, specifies the conditions allowed or denied. Use the **log** keyword to get access list logging messages, including violations. Specifies a time range to restrict when the **permit** or **deny** statement is in effect.
Step 3	`Router(config)# exit`	Exits access-list configuration mode.

Creating a Named SAP Filtering Access List

To create a named access list for filtering SAP requests, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx access-list sap name`	Defines a SAP filtering access list using a name. (SAP, GNS, and Get General Service (GGS) response filters use this type of access list.)
Step 2	`Router(config-access-list)# {deny | permit} network [.node] [network-mask.node-mask] [service-type [server-name]]`	In access-list configuration mode, specifies the conditions allowed or denied.
Step 3	`Router(config)# exit`	Exits access-list configuration mode.

Creating a NetBIOS Access List

To create a NetBIOS access list, use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)# netbios access-list host` *name* {**deny** | **permit**} *string*	Creates an access list for filtering IPX NetBIOS packets by node name. (NetBIOS filters use this type of access list.)
`Router(config)# netbios access-list bytes` *name* {**deny** | **permit**} *offset byte-pattern*	Creates an access list for filtering IPX NetBIOS packets by arbitrary byte pattern. (NetBIOS filters use this type of access list.)

Modifying IPX Named Access Lists

After you initially create an access list, you place any subsequent additions (possibly entered from the terminal) at the end of the list. In other words, you cannot selectively add access list command lines to the middle of a specific access list. However, you can use **no permit** and **no deny** commands to remove entries from a named access list.

Note When creating access lists, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end.

For an example of creating a generic filter, see the "IPX Network Access Examples" section at the end of this chapter.

Applying Named Access Lists to Interfaces

After creating an access list, you must apply it to the appropriate interface using filters as described in the "Creating Filters" section later in this chapter. Applying a filter will activate the access list.

Applying Time Ranges to Access Lists

It is now possible to implement access lists based on the time of day and week using the **time-range** command. To do so, first define the name of the time range and times of the day and week, then reference the time range by name in an access list to apply the restrictions of the time range to the access list.

Currently, IP and IPX named or numbered extended access lists are the only functions that can use time ranges. The time range allows the network administrator to define when the **permit** or **deny** statements in the access list are in effect. Prior to this time range feature, access list statements were always in effect once they were applied. The **time-range** keyword and argument are referenced in the named and numbered extended access list task tables in the previous sections, "Creating Access Lists Using Numbers" and "Creating Access Lists Using Names." The **time-range** command is configured in the "Performing Basic System Management" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*. See the "IPX Network Access Examples" section at the end of this chapter for a configuration example of IPX time ranges.

There are many possible benefits of time ranges, such as the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set time-based security policy, including:
  - Perimeter security using the Cisco IOS Firewall feature set or access lists

  - Data confidentiality with Cisco Encryption Technology or IPS
- Policy-based routing and queueing functions are enhanced.
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- Service providers can dynamically change a committed access rate (CAR) configuration to support the quality of service (QoS) Service Level Agreements (SLAs) that are negotiated for certain times of day.
- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

Creating Filters

Filters allow you to control which traffic is forwarded or blocked at the interfaces of the router. Filters apply specific numbered or named access lists to interfaces.

To create filters, perform the tasks in the following sections:

- Creating Generic Filters (Optional)
- Creating Filters for Updating the Routing Table (Optional)
- Creating SAP Filters (Optional)
- Creating GNS Response Filters (Optional)
- Creating GGS Response Filters (Optional)
- Creating IPX NetBIOS Filters (Optional)
- Creating Broadcast Message Filters (Optional)

Creating Generic Filters

Generic filters determine which data packets to receive from or send to an interface, based on the source and destination addresses, IPX protocol type, and source and destination socket numbers of the packet.

To create generic filters, first create a standard or an extended access list as described in the "Creating Access Lists" section earlier in this chapter and then apply a filter to an interface.

To apply a generic filter to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx access-group {access-list-number | name}[in | out]`	Applies a generic filter to an interface.

You can apply only one input filter and one output filter per interface or subinterface. You cannot configure an output filter on an interface where autonomous switching is already configured. Similarly, you cannot configure autonomous switching on an interface where an output filter is already present. You cannot configure an input filter on an interface if autonomous switching is already configured on *any* interface. Likewise, you cannot configure input filters if autonomous switching is already enabled on *any* interface.

For an example of creating a generic filter, see the "IPX Network Access Examples" section at the end of this chapter.

Creating Filters for Updating the Routing Table

Routing table update filters control the entries that the Cisco IOS software accepts for its routing table, and the networks that it advertises in its routing updates.

To create filters to control updating of the routing table, first create a standard or an extended access list as described in the "Creating Access Lists" section earlier in this chapter and then apply one or more routing filters to an interface.

To apply routing table update filters to an interface, use one or more of the following commands in interface configuration or router configuration mode:

Command	Purpose
`Router(config-if)# ipx input-network-filter {access-list-number | name}`	Controls which networks are added to the routing table when IPX routing updates are received.
`Router(config-if)# ipx output-network-filter {access-list-number | name}`	Controls which networks are advertised in RIP routing updates sent out by the Cisco IOS software.
`Router(config-router)# distribute-list {access-list-number | name} out [interface-name | routing-process]`	Controls which networks are advertised in the Enhanced IGRP routing updates sent out by the Cisco IOS software.
`Router(config-if)# ipx router-filter {access-list-number | name}`	Controls the routers from which routing updates are accepted.

Note The **ipx output-network-filter** command applies to the IPX RIP only. To control the advertising of routes when filtering routing updates in Enhanced IGRP, use the **distribute-list out** command. See the "Controlling the Advertising of Routes in Routing Updates" section earlier in this chapter for more information.

Creating SAP Filters

A common source of traffic on Novell networks is SAP messages, which are generated by NetWare servers and the Cisco IOS software when they broadcast their available services.

To control how SAP messages from network segments or specific servers are routed among IPX networks, first create a SAP filtering access list as described in the "Creating Access Lists" section earlier in this chapter and then apply one or more filters to an interface.

To apply SAP filters to an interface, use one or more of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx input-sap-filter {access-list-number | name}`	Filters incoming service advertisements.

Command	Purpose
`Router(config-if)# ipx output-sap-filter {access-list-number | name}`	Filters outgoing service advertisements.
`Router(config-if)# ipx router-sap-filter {access-list-number | name}`	Filters service advertisements received from a particular router.

You can apply one of each SAP filter to each interface.

For examples of creating and applying SAP filters, see the "SAP Input Filter Example" and "SAP Output Filter Example" sections at the end of this chapter.

Creating GNS Response Filters

To create filters for controlling which servers are included in the GNS responses sent by the Cisco IOS software, first create a SAP filtering access list as described in the "Creating Access Lists" section earlier in this chapter and then apply a GNS filter to an interface.

To apply a GNS filter to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx output-gns-filter {access-list-number | name}`	Filters the list of servers in GNS response messages.

Creating GGS Response Filters

To create filters for controlling which servers are included in the Get General Service (GGS) responses sent by the Cisco IOS software, first create a SAP filtering access list as described in the "Creating Access Lists" section earlier in this chapter and then apply a GGS filter to an interface.

Note Because GGS SAP response filters are applied ahead of output SAP filters, a SAP entry permitted to pass through the GGS SAP response filter can still be filtered by the output SAP filter.

To apply a GGS filter to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx output-ggs-filter`	Filters the list of servers in GGS response messages.

For an example of creating a GGS SAP response filter, see the "IPX Network Access Examples" section at the end of this chapter.

Creating IPX NetBIOS Filters

The Novell IPX NetBIOS allows messages to be exchanged between nodes using alphanumeric names and node addresses. Therefore, the Cisco IOS software lets you filter incoming and outgoing NetBIOS FindName packets by the node name or by an arbitrary byte pattern (such as the node address) in the packet.

Note These filters apply to IPX NetBIOS FindName packets only. They have no effect on Logic Link Control, type 2 (LLC2) NetBIOS packets.

Implementation Considerations

Remember the following when configuring IPX NetBIOS access control:

- Host (node) names are case sensitive.
- Host and byte access lists can have the same names because the two types of lists are independent of each other.
- When nodes are filtered by name, the names in the access lists are compared with the destination name field for IPX NetBIOS "find name" requests.
- Access filters that filter by byte offset can have a significant impact on the packet transmission rate because each packet must be examined. You should use these access lists only when absolutely necessary.
- If a node name is not found in an access list, the default action is to deny access.

Configuring IPX NetBIOS Filters

To create filters for controlling IPX NetBIOS access, first create a NetBIOS access list as described in the "Creating Access Lists" section earlier in this chapter and then apply the access list to an interface.

To apply a NetBIOS access list to an interface, use one or more of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx netbios input-access-filter host name`	Filters incoming packets by node name.
`Router(config-if)# ipx netbios input-access-filter bytes name`	Filters incoming packets by byte pattern.
`Router(config-if)# ipx netbios output-access-filter host name`	Filters outgoing packets by node name.
`Router(config-if)# ipx netbios output-access-filter bytes name`	Filters outgoing packets by byte pattern.

You can apply one of each of these four filters to each interface.

For an example of how to create filters for controlling IPX NetBIOS, see the "IPX NetBIOS Filter Examples" section at the end of this chapter.

Creating Broadcast Message Filters

Routers normally block all broadcast requests and do not forward them to other network segments, therefore preventing the degradation of performance inherent in broadcast traffic over the entire network. You can define which broadcast messages get forwarded to other networks by applying a broadcast message filter to an interface.

To create filters for controlling broadcast messages, first create a standard or an extended access list as described in the "Creating Access Lists" section earlier in this chapter and then apply a broadcast message filter to an interface.

To apply a broadcast message filter to an interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ipx helper-address network.node`	Specifies a helper address for forwarding broadcast messages.
Step 2	`Router(config-if)# ipx helper-list {access-list-number | name}`	Applies a broadcast message filter to an interface.

Note A broadcast message filter has no effect unless you have issued an **ipx helper-address** or an **ipx type-20-propagation** command on the interface to enable and control the forwarding of broadcast messages. These commands are discussed later in this chapter.

For examples of creating and applying broadcast message filters, see the "Helper Facilities to Control Broadcast Examples" section at the end of this chapter.

Tuning IPX Network Performance

To tune IPX network performance, perform the tasks in one or more of the following sections:

- Controlling Novell IPX Compliance (Optional)
- Adjusting RIP and SAP Information (Optional)
- Configuring Load Sharing (Optional)
- Specifying the Use of Broadcast Messages (Optional)
- Disabling IPX Fast Switching (Optional)
- Adjusting the Route Cache (Optional)
- Adjusting Default Routes (Optional)
- Padding Odd-Length Packets (Optional)

Controlling Novell IPX Compliance

The Cisco implementation of the Novell IPX protocol is certified to provide full IPX router functionality, as defined by the Novell *IPX Router Specification, version 1.10* publication published November 17, 1992.

To control compliance to Novell specifications, perform the tasks in the following sections:

- Controlling the Forwarding of Type 20 Packets (Optional)
- Controlling Interpacket Delay (Optional)
- Shutting Down an IPX Network (Optional)
- Achieving Full Novell Compliance (Optional)

Controlling the Forwarding of Type 20 Packets

NetBIOS over IPX uses Type 20 propagation broadcast packets flooded to all networks to get information about the named nodes on the network. NetBIOS uses a broadcast mechanism to get this information, because it does not implement a network layer.

Routers normally block all broadcast requests. By enabling Type 20 packet propagation, IPX interfaces on the router may accept and forward Type 20 packets.

How Type 20 Packet Propagation Works

When an interface configured for Type 20 propagation receives a Type 20 packet, Cisco IOS software processes the packet according to Novell specifications. Cisco IOS software propagates the packet to the next interface. The Type 20 packet can be propagated for up to eight hop counts.

Loop Detection and Other Checks

Before forwarding (flooding) the packets, the router performs loop detection as described by the IPX router specification.

You can configure the Cisco IOS software to apply extra checks to Type 20 propagation packets above and beyond the loop detection described in the IPX specification. These checks are the same ones that are applied to helpered all-nets broadcast packets. They can limit unnecessary duplication of Type 20 broadcast packets. The extra helper checks are as follows:

- Accept Type 20 propagation packets only on the primary network, which is the network that is the primary path back to the source network.
- Forward Type 20 propagation packets only via networks that do not lead back to the source network.

Although this extra checking increases the robustness of Type 20 propagation packet handling by decreasing the amount of unnecessary packet replication, it has the following two side effects:

- If Type 20 packet propagation is not configured on all interfaces, these packets might be blocked when the primary interface changes.
- It might be impossible to configure an arbitrary, manual spanning tree for Type 20 packet propagation.

Relationship Between Type 20 Propagation and Helper Addresses

You use helper addresses to forward non-Type 20 broadcast packets to other network segments. For information on forwarding other broadcast packets, see the "Using Helper Addresses to Forward Broadcast Packets" section later in this chapter.

You can use helper addresses and Type 20 propagation together in your network. Use helper addresses to forward non-Type 20 broadcast packets and use Type 20 propagation to forward Type 20 broadcast packets.

Type 20 Packets Configuration Task List

You can enable the forwarding of Type 20 packets on individual interfaces. Additionally, you can restrict the acceptance and forwarding of Type 20 packets. You can also choose to not comply with Novell specifications and forward Type 20 packets using helper addresses rather than using Type 20 propagation. The following sections describe these tasks:

- Enabling the Forwarding of Type 20 Packets (Optional)
- Restricting the Acceptance of Incoming Type 20 Packets (Optional)
- Restricting the Forwarding of Outgoing Type 20 Packets (Optional)
- Forwarding Type 20 Packets Using Helper Addresses (Optional)

Enabling the Forwarding of Type 20 Packets

By default, Type 20 propagation packets are dropped by the Cisco IOS software. You can configure the software to receive Type 20 propagation broadcast packets and forward (flood) them to other network segments, subject to loop detection.

To enable the receipt and forwarding of Type 20 packets, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx type-20-propagation`	Forwards IPX Type 20 propagation packet broadcasts to other network segments.

When you enable Type 20 propagation, Cisco IOS propagates the broadcast to the next interface up to eight hops.

Restricting the Acceptance of Incoming Type 20 Packets

For incoming Type 20 propagation packets, the Cisco IOS software is configured by default to accept packets on all interfaces enabled to receive Type 20 propagation packets. You can configure the software to accept packets only from the single network that is the primary route back to the source network, which means that similar packets from the same source that are received via other networks will be dropped.

Checking of incoming Type 20 propagation broadcast packets is done only if the interface is configured to receive and forward Type 20 packets.

To impose restrictions on the receipt of incoming Type 20 propagation packets in addition to the checks defined in the IPX specification, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx type-20-input-checks`	Restricts the acceptance of IPX Type 20 propagation packets.

Restricting the Forwarding of Outgoing Type 20 Packets

For outgoing Type 20 propagation packets, the Cisco IOS software is configured by default to send packets on all interfaces enabled to send Type 20 propagation packets, subject to loop detection. You can configure the software to send these packets only to networks that are not routes back to the source network. (The software uses the current routing table to determine routes.)

Checking of outgoing Type 20 propagation broadcast packets is done only if the interface is configured to receive and forward Type 20 packets.

To impose restrictions on the transmission of Type 20 propagation packets, and to forward these packets to all networks using only the checks defined in the IPX specification, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx type-20-output-checks`	Restricts the forwarding of IPX Type 20 propagation packets.

Forwarding Type 20 Packets Using Helper Addresses

You can also forward Type 20 packets to specific network segments using helper addresses rather than using the Type 20 packet propagation.

You may want to forward Type 20 packets using helper addresses when some routers in your network are running versions of Cisco IOS that do not support Type 20 propagation. When some routers in your network support Type 20 propagation and others do not, you can avoid flooding packets everywhere in the network by using helper addresses to direct packets to certain segments only.

Cisco IOS Release 9.1 and earlier versions do not support Type 20 propagation.

Note Forwarding Type 20 packets using helper addresses does not comply with the Novell IPX router specification.

To forward Type 20 packets addresses using helper addresses, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx type-20-helpered`	Forwards IPX Type 20 packets to specific networks segments. This step turns off Type 20 propagation.
Step 2	`Router(config-if)# ipx helper-address` *network.node*	From interface configuration mode, specifies a helper address for forwarding broadcast messages, including IPX Type 20 packets.

The Cisco IOS software forwards Type 20 packets to only those nodes specified by the **ipx helper-address** command.

Note Using the **ipx type-20-helpered** command disables the receipt and forwarding of Type 20 propagation packets as directed by the **ipx type-20-propagation** command.

Controlling Interpacket Delay

To control interpacket delay, you can use a combination of global configuration and interface configuration commands.

Use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx default-output-rip-delay` *delay*	Sets the interpacket delay of multiple-packet routing updates sent on all interfaces.
`Router(config)# ipx default-triggered-rip-delay` *delay*	Sets the interpacket delay of multiple-packet triggered routing updates sent on all interfaces.
`Router(config)# ipx default-output-sap-delay` *delay*	Sets the interpacket delay of multiple-packet SAP updates sent on all interfaces.
`Router(config)# ipx default-triggered-sap-delay` *delay*	Sets the interpacket delay of multiple-packet triggered SAP updates sent on all interfaces.

Use one or more of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx output-rip-delay delay`	Sets the interpacket delay of multiple-packet routing updates sent on a single interface.
`Router(config-if)# ipx triggered-rip-delay delay`	Sets the interpacket delay of multiple-packet triggered routing updates sent on a single interface.
`Router(config-if)# ipx output-sap-delay delay`	Sets the interpacket delay of multiple-packet SAP updates sent on a single interface.
`Router(config-if)# ipx triggered-sap-delay delay`	Sets the interpacket delay of multiple-packet triggered SAP updates sent on a single interface.

Note We recommend that you use the **ipx output-rip-delay** and **ipx output-sap-delay** commands on slower speed WAN interfaces. The default delay for Cisco IOS Release 11.1 and later versions is 55 milliseconds.

Shutting Down an IPX Network

To shut down an IPX network using a Novell-compliant method, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx down network`	Administratively shuts down an IPX network on an interface. This removes the network from the interface.

Convergence is faster when you shut down an IPX network using the **ipx down** command than when using the **shutdown** command.

Achieving Full Novell Compliance

To achieve full compliance on each interface configured for IPX, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ipx output-rip-delay 55`	Sets the interpacket delay of multiple-packet routing updates to 55 milliseconds.
Step 2	`Router(config-if)# ipx output-sap-delay 55`	Sets the interpacket delay of multiple-packet SAP updates to 55 milliseconds.
Step 3	`Router(config-if)# ipx type-20-propagation`	Optionally enables Type 20 packet propagation if you want to forward Type 20 broadcast traffic across the router.

You can also globally set interpacket delays for multiple-packet RIP and SAP updates to achieve full compliance, eliminating the need to set delays on each interface. To set these interpacket delays, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx default-output-rip-delay 55`	Sets the interpacket delay of multiple-packet routing updates sent on all interfaces to 55 milliseconds.
Step 2	`Router(config)# ipx default-output-sap-delay 55`	Sets the interpacket delay of multiple-packet SAP updates sent on all interfaces to 55 milliseconds.

Note The default delay for Cisco IOS Release 11.1 and later versions is 55 milliseconds.

Adjusting RIP and SAP Information

To adjust RIP and SAP information, perform one or more of the optional tasks in the following sections:

- Configuring Static Routes (Optional)
- Adjusting the RIP Delay Field (Optional)
- Controlling Responses to RIP Requests (Optional)
- Adjusting RIP Update Timers (Optional)
- Configuring RIP Update Packet Size (Optional)
- Configuring Static SAP Table Entries (Optional)
- Configuring the Queue Length for SAP Requests (Optional)
- Adjusting SAP Update Timers (Optional)
- Configuring SAP Update Packet Size (Optional)
- Enabling SAP-after-RIP (Optional)
- Disabling Sending of General RIP or SAP Queries (Optional)
- Controlling Responses to GNS Requests (Optional)

Configuring Static Routes

IPX uses RIP, Enhanced IGRP, or NLSP to determine the best path when several paths to a destination exist. The routing protocol then dynamically updates the routing table. However, you might want to add static routes to the routing table to explicitly specify paths to certain destinations. Static routes always override any dynamically learned paths.

Be careful when assigning static routes. When links associated with static routes are lost, traffic may stop being forwarded or traffic may be forwarded to a nonexistent destination, even though an alternative path might be available.

To add a static route to the routing table, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx route {network [network-mask] | default} {network.node | interface}[ticks] [hops]`	Adds a static route to the routing table.

You can configure static routes that can be overridden by dynamically learned routes. These routes are referred to as floating static routes. You can use a floating static route to create a path of last resort that is used only when no dynamic routing information is available.

Note By default, floating static routes are not redistributed into other dynamic protocols.

To add a floating static route to the routing table, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx route {network [network-mask] | default} {network.node | interface}[ticks] [hops] [floating-static]`	Adds a floating static route to the routing table.

Adjusting the RIP Delay Field

By default, all LAN interfaces have a RIP delay of 1 and all WAN interfaces have a RIP delay of 6. Leaving the delay at its default value is sufficient for most interfaces. However, you can adjust the RIP delay field by setting the tick count. To set the tick count, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx delay ticks`	Sets the tick count, which is used in the IPX RIP delay field.

Controlling Responses to RIP Requests

To control responses to RIP requests, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx rip-response-delay ms`	Sets the delay when responding to RIP requests.

Adjusting RIP Update Timers

You can set the interval between IPX RIP updates on a per-interface basis. You can also specify the delay between the packets of a multiple-packet RIP update on a per-interface or global basis. Additionally, you can specify the delay between packets of a multiple-packet triggered RIP update on a per-interface or global basis.

You can set RIP update timers only in a configuration in which all routers are Cisco routers, or in which the IPX routers allow configurable timers. The timers should be the same for all devices connected to the same cable segment. The update value you choose affects internal IPX timers as follows:

- IPX routes are marked invalid if no routing updates are heard within three times the value of the update interval (3 * *interval*) and are advertised with a metric of infinity.
- IPX routes are removed from the routing table if no routing updates are heard within four times the value of the update interval (4 * *interval*).

- If you define a timer for more than one interface in a router, the granularity of the timer is determined by the lowest value defined for one of the interfaces in the router. The router "wakes up" at this granularity interval and sends out updates as appropriate. For more information about granularity, refer to the "Novell IPX Commands" chapter in the *Cisco IOS AppleTalk and Novell IPX Command Reference.*

You might want to set a delay between the packets in a multiple-packet update if there are some slower PCs on the network or on slower-speed interfaces.

To adjust RIP update timers on a per-interface basis, use one or all of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx update interval {rip | sap} {value | changes-only}`	Adjusts the RIP update timer.
`Router(config-if)# ipx output-rip-delay delay`	Adjusts the delay between multiple-packet routing updates sent on a single interface.
`Router(config-if)# ipx triggered-rip-delay delay`	Adjusts the delay between multiple-packet triggered routing updates sent on a single interface.

To adjust RIP update timers on a global basis, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx default-output-rip-delay delay`	Adjusts the delay between multiple-packet routing updates sent on all interfaces.
`Router(config)# ipx default-triggered-rip-delay delay`	Adjusts the delay between multiple-packet triggered routing updates sent on all interfaces.

By default, the RIP entry for a network or server ages out at an interval equal to three times the RIP timer. To configure the multiplier that controls the interval, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx rip-multiplier multiplier`	Configures the interval at which a network RIP entry ages out.

Configuring RIP Update Packet Size

By default, the maximum size of RIP updates sent out an interface is 432 bytes. This size allows for 50 routes at 8 bytes each, plus a 32-byte IPX RIP header. To modify the maximum packet size, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx rip-max-packetsize bytes`	Configures the maximum packet size of RIP updates sent out an interface.

Configuring Static SAP Table Entries

Servers use SAP to advertise their services via broadcast packets. The Cisco IOS software stores this information in the SAP table, also known as the Server Information Table. This table is updated dynamically. You might want to explicitly add an entry to the Server Information Table so that clients always use the services of a particular server. Static SAP assignments always override any identical entries in the SAP table that are learned dynamically, regardless of hop count. If a dynamic route that is associated with a static SAP entry is lost or deleted, the software will not announce the static SAP entry until it relearns the route.

To add a static entry to the SAP table, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx sap service-type name network.node socket hop-count`	Specifies a static SAP table entry.

Configuring the Queue Length for SAP Requests

The Cisco IOS software maintains a list of SAP requests to process, including all pending Get Nearest Server (GNS) queries from clients attempting to reach servers. When the network is restarted following a power failure or other unexpected event, the router can be inundated with hundreds of requests for servers. Typically, many of these are repeated requests from the same clients. You can configure the maximum length allowed for the pending SAP requests queue. SAP requests received when the queue is full are dropped, and the client must resend them.

To set the queue length for SAP requests, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx sap-queue-maximum number`	Configures the maximum SAP queue length.

Adjusting SAP Update Timers

You can adjust the interval at which SAP updates are sent. You can also set the delay between packets of a multiple-packet SAP update on a per-interface or global basis. Additionally, you can specify the delay between packets of a multiple-packet triggered SAP update on a per-interface or global basis.

Changing the interval at which SAP updates are sent is most useful on limited-bandwidth, point-to-point links such as slower-speed interfaces. You should ensure that all IPX servers and routers on a given network have the same SAP interval. Otherwise, they might decide that a server is down when it is really up.

It is not possible to change the interval at which SAP updates are sent on most PC-based servers. Therefore, you should never change the interval for an Ethernet or Token Ring network that has servers on it.

You can set the router to send an update only when changes have occurred. Using the **changes-only** keyword specifies the sending of a SAP update only when the link comes up, when the link is downed administratively, or when the databases change. The **changes-only** keyword causes the router to do the following:

- Send a single, full broadcast update when the link comes up
- Send appropriate triggered updates when the link is shut down
- Send appropriate triggered updates when specific service information changes

To modify the SAP update timers on a per-interface basis, use one or all of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx update interval {rip | sap} {value | changes-only}`	Adjusts the interval at which SAP updates are sent.
`Router(config-if)# ipx output-sap-delay delay`	Adjusts the interpacket delay of multiple-packet SAP updates sent on a single interface.
`Router(config-if)# ipx triggered-sap-delay delay`	Adjusts the interpacket delay of multiple-packet triggered SAP updates sent on a single interface.

To adjust SAP update timers on a global basis (eliminating the need to configure delays on a per-interface basis), use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx default-output-sap-delay delay`	Adjusts the interpacket delay of multiple-packet SAP updates sent on all interfaces.
`Router(config)# ipx default-triggered-sap-delay delay`	Adjusts the interpacket delay of multiple-packet triggered SAP updates sent on all interfaces.

By default, the SAP entry of a network or server ages out at an interval equal to three times the SAP update interval. To configure the multiplier that controls the interval, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx sap-multiplier multiplier`	Configures the interval at which the SAP entry of a network or server ages out.

Configuring SAP Update Packet Size

By default, the maximum size of SAP updates sent out on an interface is 480 bytes. This size allows for seven servers (64 bytes each), plus a 32-byte IPX SAP header. To modify the maximum packet size, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx sap-max-packetsize bytes`	Configures the maximum packet size of SAP updates sent out an interface.

Enabling SAP-after-RIP

The IPX SAP-after-RIP feature links SAP updates to RIP updates so that SAP broadcast and unicast updates automatically occur immediately after the completion of the corresponding RIP update. This feature ensures that a remote router does not reject service information because it lacks a valid route to the service. As a result of this feature, periodic SAP updates are sent at the same interval as RIP updates.

The default behavior of the router is to send RIP and SAP periodic updates with each using its own update interval, depending on the configuration. In addition, RIP and SAP periodic updates are jittered slightly, such that they tend to diverge from each other over time. This feature synchronizes SAP and RIP updates.

Sending all SAP and RIP information in a single update reduces bandwidth demands and eliminates erroneous rejections of SAP broadcasts.

Linking SAP and RIP updates populates the service table of the remote router more quickly, because services will not be rejected due to the lack of a route to the service. Populating the service table more quickly can be especially useful on WAN circuits where the update intervals have been greatly increased to reduce the overall level of periodic update traffic on the link.

To configure the router to send a SAP update following a RIP broadcast, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx update sap-after-rip`	Configures the router to send a SAP broadcast immediately following a RIP broadcast.

Disabling Sending of General RIP or SAP Queries

You can disable the sending of general RIP or SAP queries on a link when it first comes up to reduce traffic and save bandwidth.

RIP and SAP general queries are normally sent by remote routers when a circuit first comes up. On WAN circuits, two full updates of each kind are often sent across the link. The first update is a full broadcast update, triggered locally by the link-up event. The second update is a specific (unicast) reply triggered by the general query received from the remote router. If you disable the sending of general queries when the link first comes up, it is possible to reduce traffic to a single update, and save bandwidth.

To disable the sending of a general RIP or SAP query when an interface comes up, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx linkup-request {rip | sap}`	Disables the sending of a general RIP or SAP Query when an interface comes up.

To reenable the sending of a general RIP or SAP query, use the positive form of the command.

Controlling Responses to GNS Requests

You can set the method in which the router responds to SAP GNS requests, you can set the delay time in responding to these requests, or you can disable the sending of responses to these requests altogether.

By default, the router responds to GNS requests if appropriate. For example, if a local server with a better metric exists, then the router does not respond to the GNS request on that segment.

The default method of responding to GNS requests is to respond with the server whose availability was learned most recently.

To control responses to GNS requests, use one or both of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx gns-round-robin`	Responds to GNS requests using a round-robin selection method.
`Router(config)# ipx gns-response-delay [milliseconds]`	Sets the delay when responding to GNS requests.

Note The **ipx gns-response-delay** command is also supported as an interface configuration command. To override the global delay value for a specific interface, use the **ipx gns-response-delay** command in interface configuration mode.

To disable GNS queries on a per-interface basis, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx gns-reply-disable`	Disables the sending of replies to Get Nearest Server (GNS) queries.

Configuring Load Sharing

To configure IPX to perform round-robin or per-host load sharing, perform the tasks described in the following sections:

- Enabling Round-Robin Load Sharing (Optional)
- Enabling per-Host Load Sharing (Optional)

Enabling Round-Robin Load Sharing

You can set the maximum number of equal-cost, parallel paths to a destination. (Note that when paths have differing costs, the Cisco IOS software chooses lower-cost routes in preference to higher-cost routes.) The software then distributes output on a packet-by-packet basis in round-robin fashion. That is, the first packet is sent along the first path, the second packet along the second path, and so on. When the final path is reached, the next packet is sent to the first path, the next to the second path, and so on. This round-robin scheme is used regardless of whether fast switching is enabled.

Limiting the number of equal-cost paths can save memory on routers with limited memory or very large configurations. Additionally, in networks with a large number of multiple paths and systems with limited ability to cache out-of-sequence packets, performance might suffer when traffic is split between many paths.

To set the maximum number of paths, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx maximum-paths paths`	Sets the maximum number of equal-cost paths to a destination.

Enabling per-Host Load Sharing

Round-robin load sharing is the default behavior when you configure **ipx maximum-paths** to a value greater than 1. Round-robin load sharing works by sending data packets over successive equal cost paths without regard to individual end hosts or user sessions. Path utilization increases transmission speed, but, because packets destined for a given end host may take different paths, they might arrive out of order.

You can address the possibility of packets arriving out of order by enabling per-host load sharing. With per-host load sharing, the router still uses multiple, equal-cost paths to achieve load sharing; however, packets for a given end host are guaranteed to take the same path, even if multiple, equal-cost paths are available. Traffic for different end hosts tend to take different paths, but true load balancing is not guaranteed. The exact degree of load balancing achieved depends on the exact nature of the workload.

To enable per-host load sharing, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx maximum-paths` *paths*	Sets the maximum number of equal cost paths to a destination to a value greater than 1.
Step 2	`Router(config)# ipx per-host-load-share`	Enables per-host load sharing.

Specifying the Use of Broadcast Messages

To specify the use of broadcast messages, perform the tasks described in the following sections:

- Using Helper Addresses to Forward Broadcast Packets (Optional)
- Enabling Fast Switching of IPX Directed Broadcast Packets (Optional)

Using Helper Addresses to Forward Broadcast Packets

Routers normally block all broadcast requests and do not forward them to other network segments, therefore preventing the degradation of performance over the entire network. However, you can enable the router to forward broadcast packets to helper addresses on other network segments.

How Helper Addresses Work

Helper addresses specify the network and node on another segment that can receive unrecognized broadcast packets. Unrecognized broadcast packets are non-RIP and non-SAP packets that are not addressed to the local network.

When the interface configured with helper addresses receives an unrecognized broadcast packet, Cisco IOS software changes the broadcast packet to a unicast and sends the packet to the specified network and node on the other network segment. Unrecognized broadcast packets are not flooded everywhere in your network.

With helper addresses, there is no limit on the number of hops that the broadcast packet can make.

Fast Switching Support

Cisco IOS supports fast switching of helpered broadcast packets.

When to Use Helper Addresses

You use helper addresses when you want to forward broadcast packets (except Type 20 packets) to other network segments.

Forwarding broadcast packets to helper addresses is sometimes useful when a network segment does not have an end-host capable of servicing a particular type of broadcast request. You can specify the address of a server, network, or networks that can process the broadcast packet.

Relationship Between Helper Addresses and Type 20 Propagation

You use Type 20 packet propagation to forward Type 20 packets to other network segments. For information on forwarding Type 20 packets, see the "Controlling the Forwarding of Type 20 Packets" section earlier in this chapter.

You can use helper addresses and Type 20 propagation together in your network. Use helper addresses to forward non-Type 20 broadcast packets and use Type 20 propagation to forward Type 20 broadcast packets.

Implementation Considerations

Using helper addresses is not Novell-compliant. However, it does allow routers to forward broadcast packets to network segments that can process them without flooding the network. It also allows routers running versions of Cisco IOS that do not support Type 20 propagation to forward Type 20 packets.

The Cisco IOS software supports all-networks flooded broadcasts (sometimes referred to as *all-nets flooding*). These are broadcast messages that are forwarded to all networks. Use all-nets flooding carefully and only when necessary, because the receiving networks may be overwhelmed to the point that no other traffic can traverse them.

Use the **ipx helper-list** command, described earlier in this chapter, to define access lists that control which broadcast packets get forwarded.

Using Helper Addresses

To specify a helper address for forwarding broadcast packets, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx helper-address` *network.node*	Specifies a helper address for forwarding broadcast messages.

You can specify multiple helper addresses on an interface.

For an example of using helper addresses to forward broadcast messages, see the "Helper Facilities to Control Broadcast Examples" section at the end of this chapter.

Enabling Fast Switching of IPX Directed Broadcast Packets

By default, Cisco IOS software switches packets that have been helpered to the broadcast address. To enable fast switching of these IPX-directed broadcast packets, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx broadcast-fastswitching`	Enables fast switching of IPX-directed broadcast packets.

Disabling IPX Fast Switching

By default, fast switching is enabled on all interfaces that support fast switching.

Fast switching allows higher throughput by switching a packet using a cache created by previous packets. Fast switching is enabled by default on all interfaces that support fast switching.

Packet transfer performance is generally better when fast switching is enabled. However, you might want to disable fast switching in order to save memory space on interface cards and to help avoid congestion when high-bandwidth interfaces are writing large amounts of information to low-bandwidth interfaces.

Caution Turning off fast switching increases system overhead.

To disable IPX fast switching, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx route-cache`	Disables IPX fast switching.

Adjusting the Route Cache

Adjusting the route cache allows you to control the size of the route cache, reduce memory consumption, and improve router performance. You accomplish these tasks by controlling the route cache size and invalidation. The following sections describe these optional tasks:

- Controlling Route Cache Size (Optional)
- Controlling Route Cache Invalidation (Optional)

Controlling Route Cache Size

You can limit the number of entries stored in the IPX route cache to free up router memory and aid router processing.

Storing too many entries in the route cache can use a significant amount of router memory, causing router processing to slow. This situation is most common on large networks that run network management applications for NetWare.

For example, if a network management station is responsible for managing all clients and servers in a very large (greater than 50,000 nodes) Novell network, the routers on the local segment can become inundated with route cache entries. You can set a maximum number of route cache entries on these routers to free up router memory and aid router processing.

To set a maximum limit on the number of entries in the IPX route cache, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx route-cache max-size` *size*	Sets a maximum limit on the number of entries in the IPX route cache.

If the route cache has more entries than the specified limit, the extra entries are not deleted. However, they may be removed if route cache invalidation is in use. See the "Controlling Route Cache Invalidation" section later in this chapter for more information on invalidating route cache entries.

Controlling Route Cache Invalidation

You can configure the router to invalidate fast-switch cache entries that are inactive. If these entries remain invalidated for 1 minute, the router purges the entries from the route cache.

Purging invalidated entries reduces the size of the route cache, reduces memory consumption, and improves router performance. Also, purging entries helps ensure accurate route cache information.

You specify the period of time that valid fast-switch cache entries must be inactive before the router invalidates them. You can also specify the number of cache entries that the router can invalidate per minute.

To configure the router to invalidate fast-switch cache entries that are inactive, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx route-cache inactivity-timeout` *period* [*rate*]	Invalidates fast-switch cache entries that are inactive.

When you use the **ipx route-cache inactivity-timeout** command with the **ipx route-cache max-size** command, you can ensure a small route cache with fresh entries.

Adjusting Default Routes

You can adjust the use of default routes in your IPX network. You can turn off the use of network number -2 as the default route. You can also specify that the router advertise only default RIP routes out an interface. The following sections describe these optional tasks:

- Disabling Network Number -2 as the Default Route (Optional)
- Advertising Only Default RIP Routes (Optional)

Disabling Network Number -2 as the Default Route

The default route is used when a route to any destination network is unknown. All packets for which a route to the destination address is unknown are forwarded to the default route. By default, IPX treats network number -2 (0xFFFFFFFE) as the default route.

For an introduction to default routes, see the "IPX Default Routes" section earlier in this chapter. For more background information on how to handle IPX default routes, refer to the Novell *NetWare Link Services Protocol (NLSP) Specification, Revision 1.1* publication.

By default, Cisco IOS software treats network -2 as the default route. You can disable this default behavior and use network -2 as a regular network number in your network.

To disable the use of network number -2 as the default route, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no ipx default-route`	Disables default route handling.

Advertising Only Default RIP Routes

Unless configured otherwise, all known RIP routes are advertised out each interface. However, you can choose to advertise only the default RIP route if it is known, therefore greatly reducing the CPU overhead when routing tables are large.

To advertise only the default route via an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx advertise-default-route-only network`	Advertises only the default RIP route.

Padding Odd-Length Packets

Some IPX end hosts accept only even-length Ethernet packets. If the length of a packet is odd, the packet must be padded with an extra byte so that end host can receive it. By default, Cisco IOS pads odd-length Ethernet packets.

However, there are cases in certain topologies where nonpadded Ethernet packets are forwarded onto a remote Ethernet network. Under specific conditions, you can enable padding on intermediate media as a temporary workaround for this problem. Note that you should perform this task only under the guidance of a customer engineer or other service representative.

To enable the padding of odd-length packets, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# no ipx route-cache`	Disables fast switching.
Step 2	`Router(config-if)# ipx pad-process-switched-packets`	Enables the padding of odd-length packets.

Shutting Down an IPX Network

You can administratively shut down an IPX network in two ways. In the first way, the network still exists in the configuration, but is not active. When shutting down, the network sends out update packets informing its neighbors that it is shutting down, therefore allowing the neighboring systems to update their routing, SAP, and other tables without needing to wait for routes and services learned via this network to time out.

To shut down an IPX network such that the network still exists in the configuration, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx down network`	Shuts down an IPX network, but allows the network to still exist in the configuration.

To shut down an IPX network and remove it from the configuration, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# no ipx network`	Shuts down an IPX network and removes it from the configuration.
`Router(config-if)# no ipx network` *network* (where *network* is 1, the primary interface)	When multiple networks are configured on an interface, shuts down all networks and removes them from the interface.
`Router(config-if)# no ipx network` *network* (where *network* is the number of the secondary interface [not 1])	When multiple networks are configured on an interface, shuts down one of the secondary networks and removes it from the interface.

When multiple networks are configured on an interface and you want to shut down one of the secondary networks and remove it from the interface, use the second command in the previous table specifying the network number of one of the secondary networks.

For an example of shutting down an IPX network, see the "IPX Routing Examples" section at the end of this chapter.

Configuring IPX Accounting

IPX accounting enables you to collect information about IPX packets and the number of bytes that are switched through the Cisco IOS software. You collect information based on the source and destination IPX address. IPX accounting tracks only IPX traffic that is routed out an interface on which IPX accounting is configured; it does not track traffic generated by or terminated at the router itself.

The Cisco IOS software maintains two accounting databases: an active database and a checkpoint database. The active database contains accounting data tracked until the database is cleared. When the active database is cleared, its contents are copied to the checkpoint database. Using these two databases together enables you to monitor both current traffic and traffic that has previously traversed the router.

Switching Support

Process and fast switching support IPX accounting statistics. Autonomous and silicon switching engine (SSE) switching do not support IPX accounting statistics.

Note CiscoBus (Cbus) and SSE are not supported on the MIP interface.

Access List Support

IPX access lists support IPX accounting statistics.

IPX Accounting Task List

To configure IPX accounting, perform the tasks in the following sections. The first task is required; the remaining task is optional.

- Enabling IPX Accounting (Required)
- Customizing IPX Accounting (Optional)

Enabling IPX Accounting

To enable IPX accounting, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx accounting`	Enables IPX accounting.

Customizing IPX Accounting

To customize IPX accounting, use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)# ipx accounting-threshold` *threshold*	Sets the maximum number of accounting entries.
`Router(config)# ipx accounting-transits` *count*	Sets the maximum number of transit entries.
`Router(config)# ipx accounting-list` *number mask*	Defines the filter networks for which IPX accounting information is kept. Use one command for each network.

Transit entries are entries in the database that do not match any of the networks specified by the **ipx accounting-list** commands.

If you enable IPX accounting on an interface but do not specify an accounting list, IPX accounting tracks all traffic through the interface (all transit entries) up to the accounting threshold limit.

For an example of how to configure IPX accounting, see the “IPX Accounting Example” section at the end of this chapter.

Configuring IPX Between LANs

Cisco IOS software supports routing IPX between Ethernet-emulated LANs and Token Ring-emulated LANs. For more information on emulated LANs and routing IPX between them, refer to the “Configuring LAN Emulation” chapter of the *Cisco IOS Switching Services Configuration Guide*.

Configuring IPX Between VLANs

Cisco IOS software supports routing IPX between VLANs. Users with Novell NetWare environments can configure any one of the four IPX Ethernet encapsulations to be routed using the Inter-Switch Link (ISL) encapsulation across VLAN boundaries. For more information on VLANs and routing IPX between them over ISL, refer to the “Configuring Routing Between VLANs with ISL Encapsulation” chapter of the *Cisco IOS Switching Services Configuration Guide*.

Configuring IPX Multilayer Switching

Cisco IOS software supports IPX Multilayer Switching (MLS). For more information on IPX MLS, refer to the "Multilayer Switching" chapter of the *Cisco IOS Switching Services Configuration Guide*.

Monitoring and Maintaining the IPX Network

To monitor and maintain your IPX network, perform the optional tasks described in the following sections:

- General Monitoring and Maintaining Tasks (Optional)
- Monitoring and Maintaining IPX Enhanced IGRP (Optional)
- Monitoring and Maintaining IPX Accounting (Optional)

General Monitoring and Maintaining Tasks

You can perform one or more of these general monitoring and maintaining tasks as described in the following sections:

- Monitoring and Maintaining Caches, Tables, Interfaces, and Statistics (Optional)
- Specifying the Type and Use of Ping Packets (Optional)
- Troubleshooting Network Connectivity (Optional)

Monitoring and Maintaining Caches, Tables, Interfaces, and Statistics

To monitor and maintain caches, tables, interfaces, or statistics in a Novell IPX network, use one or more of the following commands in EXEC mode:

Command	Purpose
Router> **clear ipx cache**	Deletes all entries in the IPX fast-switching cache.
Router> **clear ipx route** [*network* | *****]	Deletes entries in the IPX routing table.
Router> **clear ipx traffic**	Clears IPX traffic counters.
Router> **show ipx cache**	Lists the entries in the IPX fast-switching cache.
Router> **show ipx interface** [*type number*]	Displays the status of the IPX interfaces configured in the router and the parameters configured on each interface.
Router> **show ipx route** [*network*] [**default**] [**detailed**]	Lists the entries in the IPX routing table.
Router> **show ipx servers** [**unsorted** | **sorted** [**name** | **net** | **type**]] [**regexp** *name*]	Lists the servers discovered through SAP advertisements.
Router> **show ipx traffic** [**since** {**bootup** | **show**}]	Displays information about the number and type of IPX packets sent and received.
Router> **show sse summary**	Displays a summary of SSE statistics.

Specifying the Type and Use of Ping Packets

The Cisco IOS software can send Cisco pings and standard Novell pings as defined in the NLSP specification or diagnostic request packets. By default, the software generates Cisco pings. To choose the ping type, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx ping-default {cisco | novell | diagnostic}`	Selects the ping type.

The IPX diagnostic ping feature addresses diagnostic related issues by accepting and processing unicast or broadcast diagnostic packets. It makes enhancements to the current IPX **ping** command to ping other stations using the diagnostic packets and display the configuration information in the response packet.

Note When a ping is sent from one station to another, the response is expected to come back immediately; when the **ipx ping-default** command is set to diagnostics, the response could consist of more than one packet and each node is expected to respond within 0.5 seconds of receipt of the request. Due to the absence of an end-of-message flag, there is a delay and the requester must wait for all responses to arrive. Therefore, in verbose mode there may be a brief delay of 0.5 seconds before the response data is displayed.

The **ipx ping** command using the **diagnostic** keyword can be used to conduct a reachability test and should not be used to measure accurate round-trip delay.

To initiate a ping, use one of the following commands in EXEC mode:

Command	Purpose
`Router# ping ipx network.node`	Diagnoses basic IPX network connectivity (user-level command).
`Router# ping [ipx] [network.node]`	Diagnoses basic IPX network connectivity (privileged command).

Troubleshooting Network Connectivity

To trace the IPX destination and measure roundtrip delays, use the following command in either user or privileged EXEC mode:

Command	Purpose
`Router> trace [protocol] [destination]`	Traces packet routes through the network (user or privileged).

Note In user EXEC mode, you are not allowed to change the trace route timeout interval, probe count, minimum and maximum time to live, and verbose mode. To do so, use the **trace** command in privileged EXEC mode.

Monitoring and Maintaining IPX Enhanced IGRP

To monitor and maintain Enhanced IGRP on an IPX network, use one or more of the following commands in EXEC mode:

Command	Purpose
Router> **show ipx eigrp neighbors** [**servers**] [*autonomous-system-number* | *type number* [**regexp** *name*]]	Lists the neighbors discovered by IPX Enhanced IGRP.
Router> **show ipx eigrp interfaces** [*type number*] [*as-number*]	Displays information about interfaces configured for Enhanced IGRP.
Router> **show ipx eigrp topology** [*network*]	Displays the contents of the IPX Enhanced IGRP topology table.
Router> **show ipx route** [*network*]	Displays the contents of the IPX routing table, including Enhanced IGRP entries.
Router> **show ipx traffic**	Displays information about IPX traffic, including Enhanced IGRP traffic.

Logging Enhanced IGRP Neighbor Adjacency Changes

You can enable the logging of neighbor adjacency changes to monitor the stability of the routing system and to help you detect problems. By default, adjacency changes are not logged.

To enable logging of Enhanced IGRP neighbor adjacency changes, use the following command in IPX-router configuration mode:

Command	Purpose
Router(config-ipx-router)# **log-neighbor-changes**	Enables logging of Enhanced IGRP neighbor adjacency changes.

Monitoring and Maintaining IPX Accounting

To monitor and maintain IPX accounting in your IPX network, use the following commands in EXEC mode:

Command	Purpose
Router> **clear ipx accounting** [**checkpoint**]	Deletes all entries in the IPX accounting or accounting checkpoint database.
Router> **show ipx accounting** [**checkpoint**]	Lists the entries in the IPX accounting or accounting checkpoint database.

Novell IPX Configuration Examples

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This following sections provide IPX configuration examples:

- IPX Routing Examples
- Enhanced IGRP Examples
- IPX over WAN Examples
- IPX Network Access Examples
- Helper Facilities to Control Broadcast Examples
- IPX Accounting Example

IPX Routing Examples

This section shows examples for enabling IPX routing on interfaces with a single network and with multiple networks. It also shows how to enable and disable various combinations of routing protocols.

The following sections provide these examples:

- IPX Routing on a Single Network Example
- IPX Routing on Multiple Networks Examples
- IPX Routing Protocols Examples

IPX Routing on a Single Network Example

The following example shows how to enable IPX routing, defaulting the IPX host address to that of the first IEEE-conformance interface (in this example, Ethernet 0). Routing is then enabled on Ethernet 0 and Ethernet 1 for IPX networks 2abc and 1def, respectively.

```
ipx routing
```

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

```
interface ethernet 0
 ipx network 2abc
interface ethernet 1
 ipx network 1def
```

IPX Routing on Multiple Networks Examples

There are two ways to enable IPX on an interface that supports multiple networks. You can use subinterfaces or primary and secondary networks. This section gives an example of each.

Subinterfaces Example

The following example shows how to use subinterfaces to create four logical networks on Ethernet interface 0. Each subinterface has a different encapsulation. Any interface configuration parameters that you specify on an individual subinterface are applied to that subinterface only.

```
ipx routing
interface ethernet 0.1
 ipx network 1 encapsulation novell-ether
interface ethernet 0.2
 ipx network 2 encapsulation snap
interface ethernet 0.3
 ipx network 3 encapsulation arpa
interface ethernet 0.4
 ipx network 4 encapsulation sap
```

Note When enabling NLSP and configuring multiple encapsulations on the same physical LAN interface, you must use subinterfaces. You cannot use secondary networks.

You can administratively shut down each of the four subinterfaces separately by using the **shutdown** interface configuration command for each subinterface. The following example shows how to administratively shut down a subinterface:

```
interface ethernet 0.3
 shutdown
```

To bring down network 1, use the following commands:

```
interface ethernet 0.1
 ipx down 1
```

To bring network 1 back up, use the following commands:

```
interface ethernet 0.1
 no ipx down 1
```

To remove all the networks on the interface, use the following interface configuration commands:

```
interface ethernet 0.1
 no ipx network
interface ethernet 0.2
 no ipx network
interface ethernet 0.3
 no ipx network
interface ethernet 0.4
 no ipx network
```

Primary and Secondary Networks Example

Note The following examples discuss primary and secondary networks. In future Cisco IOS software releases, primary and secondary networks will not be supported. Use subinterfaces.

The following example shows how to use primary and secondary networks to create the same four logical networks as shown earlier in this section. Any interface configuration parameters that you specify on this interface are applied to all the logical networks. For example, if you set the routing update timer to 120 seconds, this value is used on all four networks.

```
ipx routing
interface ethernet 0
 ipx network 1 encapsulation novell-ether
 ipx network 2 encapsulation snap secondary
 ipx network 3 encapsulation arpa secondary
 ipx network 4 encapsulation sap secondary
```

Using this method to configure logical networks, if you administratively shut down Ethernet interface 0 using the **shutdown** interface configuration command, all four logical networks are shut down. You cannot bring down each logical network independently using the **shutdown** command; however, you can bring them down using the **ipx down** command.

The following example shows how to shut down network 1:

```
interface ethernet 0
 ipx down 1
```

The following example shows how to bring the network back up:

```
interface ethernet 0
 no ipx down 1
```

The following two examples show how to shut down all four networks on the interface and remove all the networks on the interface:

```
 no ipx network
```

```
 no ipx network 1
```

The following example shows how to remove one of the secondary networks on the interface (in this case, network 2):

```
 no ipx network 2
```

The following example shows how to enable IPX routing on FDDI interfaces 0.2 and 0.3. On FDDI interface 0.2, the encapsulation type is SNAP. On FDDI interface 0.3, the encapsulation type is the Novell FDDI_RAW.

```
ipx routing
interface fddi 0.2
 ipx network f02 encapsulation snap
interface fddi 0.3
 ipx network f03 encapsulation novell-fddi
```

IPX Routing Protocols Examples

Three routing protocols can run over interfaces configured for IPX: RIP, Enhanced IGRP, and NLSP. This section provides examples of how to enable and disable various combinations of routing protocols.

When you enable IPX routing with the **ipx routing** global configuration command, the RIP routing protocol is automatically enabled. The following example shows how to enable RIP on networks 1 and 2:

```
ipx routing
!
interface ethernet 0
 ipx network 1
!
interface ethernet 1
 ipx network 2
```

The following example shows how to enable RIP on networks 1 and 2 and Enhanced IGRP on network 1:

```
ipx routing
!
interface ethernet 0
 ipx network 1
!
interface ethernet 1
 ipx network 2
!
ipx router eigrp 100
 network 1
```

The following example shows how to enable RIP on network 2 and Enhanced IGRP on network 1:

```
ipx routing
!
interface ethernet 0
 ipx network 1
!
interface ethernet 1
 ipx network 2
!
ipx router eigrp 100
 ipx network 1
!
ipx router rip
 no ipx network 1
```

The following example shows how to configure NLSP on two Ethernet interfaces of the router. Note that RIP is automatically enabled on both of these interfaces. This example assumes that the encapsulation type is Ethernet 802.2.

```
ipx routing
 ipx internal-network 3
!
ipx router nlsp area1
area-address 0 0
!
interface ethernet 0
 ipx network e0 encapsulation sap
ipx nlsp area1 enable
!
interface ethernet 1
 ipx network e1 encapsulation sap
 ipx nlsp area1 enable
```

Enhanced IGRP Examples

The following sections show several examples of how to configure IPX Enhanced IGRP routing:

- IPX Enhanced IGRP Example
- IPX SAP-Incremental IGRP Example
- Enhanced IGRP SAP Update Examples
- Advertisement and Processing of SAP Update Examples
- IPX Enhanced IGRP Bandwidth Configuration Example

IPX Enhanced IGRP Example

The following example shows how to configure two interfaces for Enhanced IGRP routing in autonomous system 1:

```
ipx routing
!
interface ethernet 0
 ipx network 10
!
interface serial 0
 ipx network 20
!
ipx router eigrp 1
 network 10
 network 20
```

IPX SAP-Incremental IGRP Example

The following example shows a sample configuration for enabling the IPX SAP Enhanced IGRP:

```
ipx routing
!
interface ethernet 0
 ipx network 1
 ipx sap-incremental eigrp 1
 ipx sap-incremental split-horizon
!
ipx router eigrp 100
 network 1
```

Enhanced IGRP SAP Update Examples

If an Ethernet interface has neighbors that are all configured for Enhanced IGRP, you might want to reduce the bandwidth used by SAP packets by sending SAP updates incrementally. The following example shows how to send SAP updates incrementally:

```
ipx routing
!
interface ethernet 0
 ipx network 10
 ipx sap-incremental eigrp 1
!
interface serial 0
 ipx network 20
!
ipx router eigrp 1
 network 10
 network 20
```

The following example shows how to send only incremental SAP updates on a serial line that is configured for Enhanced IGRP:

```
ipx routing
!
interface ethernet 0
 ipx network 10
!
interface serial 0
 ipx network 20
 ipx sap-incremental eigrp 1 rsup-only
```

```
!
ipx router eigrp 1
 network 10
 network 20
```

Advertisement and Processing of SAP Update Examples

The following example shows how to cause only services from network 3 to be advertised by an Enhanced IGRP routing process:

```
access-list 1010 permit 3
access-list 1010 deny -1
!
ipx router eigrp 100
 network 3
 distribute-sap-list 1010 out
```

The following example shows how to configure the router to redistribute Enhanced IGRP into NLSP area1. Only services for networks 2 and 3 are accepted by the NLSP routing process.

```
access-list 1000 permit 2
access-list 1000 permit 3
access-list 1000 deny -1
!
ipx router nlsp area1
 redistribute eigrp
 distribute-sap-list 1000 in
```

IPX Enhanced IGRP Bandwidth Configuration Example

The following example shows how to configure the bandwidth used by IPX Enhanced IGRP. In this example, Enhanced IGRP process 109 is configured to use a maximum of 25 percent (or 32-kbps) of a 128-kbps circuit:

```
interface serial 0
 bandwidth 128
 ipx bandwidth-percent eigrp 109 25
```

The following example shows how to configure the bandwidth of a 56-kbps circuit to 20 kbps for routing policy reasons. The Enhanced IGRP process 109 is configured to use a maximum of 200 percent (or 40 kbps) of the circuit.

```
interface serial 1
 bandwidth 20
 ipx bandwidth-percent eigrp 109 200
```

IPX over WAN Examples

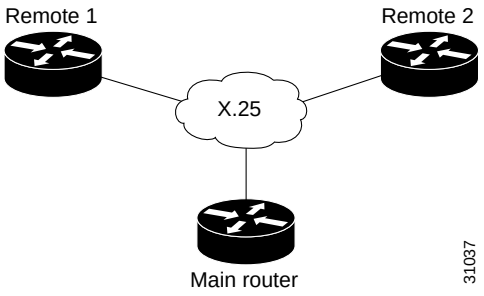
The following sections show examples of how to configure IPX over WAN and dial interfaces.

- IPX over a WAN Interface Example
- IPX over DDR Example

IPX over a WAN Interface Example

When you configure the Cisco IOS software to transport IPX packets over a serial interface that is running a WAN protocol such as X.25 or PPP, you specify how the packet will be encapsulated for transport. This encapsulation is not the same as the encapsulation used on an IPX LAN interface. Figure 1 illustrates IPX over a WAN interface.

Figure 1 IPX over a WAN Interface



The following example shows how to configure a serial interface for X.25 encapsulation and for several IPX subinterfaces used in a nonmeshed topology:

Configuration for Main Router

```
hostname Main
!
no ip routing
novell routing 0000.0c17.d726
!
interface ethernet 0
 no ip address
 Novell network 100
 media-type 10BaseT
!
interface serial 0
 no ip address
 shutdown
!
interface serial 1
 no ip address
 encapsulation x25
 x25 address 33333
 x25 htc 28
!
interface serial 1.1 point-to-point
 no ip address
 novell network 2
 x25 map novell 2.0000.0c03.a4ad 11111 BROADCAST
!
interface serial 1.2 point-to-point
 no ip address
 novell network 3
 x25 map novell 3.0000.0c07.5e26 55555 BROADCAST
```

Configuration for Router 1

```
hostname Remote1
!
no ip routing
```

```
novell routing 0000.0c03.a4ad
!
interface ethernet 0
 no ip address
 novell network 1
!
interface serial 0
 no ip address
 encapsulation x25
 novell network 2
 x25 address 11111
 x25 htc 28
 x25 map novell 2.0000.0c17.d726 33333 BROADCAST
```

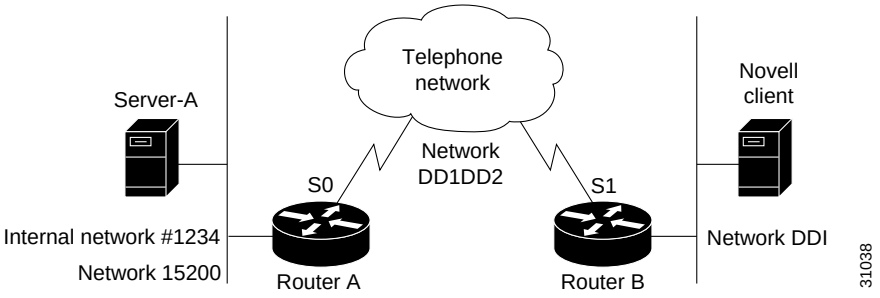
Configuration for Router 2

```
hostname Remote2
!
no ip routing
novell routing 0000.0c07.5e26
!
interface ethernet 0
 no ip address
 novell network 4
 media-type 10BaseT
!
interface serial 0
 no ip address
 shutdown
!
interface serial 1
 no ip address
 encapsulation x25
 novell network 3
 x25 address 55555
 x25 htc 28
 x25 map novell 3.0000.0c17.d726 33333 BROADCAST
```

IPX over DDR Example

In the configuration shown in Figure 2, an IPX client is separated from its server by a DDR telephone line.

Figure 2 IPX over DDR Configuration



Routing and service information is sent every 60 seconds. The output RIP and SAP filters defined in this example filter these updates, preventing them from being sent between Router A and Router B. If you forwarded these packets, each of the two routers would need to telephone the other once every 60 seconds. On a serial link whose charges are based on the number of packets sent, this activity is generally not desirable. (This problem may not occur on a dedicated serial line.)

Once the server and client have established contact, the server will send watchdog keepalive packets regularly. When SPX is used, both the server and the client send keepalive packets whose purpose is to ensure that the connection between the server and the client is still functional; these packets contain no other information. Servers send watchdog packets approximately every 5 minutes.

If Router A were allowed to forward the keepalive packets of the server to Router B, Router A would need to telephone Router B every 5 minutes just to send these packets. Again, on a serial link whose charges are based on the number of packets sent, this activity is generally not desirable. Instead of having Router A telephone Router B only to send keepalive packets, you can enable watchdog spoofing on Router A. The result will be that when the server connected to this router sends keepalive packets, Router A will respond on behalf of the remote client (the client connected to Router B). When SPX is used, enable spoofing of SPX keepalive packets on both routers A and B to inhibit the sending of them because both the server and the client send keepalive packets.

Use the **ipx watchdog-spoof** interface configuration command to enable and set the duration of watchdog spoofing. You can specify the number of consecutive hours spoofing is to stay enabled and the number of minutes spoofing is to stay disabled. Use this command only on a serial interface whose fast switching and autonomous switching are disabled.

The following example shows how to configure Router A. Watchdog spoofing will be enabled for 1 hour and disabled for 20 minutes, allowing the server to clean up inactive connections before being enabled again.

```
ipx routing 0000.0c04.4878
!
interface Ethernet0
     ipx network 15200
!
interface Serial0
! PPP encap for DDR(recommended)
 encapsulation ppp
 ipx network DD1DD2
! Kill all rip updates
ipx output-network-filter 801
! Kill all sap updates
ipx output-sap-filter 1001
! fast-switching off for watchdog spoofing
no ipx route-cache
! Don't listen to rip
ipx router-filter 866
! IPX watchdog spoofing
ipx watchdog-spoof 1 20
!SPX watchdog spoofing
ipx spx-spoof
! Turn on DDR
dialer in-band
 dialer idle-timeout 200
 dialer map IP 198.92.96.132 name R13 7917
 dialer map IPX DD1DD2.0000.0c03.e3c3 7917
  dialer-group 1
ppp authentication chap
! Chap authentication required
pulse-time 1
!
access-list 801 deny  FFFFFFFF
```

```
access-list 866 deny  FFFFFFFF
!  Serialization packets
access-list 900 deny  0 FFFFFFFF 0 FFFFFFFF 457
! RIP packets
access-list 900 deny  1 FFFFFFFF 453 FFFFFFFF 453
! SAP packets
access-list 900 deny  4 FFFFFFFF 452 FFFFFFFF 452
! Permit everything else
access-list 900 permit -1 FFFFFFFF 0 FFFFFFFF 0
!
access-list 1001 deny  FFFFFFFF
!
! Static ipx route for remote network
ipx route DD1 DD1DD2.0000.0c03.e3c3
!
!
! IPX will trigger the line up (9.21 and later)
dialer-list 1 list 900
```

IPX Network Access Examples

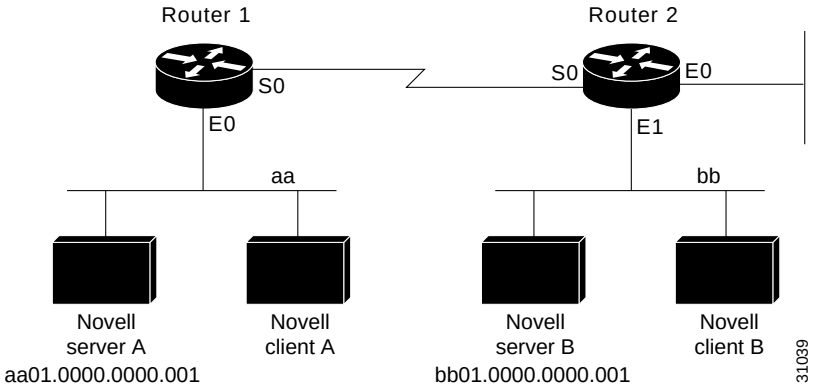
The following sections show examples of how to control access to your IPX network. The sections show the configurations for various access lists and filters.

- IPX Network Access Example
- Standard Named Access List Example
- Extended Named Access List Time Range Example
- SAP Input Filter Example
- SAP Output Filter Example
- GGS SAP Response Filter Example
- IPX NetBIOS Filter Examples

IPX Network Access Example

Using access lists to manage traffic routing is a powerful tool in overall network control. However, it requires a certain amount of planning and the appropriate application of several related commands. Figure 3 illustrates a network featuring two routers on two network segments.

Figure 3 Novell IPX Servers Requiring Access Control



Suppose you want to prevent clients and servers on Network aa from using the services on Network bb, but you want to allow the clients and servers on Network bb to use the services on Network aa. To achieve this configuration, you would need an access list on Ethernet interface 1 on Router 2 that blocks all packets coming from Network aa and destined for Network bb. You would not need any access list on Ethernet interface 0 on Router 1.

The following example shows how to configure Ethernet interface 1 on Router 2:

```
ipx routing
access-list 800 deny aa bb01
access-list 800 permit -1 -1
interface ethernet 1
 ipx network bb
 ipx access-group 800
```

The following example shows how you can accomplish the same result as the previous example more efficiently by placing an input filter on interface Ethernet 0 of Router 1. You can also place the same output filter on Router 1, interface serial 0.

```
ipx routing
access-list 800 deny aa bb01
access-list 800 permit -1 -1
interface ethernet 0
 ipx network aa
 ipx access-group 800 in
```

Note When using access control list logging on an interface with fast switching turned on, packets that match the access list (and thus need to be logged) are slow switched, not fast switched.

Logging Access Control List Violations

The following example shows how you can keep a log of all access control list violations by using the keyword **log** at the end of the **access-list** command:

```
access-list 907 deny -1 -1 0 100 0 log
```

The previous example denies and logs all packets that arrive at the router from any source in any protocol from any socket to any destination on network 100.

The following example shows a log entry for the **access-list** command:

```
%IPX-6-ACL: 907 deny SPX B5A8 50.0000.0000.0001 B5A8 100.0000.0000.0001 10 pkts
```

In this example, ten SPX packets were denied because they matched access list number 907. The packets were coming from socket B5A8 on networks 50.0000.0000.0001 and were destined for socket B5A8 on network 100.0000.0000.0001.

Standard Named Access List Example

The following example shows how to create a standard access list named fred. It denies communication with only IPX network number 5678.

```
ipx access-list standard fred
 deny 5678 any
 permit any
```

Extended Named Access List Time Range Example

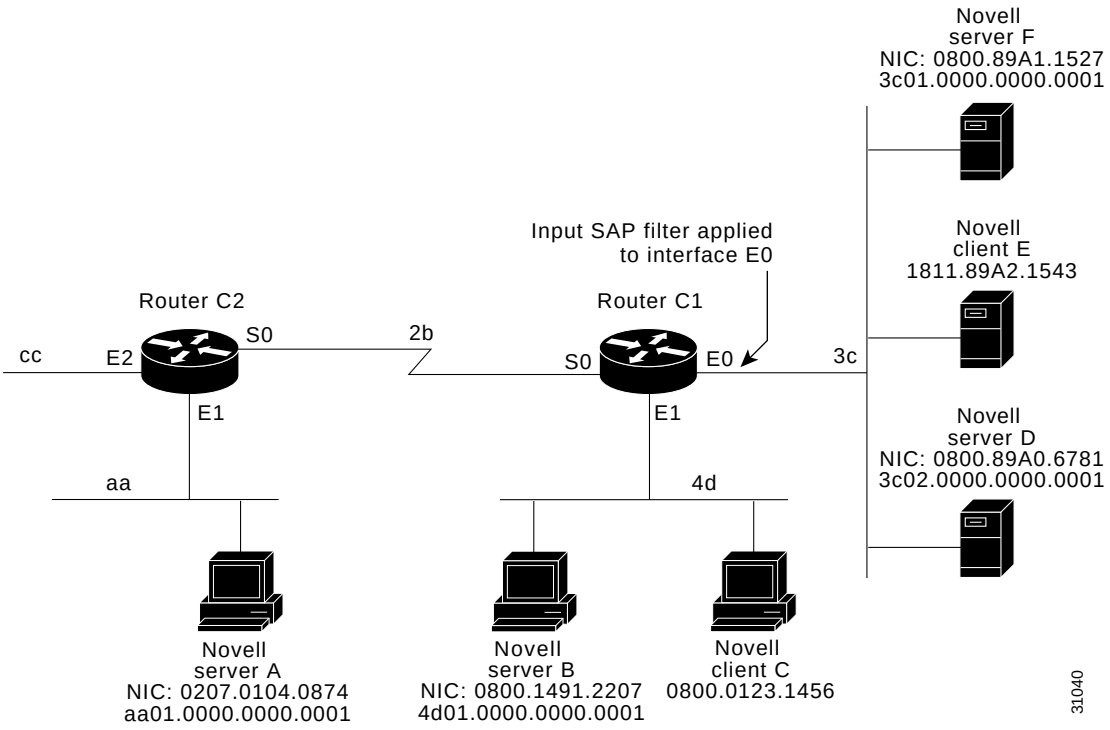
The following example shows how to create an extended access list named test. It permits SPX traffic only on Monday through Friday between the hours of 8:00 a.m. and 6:00 p.m.

```
time-range no-spx
 periodic weekdays 8:00 to 18:00
!
ipx access-list extended test
 permit spx any all any all time-range no spx
```

SAP Input Filter Example

SAP input filters allow a router to determine whether to accept information about a service. Router C1, illustrated in Figure 4, will not accept and, consequently not advertise, any information about Novell server F. However, Router C1 will accept information about all other servers on the network 3c. Router C2 receives information about servers D and B.

Figure 4 SAP Input Filter



The following example shows how to configure Router C1. The first line denies server F, and the second line accepts all other servers.

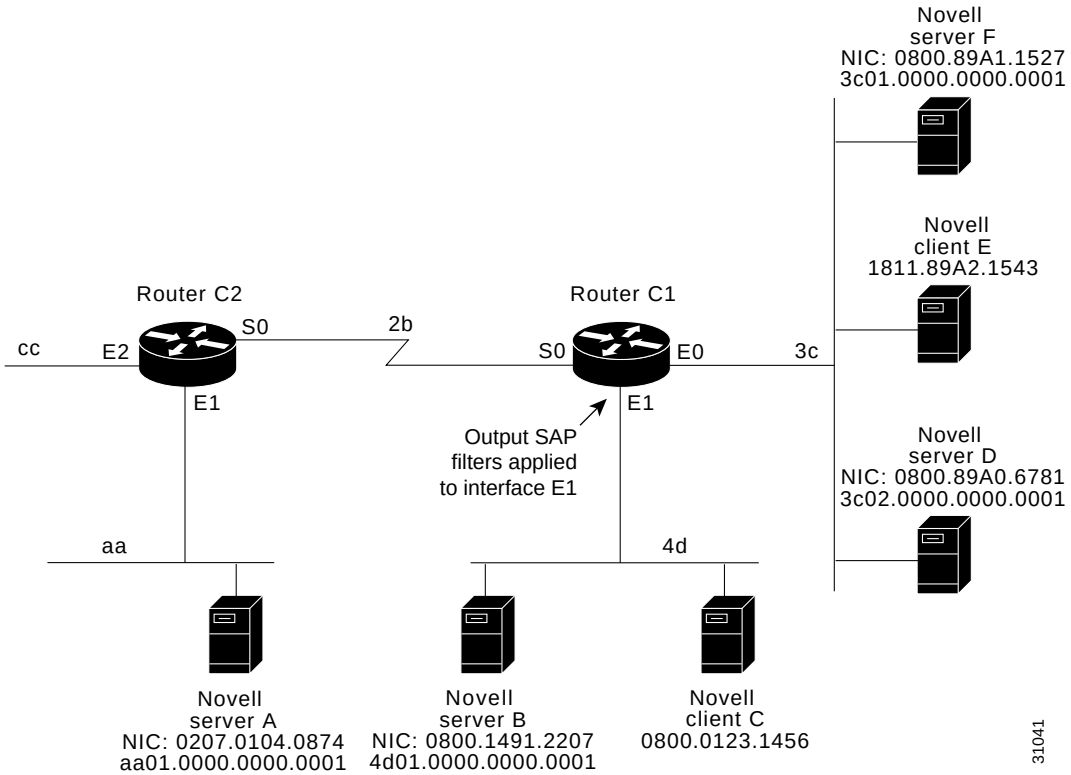
```
access-list 1000 deny 3c01.0000.0000.0001
access-list 1000 permit -1
interface ethernet 0
 ipx network 3c
 ipx input-sap-filter 1000
interface ethernet 1
 ipx network 4d
interface serial 0
 ipx network 2b
```

Note NetWare versions 3.11 and later use an internal network and node number as their address for access list commands (the first configuration command in this example).

SAP Output Filter Example

SAP output filters are applied prior to the Cisco IOS software sending information out a specific interface. In the example that follows, Router C1 (illustrated in Figure 5) is prevented from advertising information about Novell server A out interface Ethernet 1, but can advertise server A on network 3c.

Figure 5 SAP Output Filter



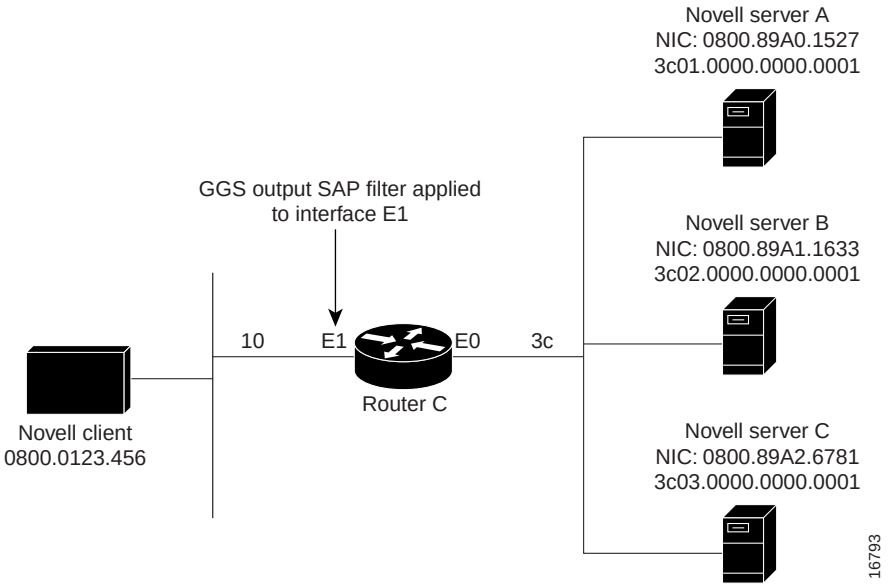
The following example shows how to configure Router C1. The first line denies server A. All other servers are permitted.

```
access-list 1000 deny aa01.0000.0000.0001
access-list 1000 permit -1
interface ethernet 0
 novell net 3c
interface ethernet 1
 ipx network 4d
 ipx output-sap-filter 1000
interface serial 0
 ipx network 2b
```

GGS SAP Response Filter Example

GGS SAP response filters as shown in Figure 6 allow a router to determine whether to forward information it receives about a service.

Figure 6 GGS SAP Response Filter



The following example shows how to configure GGS SAP response filters for Router C. When the client issues a GGS request, the output GGS filter denies a response from Novell Server A and permits responses from Novell servers B and C.

```
access-list 1000 deny 3c01.0000.0000.0001
access-list 1000 permit -1
interface ethernet 0
 ipx network 3c
interface ethernet 1
 ipx output-ggs-filter 1000
 ipx network 10
```

IPX NetBIOS Filter Examples

The following example shows how to use a NetBIOS host name to filter IPX NetBIOS frames. The example denies all outgoing IPX NetBIOS frames with a NetBIOS host name of Boston on Ethernet interface 0.

```
netbios access-list host token deny Boston
netbios access-list host token permit *
!
ipx routing 0000.0c17.d45d
!
interface ethernet 0
 ipx network 155 encapsulation ARPA
 ipx output-rip-delay 60
 ipx triggered-rip-delay 30
 ipx output-sap-delay 60
 ipx triggered-sap-delay 30
 ipx type-20-propagation
 ipx netbios output-access-filter host token
 no mop enabled
!
interface ethernet 1
 no ip address
 ipx network 105
!
```

```
interface fddi 0
 no ip address
 no keepalive
ipx network 305 encapsulation SAP
!
interface serial 0
 no ip address
 shutdown
!
interface serial 1
 no ip address
 no keepalive
 ipx network 600
 ipx output-rip-delay 100
 ipx triggered-rip-delay 60
 ipx output-sap-delay 100
 ipx triggered-sap-delay 60
 ipx type-20-propagation
```

The following example shows how to use a byte pattern to filter IPX NetBIOS frames. This example permits IPX NetBIOS frames from IPX network numbers that end in 05, which means that all IPX NetBIOS frames from Ethernet interface 1 (network 105) and FDDI interface 0 (network 305) will be forwarded by serial interface 0. However, this interface will filter out and not forward all frames from Ethernet interface 0 (network 155).

```
netbios access-list bytes finigan permit 2 **05
!
ipx routing 0000.0c17.d45d
!
ipx default-output-rip-delay 1000
ipx default-triggered-rip-delay 100
ipx default-output-sap-delay 1000
ipx default-triggered-sap-delay 100
!
interface ethernet 0
 ipx network 155 encapsulation ARPA
 ipx output-rip-delay 55
 ipx triggered-rip-delay 55
 ipx output-sap-delay 55
 ipx triggered-sap-delay 55
 ipx type-20-propagation
 media-type 10BaseT
!
interface ethernet 1
 no ip address
 ipx network 105
 ipx output-rip-delay 55
 ipx triggered-rip-delay 55
 ipx output-sap-delay 55
 ipx triggered-sap-delay 55
 media-type 10BaseT
!
interface fddi 0
 no ip address
 no keepalive
 ipx network 305 encapsulation SAP
 ipx output-sap-delay 55
 ipx triggered-sap-delay 55
!
interface serial 0
 no ip address
 shutdown
!
```

```
interface serial 1
 no ip address
 no keepalive
 ipx network 600
 ipx type-20-propagation
 ipx netbios input-access-filter bytes finigan
```

Helper Facilities to Control Broadcast Examples

The following sections show examples of how to control broadcast messages on IPX networks:

- Forwarding to an Address Example
- Forwarding to All Networks Example
- All-Nets Flooded Broadcast Example

Note that in the following examples, packet Type 2 is used. This type has been chosen arbitrarily; the actual type to use depends on the specific application.

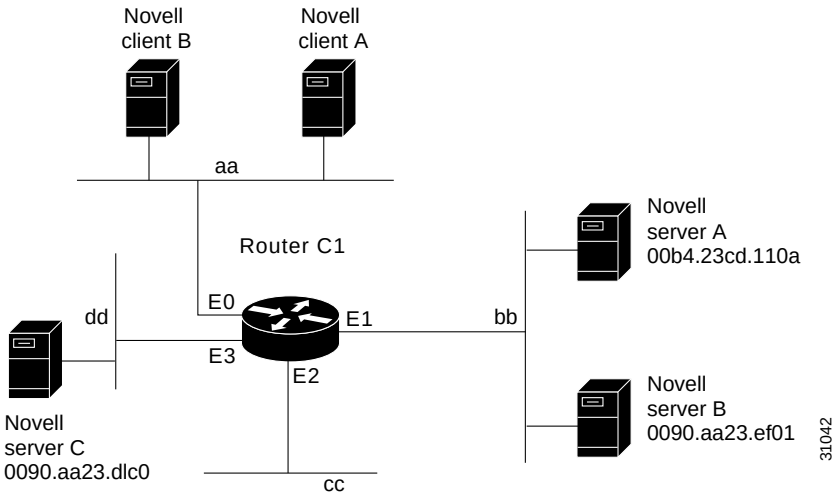
Forwarding to an Address Example

All broadcast packets are normally blocked by the Cisco IOS software. However, Type 20 propagation packets may be forwarded, subject to certain loop-prevention checks. Other broadcasts may be directed to a set of networks or a specific host (node) on a segment. The following examples illustrate these options.

Figure 7 shows a router (C1) connected to several Ethernet interfaces. In this environment, all IPX clients are attached to segment aa, while all servers are attached to segments bb and dd. In controlling broadcasts, the following conditions are to be applied:

- Only Type 2 and Type 20 broadcasts are to be forwarded.
- The IPX clients on network aa are allowed to broadcast via Type 2 to any server on networks bb and dd.
- The IPX clients are allowed to broadcast via Type 20 to any server on network dd.

Figure 7 IPX Clients Requiring Server Access Through a Router



The following example shows how to configure the router shown in Figure 7. The first line permits broadcast traffic of Type 2 from network aa. The interface and network commands configure each specific interface. The **ipx helper-address** interface configuration commands permit broadcast forwarding from network aa to bb and from network aa to dd. The helper list allows Type 2 broadcasts to be forwarded. (Note that Type 2 broadcasts are chosen as an example only. The actual type to use depends on the specific application.) The **ipx type-20-propagation** interface configuration command is also required to allow Type 20 broadcasts. The IPX helper-list filter is applied to both the Type 2 packets forwarded by the helper-address mechanism and the Type 20 packets forwarded by Type 20 propagation.

```
access-list 900 permit 2 aa
interface ethernet 0
 ipx network aa
 ipx type-20-propagation
 ipx helper-address bb.ffff.ffff.ffff
 ipx helper-address dd.ffff.ffff.ffff
 ipx helper-list 900
interface ethernet 1
 ipx network bb
interface ethernet 3
 ipx network dd
 ipx type-20-propagation
```

This configuration means that any network that is downstream from network aa (for example, some arbitrary network aa1) will not be able to broadcast (Type 2) to network bb through Router C1 unless the routers partitioning networks aa and aa1 are configured to forward these broadcasts with a series of configuration entries analogous to the example provided for Figure 7. These entries must be applied to the input interface and be set to forward broadcasts between directly connected networks. In this way, such traffic can be passed along in a directed manner from network to network. A similar situation exists for Type 20 packets.

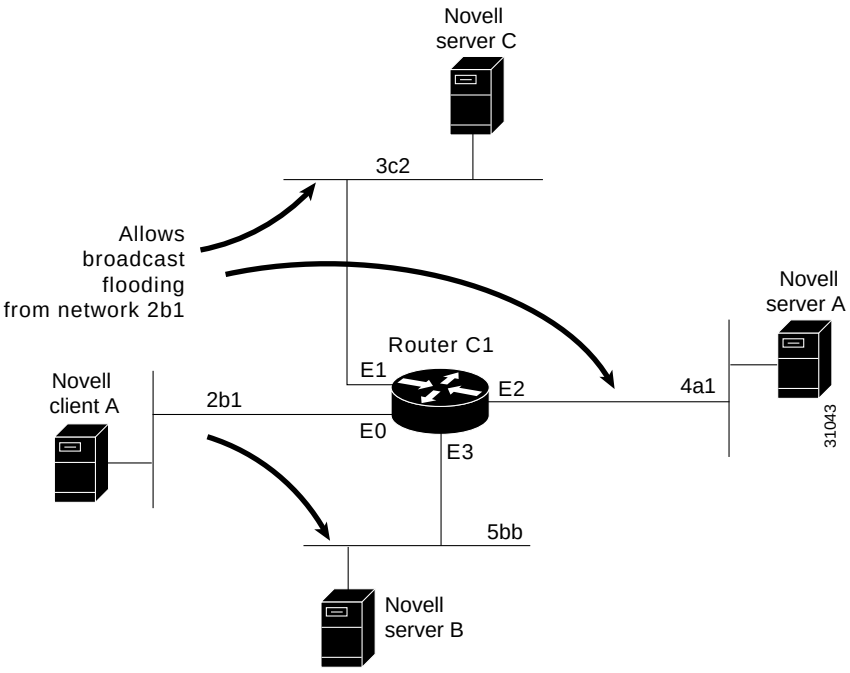
The following example shows how to rewrite the **ipx helper-address** interface configuration command line to direct broadcasts to server A:

```
ipx helper-address bb.00b4.23cd.110a
! Permits node-specific broadcast forwarding to
! Server A at address 00b4.23cd.110a on network bb.
```

Forwarding to All Networks Example

In some networks, it might be necessary to allow client nodes to broadcast to servers on multiple networks. If you configure your router to forward broadcasts to all attached networks, you are flooding the interfaces. In the environment illustrated in Figure 8, client nodes on network 2b1 must obtain services from IPX servers on networks 3c2, 4a1, and 5bb through Router C1. To support this requirement, use the flooding address (-1.ffff.ffff.ffff) in your **ipx helper-address** interface configuration command specifications.

Figure 8 Type 2 Broadcast Flooding



The first line in the following example shows how to permit traffic of Type 2 from network 2b1. Then the first interface is configured with a network number. The all-nets helper address is defined and the helper list limits forwarding to Type 2 traffic. Type 2 broadcasts from network 2b1 are forwarded to all directly connected networks. All other broadcasts, including Type 20, are blocked. To permit broadcasts, delete the **ipx helper-list** entry. To allow Type 20 broadcast, enable the **ipx type-20-propagation** interface configuration command on all interfaces.

```
access-list 901 permit 2 2b1
interface ethernet 0
 ipx network 2b1
 ipx helper-address -1.ffff.ffff.ffff
 ipx helper-list 901
interface ethernet 1
 ipx network 3c2
interface ethernet 2
 ipx network 4a1
interface ethernet 3
 ipx network 5bb
```

All-Nets Flooded Broadcast Example

The following example shows how to configure all-nets flooding on an interface. As a result of this configuration, Ethernet interface 0 will forward all broadcast messages (except Type 20) to all the networks it knows how to reach. This flooding of broadcast messages might overwhelm these networks with so much broadcast traffic that no other traffic may be able to pass on them.

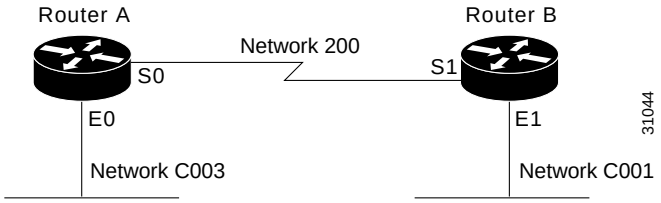
```
interface ethernet 0
 ipx network 23
 ipx helper-address -1.FFFF.FFFF.FFFF
```

IPX Accounting Example

The following example shows how to configure two Ethernet network segments that are connected via a serial link (see Figure 9). On Router A, IPX accounting is enabled on both the input and output interfaces (that is, on Ethernet interface 0 and serial interface 0), which means that statistics are gathered for traffic traveling in both directions (that is, out to the Ethernet network and out the serial link).

On Router B, IPX accounting is enabled only on the serial interface and not on the Ethernet interface, which means that statistics are gathered only for traffic that passes out the router on the serial link. Also, the accounting threshold is set to 1000, which means that IPX accounting will track all IPX traffic passing through the router up to 1000 source and destination pairs.

Figure 9 IPX Accounting Example



Configuration for Router A

```
ipx routing
interface ethernet 0
 no ip address
 ipx network C003
 ipx accounting
interface serial 0
 no ip address
 ipx network 200
 ipx accounting
```

Configuration for Router B

```
ipx routing
interface ethernet 1
 no ip address
 no keepalive
 ipx network C001
 no mop enabled
interface serial 1
 no ip address
 ipx network 200
 ipx accounting
 ipx accounting-threshold 1000
```