



# NX64000 Configuration Guide

Release 1.7

---

**Copyright© 2001 Lucent Technologies. All Rights Reserved.**

This material is protected by the copyright laws of the United States and other countries. It may not be reproduced, distributed, or altered in any fashion by any entity (either internal or external to Lucent Technologies), except in accordance with applicable agreements, contracts or licensing, without the express written consent of Lucent Technologies.

**Notice.** Every effort was made to ensure that the information in this document was complete and accurate at the time of printing. However, information is subject to change.

**Trademarks.** NX64000, NX32000, NX-IS, NX-MS, and NXtreme are trademarks of Lucent Technologies. Other trademarks and trade names mentioned in this document belong to their respective owners.

**Limited Warranty.** Lucent Technologies provides a limited warranty to this product.

**Ordering Information.** To order copies of this document, contact your Lucent Technologies account representative or visit the Lucent Product Documentation web site at <http://www.lucentdocs.com>.

**Technical Support.** For technical support and other services, contact Lucent Customer Support. (See the Customer Support contact information at the end of this book.)

# Contents

## About this Guide

Intended Audience.....	xix
Conventions.....	xix
Product Documentation.....	xx

## Chapter 1

### Configuration Overview

System Hardware .....	1-1
Line Card–Processor Engine Architecture .....	1-2
Packet Forwarding.....	1-3
Redundancy Features.....	1-3
Agency Specification Compliance and Operating Environment .....	1-3
System Software .....	1-3
Network Protocols and Services .....	1-4
Routing Protocols .....	1-5
Service Classes .....	1-5
Preparing to Configure the System.....	1-5
Configuration Examples .....	1-6
Configuration Tasks .....	1-6
Updates to System Configuration.....	1-7

## Chapter 2

### Using the Command Line Interface (CLI)

Connecting to the CLI.....	2-1
Ethernet Connection .....	2-1
Console Connection .....	2-1
Setting up a Terminal Connection .....	2-2
Setting up a Modem Connection .....	2-2
Command Overview.....	2-3
Privilege Levels.....	2-3
Command Prompt Levels.....	2-4
Using Context-Sensitive Help .....	2-5

## Chapter 3

### Working with System Files

Single Image File.....	3-1
Modules for Routing Protocols .....	3-2
Backup Copy of an Image File.....	3-2
System Configuration File.....	3-2

File and Directory Organization.....	3-3
Managing System Software and Files .....	3-4
Rebooting the System.....	3-5
Working with Software Image Files .....	3-5
Changing the Location of an Image File .....	3-5
Working with a Backup Image of the Software.....	3-6
Working with Configuration Files.....	3-7
Saving System Configuration to the Startup Configuration File .....	3-7
Changing the Location of the Startup Configuration File.....	3-7
Synchronizing Files Between a Primary and Secondary RCP .....	3-7
Working with Routing Modules .....	3-8
Activating a Software Routing Module .....	3-8
Stopping a Software Routing Module .....	3-9
Working with Files and Directories .....	3-9
Working with Directories .....	3-10
Working with Files .....	3-11
Working with Files Locally .....	3-11
Working with Files on the Network .....	3-11

## Chapter 4      System Security and Access

Key Features .....	4-1
Technology Concepts .....	4-2
Password Authentication.....	4-2
Password Encryption .....	4-3
Telnet Access .....	4-3
RADIUS Authorization .....	4-4
Primary Authentication Server .....	4-4
Secondary Authentication Server .....	4-4
Authentication Database.....	4-4
Username and Password Configuration.....	4-5
Configuring Usernames and Passwords.....	4-5
Verifying Username Configuration.....	4-6
Telnet Access Configuration .....	4-6
Configuring Telnet Access .....	4-7
Clearing a Login Session.....	4-8
RADIUS Configuration .....	4-8
Configuring RADIUS on the System .....	4-9
Verifying RADIUS Configuration .....	4-10

## Chapter 5      Working with the Log and Debug Utilities

Key Features .....	5-1
Technology Concepts .....	5-1
Message Severity Levels .....	5-2
Log Buffers .....	5-2
Initial Configuration for the Log Utility.....	5-2
Displaying Log Messages .....	5-3
Module Support for the Debug Utility.....	5-4
Logging Configuration .....	5-4
Setting up Logging to a Syslog Server .....	5-6
Setting up Logging to a File.....	5-6
Viewing Log Messages .....	5-7

---

	Viewing Log Messages from the CLI .....	5-7
	Viewing Log Messages from a Console .....	5-7
	Viewing Log Messages from a Telnet Session .....	5-8
	Viewing Log Messages Inline .....	5-8
	Getting Card Information .....	5-9
	Changing Buffer Sizes .....	5-9
	Verifying Log Configuration .....	5-9
	Debugging Configuration .....	5-9
	Enabling Debugging for a Protocol .....	5-11
	Getting Information for Debugging .....	5-11
Chapter 6	Setting up System Monitoring	
	Technology Concepts .....	6-1
	SNMP Manager and Agent Model .....	6-1
	SNMP Manager .....	6-2
	SNMP Agent .....	6-2
	SNMP Polls and Traps .....	6-2
	SNMP Communities .....	6-3
	Management Information Base (MIB) .....	6-4
	Managed Object ID .....	6-5
	SNMP Configuration .....	6-5
	Configuring an SNMP Community .....	6-7
	Configuring SNMP Contact and Location Information .....	6-7
	Clearing SNMP Counters .....	6-7
	Verifying and Monitoring SNMP Configuration .....	6-8
Chapter 7	Cards and Interfaces	
	System Interface Cards .....	7-1
	Interface Speeds .....	7-3
	Removing and Replacing Modules .....	7-4
	Switch Fabric Modules .....	7-4
	SONET/SDH Timing/Alarm Modules .....	7-4
	Route Control Processor (RCP) .....	7-4
	PCMCIA Card .....	7-5
	Synchronizing the RCP files .....	7-5
	Upgrading Software .....	7-5
	Understanding the Inet on Ethernet Address .....	7-6
	Input/Output Cards .....	7-6
	Optical Interface Specifications .....	7-7
	Working with System Cards .....	7-9
	Setting the Threshold .....	7-10
	Setting the Failurecount .....	7-10
	Deactivating and Reactivating the Cards .....	7-10
	Configuring Interfaces .....	7-11
	Ethernet, Loopback, Tunnel, and Null Interfaces .....	7-12
	Quality of Service for Traffic Management .....	7-13
	Interface Configuration Commands .....	7-13
	Basic Interface Configuration Tasks .....	7-14
	Creating the Interface .....	7-15
	Setting Operating Compatibility .....	7-15
	Configuring Subinterfaces .....	7-16

Configuring Gigabit Ethernet Interfaces .....	7-16
Verifying Cards and Interface Configuration .....	7-18

## Chapter 8

### Frame Relay Configuration

Key Features .....	8-1
Technology Concepts .....	8-2
RFCs and Standards.....	8-2
Virtual Circuits.....	8-2
Data Link Connection Identifier (DLCI) .....	8-3
Local Management Interface (LMI) Protocol .....	8-3
Link Access Procedure for Frame Relay (LAPF) .....	8-5
Encapsulation .....	8-5
Frame Relay Routing and Switching .....	8-5
Frame Relay Dynamic and Static Address Mapping.....	8-6
Dynamic Address Mapping .....	8-6
Static Address Mapping.....	8-6
Quality of Service (QoS) and Traffic Management .....	8-6
Cyclic Redundancy Checking (CRC) .....	8-7
Frame Relay Network Interfaces/Devices.....	8-7
Debugging and Logging Facilities .....	8-7
Frame Relay Frame Formats .....	8-8
Frame Relay Frame.....	8-8
Frame Relay Frame Header.....	8-8
LMI Frame Format .....	8-9
Frame Relay Configuration.....	8-9
Basic Frame Relay Configuration Tasks .....	8-12
Basic Frame Relay Configuration.....	8-13
Configuring Dynamic or Static Address Mapping .....	8-15
Dynamic Address Mapping .....	8-15
Static Address Mapping.....	8-16
Local Management Interface (LMI) .....	8-16
LAPF Configuration.....	8-17
Quality of Service (QoS) and Traffic Management Configuration .....	8-18
Verifying and Monitoring Frame Relay Connections .....	8-19
Implementation Differences.....	8-20

## Chapter 9

### Asynchronous Transfer Mode (ATM) Configuration

Key Features .....	9-1
Technology Concepts .....	9-1
Standards.....	9-2
Virtual Circuits and Virtual Paths.....	9-2
Permanent Virtual Circuits (PVCs) .....	9-2
Configuring PVCs.....	9-2
Using VCDs and VPI/VCI Pairs.....	9-3
Using Sub-aggregation .....	9-3
Deleting or Changing a PVC .....	9-3
Virtual Path Identifiers (VPIs) and Virtual Channel Identifiers (VCIs).....	9-3
VPI/VCI Pairs and the Virtual Circuit Descriptor (VCD) .....	9-4
Determining VCs and VPs .....	9-4
ATM Adaptation Layer .....	9-5
AAL and Encapsulation.....	9-6

---

	ATM UNI Cell Format.....	9-6
	Debugging and Logging Facilities .....	9-7
	ATM Configuration.....	9-8
	Basic ATM Configuration Tasks.....	9-10
	Setting Up Port (Physical) Interfaces .....	9-10
	Managing Virtual Channels.....	9-11
	VP Bit Range .....	9-11
	Configuring VC and VP Parameters .....	9-12
	Configuring Point-to-Point PVCs .....	9-14
	Configuring Point-to-Multipoint PVCs .....	9-20
	Using Map Lists for Static Routing .....	9-23
	Verifying a Map List .....	9-25
	Verifying ATM Configuration .....	9-25
Chapter 10	Point-to-Point Protocol Configuration	
	Key Features .....	10-1
	Technology Concepts .....	10-1
	Standards.....	10-1
	Encapsulation .....	10-2
	Link Control Protocol .....	10-2
	Network Control Protocols.....	10-2
	Debugging and Logging Facilities .....	10-2
	PPP Interface Configuration.....	10-2
	Configuring PPP on an Unnumbered Interface .....	10-4
	Verifying and Monitoring PPP Connections .....	10-5
Chapter 11	Internet Protocol Configuration	
	Key Features .....	11-1
	Technology Concepts .....	11-2
	IP Addressing.....	11-2
	Subnet Bit Masks .....	11-3
	Access Lists .....	11-4
	Internet Control Protocols.....	11-4
	Domain Name System (DNS).....	11-4
	Address Resolution Protocol (ARP) .....	11-4
	IP Routing.....	11-5
	Routing Protocols.....	11-5
	Differentiated Services .....	11-6
	Debugging and Logging Facilities .....	11-6
	IP Header Format .....	11-7
	Internet Protocol Configuration.....	11-9
	Configuration Overview.....	11-10
	DNS Server IP Address Configuration Example .....	11-12
	Assigning an IP Address to the Ethernet Port.....	11-12
	IP Unnumbered Configuration Example.....	11-13
	Changing an IP address on a Loopback Interface.....	11-14
	Verifying and Monitoring IP Connections .....	11-14
Chapter 12	Access List Configuration	
	Key Features .....	12-1
	Technology Concepts .....	12-1

---

---

Access Lists for Protocol Configuration .....	12-2
Packet Filters for Traffic Management.....	12-2
Limiting Network Access.....	12-2
Supporting Network Security .....	12-2
Filtering Process.....	12-2
Filtering Criteria .....	12-2
Standard Access Lists .....	12-3
Extended Access Lists.....	12-3
Monitoring Filtered Traffic.....	12-8
Filter Application.....	12-8
Access List Activation on Interfaces.....	12-9
Packet Formats.....	12-9
IP Header Fields.....	12-10
TCP Header Fields.....	12-11
UDP Header Fields.....	12-11
Access List Configuration.....	12-12
Configuring a Standard Access List .....	12-13
Configuring Extended Access Lists.....	12-13
Setting Rules for IP, UDP, and ICMP Access.....	12-13
Setting Rules for IP, TCP, UDP, and ICMP Access.....	12-14
Configuring an Extended Access List with Logging.....	12-15
Assigning Extended Access Lists to Interfaces .....	12-16
Verifying Access List Configuration .....	12-16

## Chapter 13 Quality of Service Configuration for Traffic Management

Key Features .....	13-2
Technology Concepts .....	13-2
Traffic Classification.....	13-2
Traffic Requirements.....	13-2
Priority Lists .....	13-3
Priority Queues .....	13-3
Bandwidth Management.....	13-4
Queue Bandwidth.....	13-5
Protocol Overhead .....	13-5
Congestion Management.....	13-6
Connection Admission Control (CAC) .....	13-6
Congestion Watermark .....	13-6
Weighted Random Early Detection (WRED) .....	13-6
Packet Discard .....	13-7
Traffic Scheduling.....	13-7
Examples of Traffic Scheduling.....	13-8
Debugging and Logging Facilities .....	13-10
Differentiated Services Field .....	13-10
QoS Configuration .....	13-11
Basic QoS Configuration Tasks.....	13-12
Creating a Priority List.....	13-13
Deleting Priority Lists .....	13-14
Assigning a Priority Group on an Interface .....	13-14
Configuring WRED on an Outgoing Interface.....	13-14
Configuring Queue Bandwidth .....	13-15
Changing the Line Rate.....	13-16
Verifying QoS Configuration and Monitoring Traffic Transmission .....	13-17



Chapter 14	Multi-protocol Label Switching (MPLS) Configuration	
	Key Features .....	14-1
	Technology Concepts .....	14-2
	RFCs and Standards .....	14-2
	MPLS Labels .....	14-2
	Label-switched Routers .....	14-3
	Label-switched Paths .....	14-3
	Path components .....	14-3
	Path types .....	14-4
	MPLS Packet Forwarding .....	14-5
	Interfaces .....	14-5
	Label Switching .....	14-6
	Path Merge .....	14-7
	Integration with Routing Protocols .....	14-7
	Traffic Engineering .....	14-8
	LSP Mode .....	14-8
	Exchanging Routing Control Packets .....	14-8
	Service Classes for MPLS LSPs .....	14-8
	Per-hop Behavior Groups .....	14-8
	Diff-Serv Map Classes .....	14-10
	Quality of Service for MPLS LSPs .....	14-11
	Debugging and Logging for MPLS and RSVP .....	14-11
	Packet Format .....	14-11
	MPLS Configuration .....	14-12
	Static Label-switched Path Configuration .....	14-14
	Setting up the End Points for an Incoming and Outgoing LSP .....	14-15
	Setting Up LSP Integration with an IGP .....	14-16
	Setting Up OSPF to Use an LSP .....	14-16
	Setting Up IS-IS to Use an LSP .....	14-17
	Configuring Cross Connects .....	14-18
	Setting up Service Class Support for Static LSPs .....	14-19
	Creating Diff-Serv Map Classes .....	14-20
	Configuring Diff-Serv for Cross Connects .....	14-20
	RSVP-signaled Label-switched Path Configuration .....	14-21
	Setting up Transit LSRs .....	14-22
	Setting up Signaled Paths at the Ingress LSR .....	14-23
	Verifying Path Configuration .....	14-25
	Changing the Configuration of an Explicit Path .....	14-26
	Setting up an IGP to Use the LSP .....	14-26
	Setting up Service Class Support for Signaled LSPs .....	14-26
	MPLS Commands to Verify Configuration .....	14-27
	Ping and Traceroute over MPLS Paths .....	14-28
	Ping .....	14-28
	Traceroute .....	14-29
	Interoperability Issues .....	14-29
Chapter 15	Internet Protocol (IP) Multicast Configuration	
	Key Features .....	15-1
	Technology Concepts .....	15-2
	RFCs and Standards .....	15-2
	Protocol Independent Multicast (PIM) .....	15-2

PIM Sparse Mode (PIM-SM) .....	15-3
Internet Group Management Protocol (IGMP) .....	15-4
Multicast Source Discovery Protocol (MSDP) .....	15-4
Designated Routers (DRs).....	15-5
Bootstrap Routers (BSRs) .....	15-5
Rendezvous Points (RPs) .....	15-5
Access Lists.....	15-5
Packet Format .....	15-6
Message Types .....	15-6
Internet Protocol Multicast Configuration.....	15-7
Basic PIM Configuration Tasks .....	15-9
Enabling PIM .....	15-10
Configuring a Bootstrap Router (BSR) Candidate.....	15-12
Configuring a Bootstrap Router (BSR) Border .....	15-12
Configuring a Rendezvous Point (RP) Candidate.....	15-12
Basic MSDP Configuration Tasks .....	15-13
Configuring MSDP Peers .....	15-13
Configuring Default MSDP Peers .....	15-14
Configuring MSDP Mesh Groups.....	15-14
Shutting Down MSDP Peers .....	15-15
Configuring Source-Active (SA) Messages.....	15-15
Configuring SA Caching.....	15-15
Configuring SA Requests .....	15-16
Configuring SA Message Distribution .....	15-16
Filtering SA Request Messages.....	15-16
Configuring Outgoing SA Message Filtering.....	15-17
Configuring Incoming SA Message Filtering.....	15-17
Configuring Time-to-Live (TTL) Thresholds .....	15-17
Optional IGMP Configuration Tasks .....	15-18
Configuring the IGMP Last Member Query Interval.....	15-18
Configuring IGMP Host Query Message Intervals.....	15-18
Configuring the IGMP Version .....	15-19
Configuring the Maximum Query Response Time.....	15-19
Verifying PIM Configuration.....	15-19

## Chapter 16

### OSPF Configuration

Key Features .....	16-1
Technology Concepts .....	16-2
RFCs and Standards.....	16-2
Network Topology .....	16-2
Autonomous Systems .....	16-3
Areas .....	16-3
Neighbors and Adjacencies .....	16-5
Designated Router (DR) and Backup Designated Router (BDR) .....	16-5
Autonomous Systems Boundary Router (ASBR) .....	16-6
Hello Protocol .....	16-6
Link-State Protocol.....	16-7
Link State Advertisements and Link State Database.....	16-7
OSPF Network Types .....	16-8
Interface Cost .....	16-9
Authentication .....	16-9
OSPF Packets Format .....	16-9

Common OSPF Header Format .....	16-10
OSPF Hello Packet .....	16-11
OSPF LSA Header .....	16-12
OSPF Configuration .....	16-13
Configuration Overview .....	16-15
Loading and Enabling OSPF.....	16-16
Configuring Authentication.....	16-17
Configuring Virtual Links.....	16-18
Creating a Stub Area.....	16-19
Using Summary Addresses.....	16-19
Route Redistribution.....	16-20
Verifying OSPF Configuration .....	16-21
 Chapter 17	
IS-IS Configuration	
Key Features .....	17-1
Technology Concepts .....	17-2
RFCs and Standards.....	17-2
Defining an Intermediate System.....	17-2
Routing Levels.....	17-3
Level 1 Routing.....	17-3
Level 2 Routing.....	17-3
Level 1-2 Routing.....	17-4
Network Topology .....	17-4
Areas .....	17-4
Domains .....	17-4
Routing Domain Types .....	17-5
IP-Only Routing Domains.....	17-5
OSI-Only Routing Domains .....	17-5
Dual Routing Domains.....	17-5
Understanding a Link-State Protocol.....	17-6
Debugging and Logging Facilities .....	17-6
IS-IS Packet Types and Formats .....	17-6
IS-IS Packet Types .....	17-6
IS-IS Packet Formats.....	17-12
IS-IS Address Structure .....	17-13
IS-IS Adjacency Establishment and Maintenance .....	17-14
Routing Processes.....	17-14
Link-State Database Maintenance .....	17-15
Avoiding Overload Problems .....	17-15
Route Selection Process.....	17-15
Influencing Interface Cost.....	17-16
Default Routes .....	17-17
Authentication.....	17-17
Router-based Authentication.....	17-17
Interface-based Authentication.....	17-17
Designated Routers/Pseudonodes.....	17-18
IS-IS Configuration .....	17-18
Basic IS-IS Configuration Tasks .....	17-20
Loading and Enabling IS-IS.....	17-21
Enabling IS-IS on the Interface .....	17-22
Configuring the Routing and Circuit Levels.....	17-23
Configuring Authentication .....	17-24

Configuring LSP-based Passwords .....	17-24
Configuring Interface-level Passwords.....	17-25
Manipulating the Routing Decision Process .....	17-26
Changing the Default Metric.....	17-26
Route Redistribution.....	17-27
Using Summary Addresses.....	17-27
Blocking Adjacency Formation .....	17-27
Verifying IS-IS Configuration .....	17-28

## Chapter 18 BGP Configuration

Key Features .....	18-1
Technology Concepts .....	18-2
Standards and RFCs.....	18-2
Autonomous Systems.....	18-2
External BGP.....	18-3
Internal BGP .....	18-3
BGP Route Decision and Installation Process .....	18-4
Update Messages .....	18-4
Neighbor Statements.....	18-5
Path Selection .....	18-5
Path Attributes .....	18-7
Debugging and Logging Facilities .....	18-8
BGP Packet Format.....	18-9
Message Types.....	18-9
Open Message .....	18-9
Update Message .....	18-10
Keepalive Message .....	18-12
Notification Message .....	18-12
Controlling Routing Information Flow .....	18-13
Peer Groups.....	18-13
Route Reflectors.....	18-13
Confederation .....	18-14
Aggregation.....	18-15
Redistribution.....	18-15
Route Filtering .....	18-15
BGP Configuration.....	18-16
Basic BGP Configuration Tasks .....	18-20
Enabling BGP.....	18-21
Neighbor Statements .....	18-22
Controlling Route Advertisements .....	18-25
Configuring Which Networks to Announce .....	18-25
Using Distribute Lists.....	18-26
Using Access Lists.....	18-27
Using Filter Lists and AS-Path Access Lists .....	18-28
Using Filtering and Route Redistribution for Flow Control .....	18-28
Using Route Maps .....	18-29
Using Peer Groups for Route Redistribution.....	18-29
Managing Network Configuration .....	18-30
Configuring Route Reflectors.....	18-31
Configuring Route Aggregation .....	18-31
Using Attributes.....	18-32
Verifying BGP Configuration .....	18-33

---

	Interoperability Issues .....	18-34
Chapter 19	Route Filter Configuration	
	Technology Concepts .....	19-1
	Understanding the Filtering Process .....	19-2
	Filtering Inbound and Outbound Updates .....	19-2
	Filtering on IP Address .....	19-2
	Filtering on AS Path .....	19-3
	Using Regular Expressions .....	19-4
	Matching Criteria and Setting Attributes .....	19-7
	Using Match Commands .....	19-7
	Using Set Commands .....	19-7
	Understanding Route Maps .....	19-7
	Using Lists to Identify Match Criteria .....	19-9
	Matching Using IP Address Access Lists .....	19-9
	Matching Using AS-Path Access Lists .....	19-9
	Matching Using Distribute Lists .....	19-10
	Matching Using Filter Lists .....	19-10
	Understanding Order of Precedence .....	19-10
	Peer Groups and Filters .....	19-11
	Overriding Settings .....	19-11
	Other Protocols .....	19-11
	Route Filtering Configuration .....	19-12
	Basic Route Filtering Configuration Tasks .....	19-14
	Filtering on IP Address Using Distribute Lists .....	19-15
	Filtering on AS Path Using Filter Lists .....	19-16
	Changing Attributes to Accomplish Path Editing .....	19-17
	Overriding Peer Group Settings .....	19-18
	Redistributing OSPF into BGP with Route Maps .....	19-18
	Verifying Route Filtering Configuration .....	19-19
	Interoperability Issues .....	19-20
Appendix A	Acronyms	
	Acronyms .....	A-1
	Customer Support	

## List of Tables

Table 2-1.	Terminal Settings .....	2-2
Table 2-2.	Modem Settings .....	2-2
Table 2-3.	CLI Prompt Levels .....	2-5
Table 3-1.	Software Module Names.....	3-2
Table 3-2.	PCMCIA0 Directory Contents.....	3-3
Table 3-3.	Software Management Command Usage.....	3-4
Table 3-4.	File and Directory Command Usage .....	3-9
Table 4-1.	Username and Password Command Usage .....	4-5
Table 4-2.	Telnet Command Usage .....	4-6
Table 4-3.	RADIUS Configuration and Validation Commands.....	4-9
Table 4-4.	RADIUS Commands to Verify Configuration.....	4-10
Table 5-1.	Message Severity Levels.....	5-2
Table 5-2.	Default Severity Levels for Logging Information .....	5-3
Table 5-3.	Default Severity Levels for Message Display .....	5-3
Table 5-4.	Log Command Usage .....	5-4
Table 5-5.	Debug Command Usage.....	5-10
Table 5-6.	Show Commands for Debugging .....	5-11
Table 6-1.	SNMP Command Usage .....	6-6
Table 6-2.	SNMP Commands for Verification and Monitoring.....	6-8
Table 7-1.	Module Description.....	7-1
Table 7-2.	Slot Locations and Card Orientations .....	7-2
Table 7-3.	Interface Speed.....	7-3
Table 7-4.	Ethernet0 and inet on ethernet addressing.....	7-6
Table 7-5.	IOP to IOA Relationship.....	7-7
Table 7-6.	Line Cards Signal Levels.....	7-7
Table 7-7.	Card-related Command Usage .....	7-9
Table 7-8.	Interface Name Entry Format .....	7-12
Table 7-9.	Interface Configuration Command Usage.....	7-13
Table 7-10.	Card and Interface Commands for Verifying Configuration.....	7-18
Table 8-1.	Frame Relay Command Usage .....	8-10
Table 8-2.	Frame Relay Commands for Monitoring Connections .....	8-19
Table 8-3.	Implementation Differences.....	8-20
Table 9-1.	ATM Cell Header Fields.....	9-7
Table 9-2.	ATM Command Usage .....	9-8
Table 9-3.	ATM Commands for Verifying Configuration.....	9-25
Table 10-1.	Supported Network Control Protocols.....	10-2
Table 10-2.	PPP Commands for Verifying Configuration.....	10-5
Table 11-1.	Reserved and Available IP Addresses.....	11-2
Table 11-2.	Description of IP Header Fields .....	11-7
Table 11-3.	IP Commands Usage.....	11-9
Table 11-4.	IP Commands for Verifying Configuration .....	11-15
Table 12-1.	UDP Port Descriptors.....	12-4
Table 12-2.	TCP Port Descriptors .....	12-5
Table 12-3.	ICMP Message Types.....	12-7
Table 12-4.	IP Header.....	12-10
Table 12-5.	TCP Header .....	12-11
Table 12-6.	UDP Header.....	12-12
Table 12-7.	Access List Command Usage .....	12-12
Table 12-8.	Access Lists Commands for Verifying Configuration .....	12-16

Table 13-1.	Default Priority Queues.....	13-4
Table 13-2.	Maximum Queue Size for Card Types .....	13-7
Table 13-3.	QoS Command Usage .....	13-11
Table 13-4.	QoS Commands to Verify Configuration and Monitor Traffic...	13-17
Table 14-1.	MPLS Command Usage.....	14-12
Table 14-2.	MPLS Commands to Verify Configuration.....	14-27
Table 15-1.	PIM Header Field Descriptions.....	15-6
Table 15-2.	Internet Protocol Multicast Commands Usage.....	15-7
Table 15-3.	PIM Commands for Verifying Configuration .....	15-19
Table 16-1.	Link State Advertisements .....	16-8
Table 16-2.	Common OSPF Header Fields .....	16-10
Table 16-3.	OSPF Hello Packet Fields.....	16-11
Table 16-4.	Common LSA Header .....	16-12
Table 16-5.	OSPF Command Usage .....	16-14
Table 16-6.	OSPF Commands for Verifying Configuration.....	16-21
Table 17-1.	RFCs Implementing Standardized IS-IS .....	17-2
Table 17-2.	Summary of Routing Levels.....	17-3
Table 17-3.	Hello PDU Types.....	17-7
Table 17-4.	Hello Packet Additional Fixed Header Information .....	17-7
Table 17-5.	Variable-length Hello Packet Header Fields .....	17-8
Table 17-6.	LSP Packet Additional Fixed Header Information.....	17-9
Table 17-7.	LSP Difference Variable-length Header Fields.....	17-10
Table 17-8.	SNP Additional Fixed Header Information .....	17-11
Table 17-9.	Description of Fields in the IS-IS Fixed Header .....	17-12
Table 17-10.	NET Address Components.....	17-14
Table 17-11.	IS-IS Routing Processes .....	17-14
Table 17-12.	IS-IS Command Usage .....	17-18
Table 17-13.	IS-IS Commands for Verifying Configuration.....	17-28
Table 18-1.	RFCs Implementing Standardized BGP.....	18-2
Table 18-2.	BGP Origin Attributes .....	18-7
Table 18-3.	BGP Community Attributes .....	18-8
Table 18-4.	Description of Open Message Fields.....	18-10
Table 18-5.	Description of Update Message Fields.....	18-11
Table 18-6.	Description of Notification Message Fields.....	18-12
Table 18-7.	Route Reflector Behavior .....	18-13
Table 18-8.	BGP Command Usage .....	18-16
Table 18-9.	BGP Neighbor Statements.....	18-23
Table 18-10.	BGP Commands for Verifying Configuration.....	18-33
Table 19-1.	Regular Expression Special Characters .....	19-4
Table 19-2.	Examples of Regular Expression Special Characters.....	19-5
Table 19-3.	Route Filtering Command Usage .....	19-12
Table 19-4.	Route Filtering Commands for Verifying Configuration.....	19-20

---

## List of Figures

Figure 1-1.	System Hardware Overview.....	1-2
Figure 1-2.	IOA-IOP Architecture .....	1-2
Figure 1-3.	Packet Transmission Through the System .....	1-3
Figure 1-4.	NX-IS Routing Modules .....	1-4
Figure 1-5.	Basic Configuration Example.....	1-6
Figure 2-1.	Command Prompt Hierarchy .....	2-4
Figure 3-1.	File and Directory Location on the PCMCIA Card.....	3-3
Figure 6-1.	SNMP Manager and Agent Model .....	6-2
Figure 6-2.	Object ID Tree .....	6-4
Figure 7-1.	Front View of Chassis.....	7-2
Figure 7-2.	Rear view of Chassis .....	7-3
Figure 8-1.	Data Link Connection Identifier .....	8-3
Figure 8-2.	Frame Relay Frame .....	8-8
Figure 8-3.	Frame Relay Frame Header.....	8-8
Figure 8-4.	LMI Frame Format .....	8-9
Figure 8-5.	Frame Relay Configuration Sample Network .....	8-12
Figure 9-1.	Fields in an ATM cell.....	9-6
Figure 9-2.	Example Allocating Numbers of VCs and VPs .....	9-12
Figure 9-3.	Point-to-Point PVC Configuration Examples .....	9-15
Figure 9-4.	Point-to-Multipoint PVC Configuration Example.....	9-21
Figure 9-5.	Map Group Configuration Example.....	9-24
Figure 10-1.	PPP Sample Network.....	10-3
Figure 11-1.	The IP Header Format .....	11-7
Figure 11-2.	IP Sample Network .....	11-11
Figure 12-1.	IP Header Fields Used with Access Lists .....	12-10
Figure 12-2.	TCP Header Fields Used with Access Lists.....	12-11
Figure 12-3.	UDP Header Fields Used with Access Lists .....	12-11
Figure 13-1.	Default Bandwidth Configuration.....	13-5
Figure 13-2.	Example: Traffic Scheduling for Oversubscribed Queues .....	13-9
Figure 13-3.	Example: Bandwidth Sharing from Undersubscribed Queues..	13-10
Figure 13-4.	Diff-Serv Services Field .....	13-10
Figure 13-5.	QoS Configuration Example .....	13-13
Figure 14-1.	MPLS Domain .....	14-4
Figure 14-2.	Label Values on an LSP .....	14-4
Figure 14-3.	Label Values on Adjacent LSRs .....	14-6
Figure 14-4.	Label Switching at a Cross Connect .....	14-7
Figure 14-5.	Label Merge.....	14-7
Figure 14-6.	MPLS Header Format for PPP Packets.....	14-11
Figure 14-7.	Example of a Static LSP Configuration .....	14-14
Figure 14-8.	Example RSVP-signaled LSP Configuration.....	14-21
Figure 14-9.	Ping over 2 LSPs.....	14-28
Figure 15-1.	PIM Sparse Mode with Shared Tree and Shortest-Path Tree .....	15-3
Figure 15-2.	A Simple PIM Network .....	15-10
Figure 16-1.	OSPF Autonomous System or Routing Domain .....	16-3
Figure 16-2.	OSPF Virtual Link .....	16-4
Figure 16-3.	Common OSPF Header .....	16-10
Figure 16-4.	OSPF Hello Packet.....	16-11
Figure 16-5.	Common LSA Header .....	16-12
Figure 16-6.	Sample OSPF Network.....	16-16



---

Figure 16-7.	OSPF Configuration Example with IS-IS Addition .....	16-20
Figure 17-1.	The IS-IS Network Hierarchy .....	17-5
Figure 17-2.	IS-IS Fixed Header Fields .....	17-12
Figure 17-3.	IS-IS Configuration Example .....	17-21
Figure 18-1.	Autonomous Systems in an Internetwork .....	18-3
Figure 18-2.	External vs. Internal BGP .....	18-4
Figure 18-3.	BGP Path Selection Process .....	18-6
Figure 18-4.	BGP Header Format .....	18-9
Figure 18-5.	BGP Open Message Fields .....	18-9
Figure 18-6.	BGP Update Message Fields .....	18-10
Figure 18-7.	BGP Notification Message Fields .....	18-12
Figure 18-8.	Route Reflection Eliminate the Need for Full Mesh in an AS ..	18-14
Figure 18-9.	An Example of a BGP Confederation, AS123123 .....	18-14
Figure 18-10.	BGP Network Example .....	18-21
Figure 18-11.	The Network Command Defines Advertised Networks .....	18-25
Figure 19-1.	AS Path of a Prefix through the Network .....	19-3
Figure 19-2.	BGP Network Example for Route Filtering .....	19-15
Figure 19-3.	OSPF Addition to BGP Example Network .....	19-19



# About this Guide

The *NX64000 Configuration Guide* describes how to configure an NX64000™ IP Core Router. The book provides basic information about the services and protocols the system supports, and about basic system configurations. It *does not* provide information about all possible configurations. See the *NX64000 Command Reference* for documentation of all the configuration commands available at the CLI.

## Intended Audience

This guide is intended for system administrators or network administrators who are knowledgeable about their networks, and about the function the NX64000 system in that network.

## Conventions

This manual uses the following conventions:

Convention	Indicates	Example
Courier regular	Screen output or syntax.	logging source-interface pos2/1
<i>Courier italic</i>	Variable; generic text for which you supply a value.	show ip interface [ <i>interface-name</i> ]
<b>Courier bold</b>	User input.	nx64000# <b>show ip ospf database</b>
<b>Sans serif bold</b>	Command names, options, and keywords in text.	By omitting the no-summary option...
Braces { }	Required argument; choose one.	clock-source {line   internal}
Brackets [ ]	Optional argument.	set-overload-bit [ <i>on-startup seconds</i> ]

---

Vertical bar	Separates required or optional arguments to select from ("or").	show ip bgp neighbors [ <i>ip-address</i>   <i>as-number</i> ] events [ <i>count</i> ]
<i>ver</i>	Variable version number in filenames.	<i>rver</i> .tar

This guide also uses the following conventions to call attention to important information.



Notes provide additional information or helpful suggestions that apply to the subject text.



Cautions notify the reader to proceed carefully to avoid possible equipment damage or data loss.



Warnings notify the reader to proceed carefully to avoid possible personal injury.

## Product Documentation

The following documentation supports the NX64000 IP Core Router:

- *NX64000 Installation Guide*
- *NX64000 Command Reference*
- *NX64000 Command Quick Reference*
- *NX64000 Troubleshooting Guide*
- *Release Notes for the NX64000 IP Core Router NX-IS Software*
- *NX64000 Element Management System (NX-MS) User's Guide*
- *Release Notes for the NX64000 Element Management System (NX-MS)*

# Configuration Overview

The the NX64000<sup>TM</sup> IP Core Router delivers the core requirements of a carrier-class multiservice device. The system provides a scalable architecture to support high-speed optical interfaces as well as a high-port density. A single chassis can support multiple interface speeds and types, letting you integrate the system with your current network, and supporting growth to meet network demands.

The NX-IS<sup>TM</sup> software provides system configuration and monitoring capabilities from a command line interface (CLI). A modular architecture lets individual routing and control modules operate as separate applications over a real-time operating system. NX-IS supports the major IP routing protocols.

The NX64000 Element Management System, NX-MS<sup>TM</sup>, provides system monitoring through a graphical user interface. For information about this program, see the *NX64000 Element Management System (NX-MS) User Guide*.

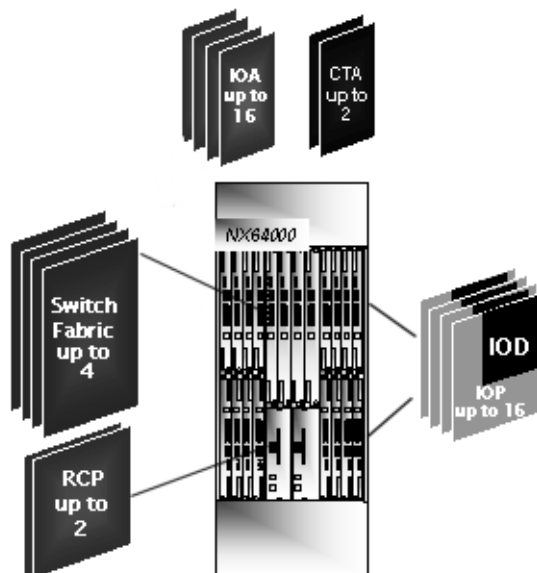
## System Hardware

The NX64000 IP Core Router provides high port density, with interface speeds at OC-3, OC-12, OC-48, and OC-192, as well as gigabit Ethernet and DS-3. The system supports up to 16 line cards.

Data travels through the system over high-speed switch fabric modules. A system route control processor (RCP) controls the system and manages traffic routing. The system uses a SONET/SDH Timing/Alarm module to manage SONET/SDH status and timing.

Communication between the modules in the system is over management buses. Internal IP addresses assigned to the different modules allow them to communicate with each other.

Figure 1-1 shows the modules supported on the NX64000 switch/route:



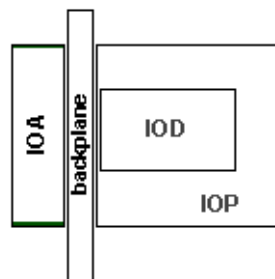
**Figure 1-1. System Hardware Overview**

For complete information about hardware features, including information about the power and cooling features, see the *NX64000 Installation Guide*.

## Line Card–Processor Engine Architecture

Each line card, also called an input-output adapter (IOA) module, communicates over a backplane with an associated processor engine, also called an input-output processor (IOP). The IOA interfaces with the optical media. The IOP forwards packets through the system and manages the module routing tables, control functions, and signaling.

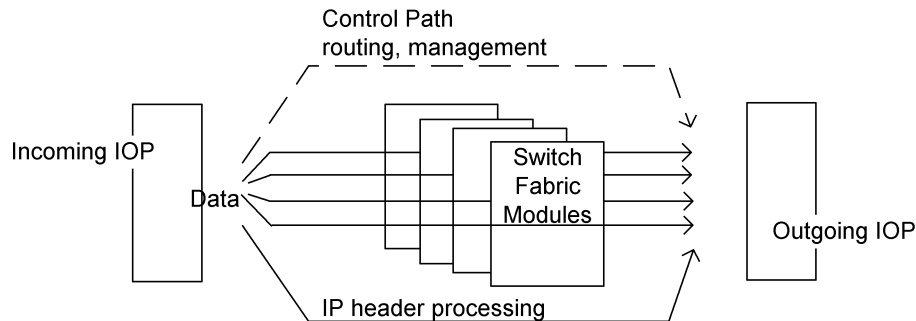
The following illustration shows the relationship of the IOA and IOP. The figure also shows an IOD, an input-output daughter card, which determines the board type. For example, the IOD determines whether an OC-3/OC-12 IOP functions as an OC-3 or an OC-12 IOP.



**Figure 1-2. IOA-IOP Architecture**

## Packet Forwarding

The system has separate data and control paths to increase forwarding speed through the system. In general, the system transmits data directly to and through the switch fabric. IP header processing occurs over the control path.



**Figure 1-3. Packet Transmission Through the System**

## Redundancy Features

A NX64000 switch/router configured as a duplex system provides redundancy for system components. A duplex system contains two of each of the following modules:

- RCP
 

A primary RCP acts as the RCP for the system. A secondary RCP takes over should the primary RCP fail. When a secondary RCP takes over, it reboots the entire system including the line cards.
- Switch fabric modules
 

When an active standby module is installed on the system, it provides backup for an individual switch fabric.
- SONET timing and alarm modules
- Power distribution units (PDUs)

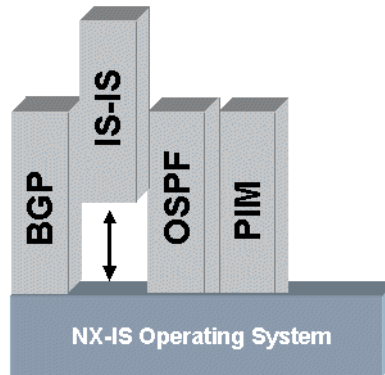
## Agency Specification Compliance and Operating Environment

To read about regulatory information, compliance with agency specifications, and information about the operating environment, see the *NX64000 Installation Guide*.

## System Software

The NX-IS software lets you configure and display information about the NX64000 switch/router from a command line interface (CLI). The CLI runs over SNMP and relies on SNMP messaging within the system.

A single-image file contains the system and protocol modules for the switch/router. This implementation makes it easier to manage system files. Each routing protocol is an independent module within the image file as illustrated by the following figure:



**Figure 1-4. NX-IS Routing Modules**

A separate system configuration file stores the software configuration information. The system reads the contents of the file during system startup to configure the switch/router. The system saves configuration changes to the configuration file only if you explicitly save them. This allows you to test a configuration with the changes residing only in system memory until you decide to make the changes permanent by saving them.

Software monitoring capabilities provide real-time system statistics. SNMPv2 trap messages and system log and debug messages provide information about system status. Integration with SNMP network management stations and with network logging facilities, such as a Syslog server, help you monitor system performance. CLI `show` commands give you direct access to information about system and protocol status.

## Network Protocols and Services

The NX-IS software provides support for the following network protocols and services:

- Frame Relay—A network protocol that transfers data over permanent virtual circuits within a wide area network  
NX-IS provides support for both Frame Relay routing and Data Link Connector Identifier (DLCI) switching.
- Point-to-Point protocol (PPP)—A connection-oriented protocol that establishes a link between two systems on the same subnet
- Internet Protocol (IP)—The network layer protocol that supports communication between different networks
- Multi-protocol label switching (MPLS)—A connection-oriented framework for IP that lets you send traffic through configured paths
- IP Multicast—Protocols that support transmission of an IP datagram to a host group or a set of hosts identified by a single, class D, or IP destination address  
This implementation of multicasting relies on Protocol Independent Multicast (PIM), Multicast Source Discovery Protocol (MSDP), and Internet Group Management Protocol (IGMP).



- Asynchronous Transfer Mode (ATM)—A network protocol for switching and multiplexing data  
The system terminates ATM virtual circuits, and can transmit traffic over system-configured virtual circuits. The NX64000 switch/router *does not switch* ATM traffic.

## Routing Protocols

The software supports the standard routing protocols:

- Border Gateway Protocol (BGP), version 4—An exterior and interior gateway routing protocol
- Intermediate System-to-Intermediate System (IS-IS) — An interior gateway routing protocol
- Open Shortest Path First (OSPF), version 2—An interior gateway routing protocol

IS-IS and OSPF both integrate with MPLS to support MPLS label-switched paths (LSPs) as part of the routing configuration.

Route filtering and system access lists let you control system and route access to the system.

## Service Classes

NX-IS works with the system hardware to support quality of service (QoS) for traffic transmission. QoS lets you predictably manage different priorities of traffic as it passes through the system.

MPLS integration with the Diff-Serv framework provides support for service classes over MPLS label-switched paths.

## Preparing to Configure the System

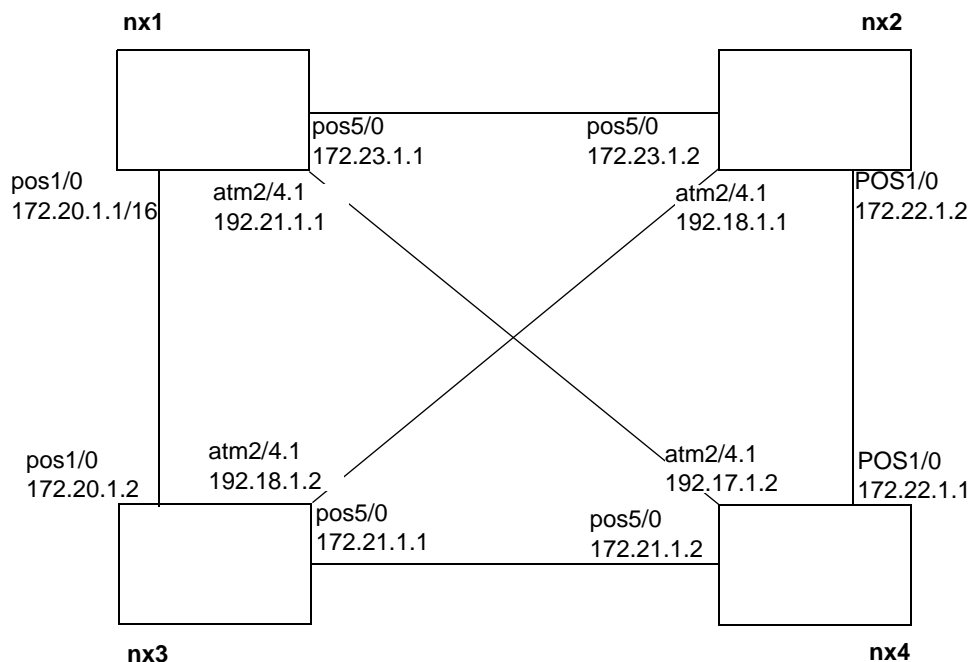
Before proceeding with system configuration, you should determine how the configuration on the NX64000 switch/router fits into your network's topology:

- What type of authentication (password protection or RADIUS authentication) do you want to use to protect management functions on the system?
- Do you want the system to send SNMP trap messages to an SNMP management station? If so, what is the SNMP configuration on your network?
- How does your network manage log messages from network devices?
- Which types of links does the network require (Frame Relay, PPP, ATM)?
- What IP addressing scheme do you want to use for system interfaces?
- Do you want to establish quality of service for packet delivery?
- Which routing protocol(s) does the network use (BGP, OSPF, IS-IS)?
- Do you want MPLS LSPs to augment your routing strategy?
- How do you want to control system and network access?
- Will the network require support for multicasting?

## Configuration Examples

Most configuration examples in this guide use a four-router topology to illustrate the various configurations determined by a router's functions on the network. While hypothetical, the four-router example provides a consistent presentation for system configuration. The examples use interfaces and IP addresses similar to those in the illustration.

The following illustration shows the sample setup of four NX64000 switch/routers, with the host names nx1, nx2, nx3, and nx4:



**Figure 1-5. Basic Configuration Example**

## Configuration Tasks

During initial system configuration the following should have been set:

- System host name
- IP address of the management Ethernet port

The following table lists the steps you follow to configure the system, and shows which chapter to see for configuration information:

For this configuration step	See this chapter
<b>1.</b> Set up authentication for system access.	<b>Chapter 4, "System Security and Access"</b>
<b>2.</b> Set up SNMP access.	<b>Chapter 6, "Setting up System Monitoring"</b>
<b>3.</b> Configure logging.	<b>Chapter 5, "Working with the Log and Debug Utilities"</b>
<b>4.</b> Configure system interfaces. The types of cards installed on the system determine the types of interfaces you can configure.	<b>Chapter 7, "Cards and Interfaces"</b>
<b>5.</b> Set up interface encapsulation or protocol.	<b>Chapter 10, "Point-to-Point Protocol Configuration"</b> <b>Chapter 8, "Frame Relay Configuration"</b> <b>Chapter 9, "Asynchronous Transfer Mode (ATM) Configuration"</b>
<b>6.</b> Set up access.	<b>Chapter 12, "Access List Configuration"</b>
<b>7.</b> Configure quality of service to manage prioritized traffic.	<b>Chapter 13, "Quality of Service Configuration for Traffic Management"</b>
<b>8.</b> Add MPLS tunnel interfaces and configuration.	<b>Chapter 14, "Multi-protocol Label Switching (MPLS) Configuration"</b>
<b>9.</b> Set up protocol independent multicast.	<b>Chapter 15, "Internet Protocol (IP) Multicast Configuration"</b>
<b>10.</b> Configure routing protocols.	<b>Chapter 16, "OSPF Configuration"</b> <b>Chapter 17, "IS-IS Configuration"</b> <b>Chapter 18, "BGP Configuration"</b> <b>Chapter 19, "Route Filter Configuration"</b>

## Updates to System Configuration

You can update the system configuration as needed. Run associated **show** commands to display information about the current configuration before making changes.



## Using the Command Line Interface (CLI)

The CLI gives you access to configuration commands for all system services and protocols. It also provides commands to verify configuration and display information about the system. You can access the CLI locally or remotely.

### Connecting to the CLI

The route control processor (RCP) provides connections to the CLI through:

- The management Ethernet port
- The console 1 port

#### Ethernet Connection

The interface for the management Ethernet port allows access to the RCP through a TCP/IP connection. The most common way to establish a remote connection is by setting up a Telnet session through the management Ethernet port to emulate a CLI session. For information about managing Telnet access to the system, see [Chapter 4, “System Security and Access.”](#)

At system installation, an IP address should be assigned to the Ethernet management port. For information about how to change or set the IP address of the management Ethernet port, see [Chapter 11, “Internet Protocol Configuration.”](#)

#### Console Connection

You can setup a connection to the console 1 port on the RCP for either a modem or a terminal. When working at the site, you can set up terminal access, then change to modem access when the system is unattended. Terminal and modem access both require the RJ-45/25-pin adaptor, making it easy to switch from one line to the other.

## Setting up a Terminal Connection

Set the following for the terminal connection:

**Table 2-1. Terminal Settings**

Setting	Value
Port speed	9600 baud
Data protocol	Standard EC
Compression	Enabled
Flow Control	None

## Setting up a Modem Connection

Set the following on the modem to ensure proper operation with the system:

**Table 2-2. Modem Settings**

Setting	Value
Data terminal ready	Use override
Verbal result	On
Results display	On
Echo commands	On
Automatic answer	On
Carrier detect	Normal
Factory defaults	On
Dumb (or terminal) mode	On

The settings for data terminal ready and carrier detect are required to provide access when the system is booting. Otherwise, if the modem detects that the system is not operational, it disconnects a current modem session, or does not form a new one.

- In some cases, you may not see output to the screen. This may indicate that flow control is not enabled. Entering Ctrl-Q should enable flow control. When flow control is enabled, the screen displays CLI output.

## Command Overview

The commands available from the CLI have the following characteristics:

- Command and argument names, with the exception of passwords, are case-insensitive.
- When entering a command name, it may be abbreviated to the first unique string. For example, you can enter `int` for the `interface` command rather than typing the full word.
- In most cases, the `no` form of a command (the command name preceded by the word `no`) reverses the action carried out by a command.
- `show` commands display information about system configuration and status.

For information about editing tools available at the command line, see *NX64000 Command Reference*.

## Privilege Levels

The privilege level configured for a user, read-only or read-write, determines which commands a user can run. Read-only access lets a user get information about the system by running `show` commands. Read-write access lets users make configuration changes. The system prompt indicates the user access level:

- Read-Only—`nx>`
- Read-Write—`nx#`

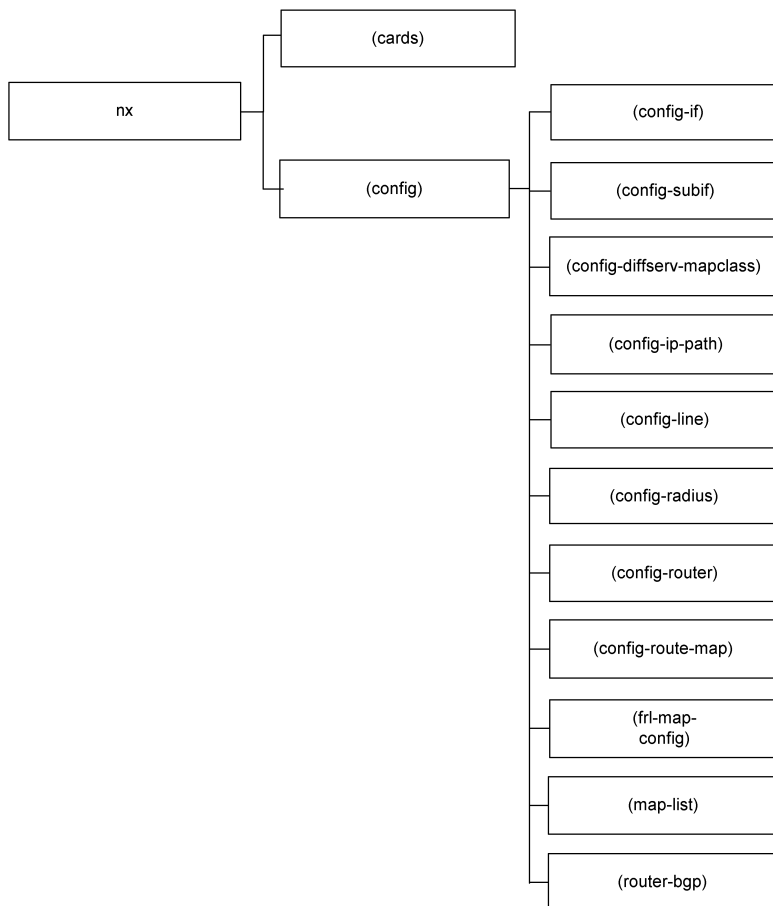
► This book uses the `nx` prompt, as the default system prompt. The actual system prompt is set by the `hostname` command.

For information about setting the privilege level for users, see [Chapter 4, “System Security and Access.”](#)

## Command Prompt Levels

The CLI groups commands under a series of prompt levels. This organization provides access to commands used to configure a service or protocol at a specified prompt. You can change from a lower prompt level, to the next level by entering the `exit` command. To go to the top-level prompt from any level within the CLI, type `end`.

The following illustration provides an overview of the CLI prompt levels:



**Figure 2-1. Command Prompt Hierarchy**



The following table briefly describes each of the system prompt levels:

**Table 2-3. CLI Prompt Levels**

This CLI prompt level	Provides access to these commands
<code>nx#</code>	All system commands
<code>nx(cards)#</code>	Card-specific commands
<code>nx(config)#</code>	Global system configuration commands and other configuration prompt levels
<code>nx(config-if)#</code>	Interface configuration commands
<code>nx(config-subif)#</code>	Subinterface configuration commands
<code>nx(config-line)#</code>	Commands to manage console and Telnet sessions
<code>nx(map-list)#</code>	Commands to configure mapping between an upper-level protocols such as IP and lower-level asynchronous transfer mode virtual circuits
<code>nx(config-router)#</code>	Intermediate System-to-Intermediate System and Open Shortest Path First routing commands
<code>nx(config-route-map)#</code>	A route-map that you can configure and then assign to route filtering commands
<code>nx(router-bgp)#</code>	Border Gateway Protocol routing commands
<code>nx(config-ip-path)#</code>	Path configuration commands for Multiprotocol Label Switching (MPLS) label switched paths
<code>nx(config-diffserv-mapclass)#</code>	Classes of service commands for MPLS
<code>nx(frl-map-config)#</code>	Quality of Service (QoS) values for a Frame Relay permanent virtual circuit
<code>nx(config-radius)#</code>	RADIUS configuration commands

## Using Context-Sensitive Help

The CLI provides context-sensitive Help. The system responds to a Help query based on the cursor's position at the command line. Use the online Help system to list available commands and command arguments and to verify syntax.

A question mark requests information from the Help system. The system provides the following types of help:

- Command description

Enter a full or partial command name followed immediately by a question mark (?) (that is, no space between the command and the ?). Help returns a description of the command or argument. For example:

```
nx(config)# router?
Global Configuration Commands:
router                  enter router configuration modes
```

- **Command options**

Enter the full or partial command name followed by a space and then a question mark (?). Help lists the options or commands that match the string. For example:

```
nx(config)# router ?
enter router configuration mode:
bgp                Configure BGP
isis               Configure IS-IS on router
ospf               OSPF Router
```

- **Command listing**

Enter a question mark at the command prompt. Help lists all commands available from that prompt. If you enter a command that the system does not recognize, the system returns an error message and lists the commands available at the prompt. For example if you enter a ? at the nx(config)# prompt:

```
nx(config)# ?
frame-relay        Frame relay global commands
hostname           hostname -- This system's network name
interface          Select an interface to configure
ip                 IP Interface Global Command Mode
line               Configure terminal lines
load               Load hot-swappable module
logging            log configuration commands
map-class          Enter map-class configuration
map-list           Enter atm map-list
mpls               MPLS Configuration Commands
no                 Negate a command or sets its defaults
priority-list      Establishing queueing priority commands
radius             Radius configuration table
remove             Remove hot-swappable module from file system
route-map          Create route-map or enter route-map command
mode
router             enter router configuration modes
service
snmp-server        snmp-server
start              Activate hot-swappable module
stop               Cold stop hot-swappable module
unload             Unload hot-swappable module
username           username <name>
workdir            Configure Default Working Directory
```

## Working with System Files

System management requires manipulating files to manage system software and to store configuration information. The files you work with are the:

- Single image file—This file provides the operating system and all software modules.
- Routing modules—These files are dynamically loadable modules within the single image file.
- Configuration file—This file supplies the configuration information the system uses at boot time to set system parameters.

The PCMCIA card in the route control processor (RCP) houses the system files.

- After you install new software, you should make backup copies of the image file and the startup configuration files on your network. For information about booting the system using files stored on the network, see the *NX64000 Troubleshooting Guide*.

### Single Image File

The single image file, *rver.tar* where *ver* indicates the software version, contains all the system software, including operational code for hardware and software modules, and the modules for the routing protocols. Reloading the image files boots the system.

## Modules for Routing Protocols

The system provides a separate software module for each routing protocol. You must load a routing protocol module to activate it on the system. The following table lists the modules for the routing protocols:

**Table 3-1. Software Module Names**

Routing Protocol	Module Name
Border Gateway Protocol (BGP)	bgp
Intermediate System-to-Intermediate System (IS-IS)	isi
Open Shortest Path First (OSPF)	spf
Protocol-Independent Multicast (PIM)	pim

## Backup Copy of an Image File

You can set up a backup image file for the system to use should it encounter a problem booting from the current image file. The system boots from a backup image file when:

- The single image file *is not* in the current directory.
- The system fails to boot from the image file in the current directory.

► The backup utility is supported in release 1.7 and higher. For systems previously running the release 1.6 software, the upgrade procedure must be completed to use the backup utility. To perform the upgrade, see the *Release Notes for the NX64000 IP Core Router, NX-IS Software Release 1.7*.

## System Configuration File

The system configuration file, called the startup configuration file, stores configuration settings used when the system boots. When you make configuration changes, the system stores the information in memory. You must explicitly save this configuration information, called the running configuration, to the configuration file to preserve the settings. The default filename of the startup configuration file is `startup.cfg`.

You can compare configuration information stored in the startup configuration file with the configuration in memory by running the `show running-config` command to display the parameter and configuration settings in memory and the `show startup-config` command to display settings stored in the configuration file.

## File and Directory Organization

The following illustration shows the location of directories and of the `startup.cfg` and `rver.tar` files on the PCMCIA card. It also shows the location of the `startup.dat` file, a file created by the system. This file stores the location of the *current* and *backup* directories.

```
pcmcia0 (directory)
  startup.cfg
  releases (directory)
    startup.dat
    current (directory)
      r170.tar
      backup (directory)
  pcs (directory)
```

**Figure 3-1. File and Directory Location on the PCMCIA Card**

The `pcmcia0` directory is the top-level directory on the PCMCIA card. The following tables describes the directories within the `pcmcia0` directory:

**Table 3-2. PCMCIA0 Directory Contents**

Directory	Contents
releases	The current directory and the backup directory. These sub-directories contain versions of the single image files used to boot the system.
current	The version of the single image file used to boot the system.
backup	The version of the single image file used to boot the system, should the system fail to boot from the version of the image file in the current directory.
pcs	Working files for the system. Should the content of this directory be deleted, the system recreates the files needed.

The PCMCIA card ships with the same version of the `rver.tar` file in both the current and backup directories. Having the same version of the `rver.tar` file in two directories provides a backup if the version in the current directory becomes corrupted or fails to boot.

When upgrading the system software, you can keep an older version of the software in a backup directory, and the newer version in the current directory. Should there be a problem booting from the updated software, the system automatically boots from the older version in the backup directory.

- The version of the image file in the backup directory must be version 1.7 or greater. The system will not boot earlier versions of the software from this directory.

## Managing System Software and Files

This section describes the frequently used software management commands. The *NX64000 Command Reference* manual describes all commands referenced in this chapter.

The following table lists the software management commands:

**Table 3-3. Software Management Command Usage**

Command	Manage image files	Manage configuration files	Manage software modules	Set the file system	Verification
backup	✓				
boot backup	✓				
boot config		✓			
boot system	✓				
configure file		✓			
copy running-config startup-config		✓			
copy sync	✓	✓	✓		
erase startup-config		✓			
load			✓		
mount				✓	
reload	✓				
show boot					✓
show flash					✓
show modules			✓		✓
show running-config		✓			✓
show startup-config		✓			✓
start			✓		

**Table 3-3. Software Management Command Usage**

Command	Manage image files	Manage configuration files	Manage software modules	Set the file system	Verification
stop			✓		
unload			✓		
unmount				✓	

## Rebooting the System

Reloading the system image file reboots the system. It uses the “current” image file, typically stored in the `/pcmcia0/releases/current` directory unless otherwise specified. At boot, the startup configuration file sets the system configuration.

When you issue the `reload` command to reboot the router, the system prompts you to save changes to the startup configuration file if there are configuration changes in memory. You can either save the file (**y**), discard the changes (**n**), cancel the reload request (**q**), or use a new filename rather than accepting the default (**f**).

The following boots the system from the image file stored in the “current” directory.

```
nx# reload
Save configuration information in /pcmcia0/startup.cfg? <y|n|q|f>y
Confirm reload? <yes|no>y
```

For information about booting from a backup image of the software, see [“Booting from a Backup Version of the Single Image File” on page 3-6](#).

## Working with Software Image Files

This section describes basic procedures for managing image files on the system.

### Changing the Location of an Image File

You can change the setting for the location of the system image by using the `boot system` command. If you want to display the directory where the `rver.tar` file resides, use the `show boot` command.

The following example sets the location of image file to `/pcmcia0/releases/current2`:

```
nx(config)# boot system /pcmcia0/releases/current2
```

## Working with a Backup Image of the Software

The PCMCIA card supplied with a software release contains the same version of the image file in the current and backup directories.

### Changing the Path to the Backup Directory

You can change the setting for the location of your backup directory by using the `boot backup` command. Before setting the location of a backup directory, the specified directory should contain a backup version of the software.

The following example shows the initial location of the backup directory as `/pcmcia0/releases/backup`, then changes the boot backup directory to `/pcmcia0/releases/newbackup`:

```
nx# show boot
BOOT SYSTEM variable = /pcmcia0/releases/current/
BOOT BACKUP variable = /pcmcia0/releases/backup/
CONFIG_FILE variable = /pcmcia0/startup.cfg
nx# configure terminal
nx(config)# boot backup /pcmcia0/releases/newbackup
```

### Moving an Image File into a Backup Directory

When you move an image file from the current directory to the backup directory, the system *removes the file* in the current directory. The following example moves the `r170.tar` file into the backup directory and deletes the `r170.tar` file from the current directory:

```
nx# backup
Current tar file R170.TAR will be removed after copying to backup.
Confirm backup? <yes|no>y
Copied R170.TAR to backup directory.
Removed /pcmcia0/releases/current/R170.TAR.
```

### Booting from a Backup Version of the Single Image File

When rebooting the system, you can boot from the image in the current directory or the version in the backup directory. For more information about rebooting the system, see [“Rebooting the System” on page 3-5](#).

The following example boots the system from a backup image:

```
nx# reload backup
Save configuration information in /pcmcia0/startup.cfg? <y|n|q|f>y
Confirm reload from backup? <yes|no>y
```

If the system does not have a backup directory configured, the following messages appears:

```
Backup image directory is unset - reload aborted
```

The following command output shows the initial boot settings, before the boot from the backup directory:

```
nx# show boot
BOOT SYSTEM variable = /pcmcia0/releases/current/
BOOT BACKUP variable = /pcmcia0/releases/backup/
CONFIG_FILE variable = /pcmcia0/startup.cfg
```



After booting the system from the backup image, the boot settings appear as:

```
nx# show boot
BOOT SYSTEM variable = /pcmcia0/releases/backup/
BOOT BACKUP variable =
CONFIG_FILE variable = /pcmcia0/startup.cfg
```

- If you later issue the `reload` or `reload current` command, the boot parameters are restored to the original values.

## Working with Configuration Files

The system stores information about configuration changes in memory. For the system to use configuration changes the next time you boot the system, you must save the changes to the system startup configuration file.

The system creates a `startup.cfg` file at the top-level `/pcmcia0` directory. You can change the location of the configuration file, or specify a new name for a configuration file.

### Saving System Configuration to the Startup Configuration File

Run the `copy running-config startup-config` command to save changes to the startup configuration file. If you do not run this command, the system discards configuration changes made at the CLI. The changes are not saved.

The following example changes the memory allocation for the logging buffer and saves the configuration change to the startup configuration file:

```
nx# configure terminal
nx (config)# logging buffered 150000
nx (config)# exit
nx# copy running-config startup-config
```

### Changing the Location of the Startup Configuration File

The `boot config` command sets the location of the startup configuration file. The following example sets the startup configuration file to `/pcmcia0/newstartup.cfg`:

```
nx# configure terminal
nx(config)# boot config /pcmcia0/newstartup.cfg
nx(config)# exit
nx# show boot
BOOT SYSTEM variable = /pcmcia0/releases/current/
CONFIG_FILE variable = /pcmcia0/newstartup.cfg
```

## Synchronizing Files Between a Primary and Secondary RCP

In a system that uses both a primary RCP and secondary RCP, the files on the PCMCIA card in the secondary RCP should be synchronized with the files on the primary RCP to ensure that the system continues to work as expected should the primary RCP fail. The `copy sync` command lets you synchronize one file, all files, or the boot parameters.

If you synchronize all files, the system reformats the PCMCIA card in the second RCP and deletes the old files on the card, except for:

- .del files (deleted files)
- Recycled directory
- Inet on ethernet boot parameter

The following example synchronizes all of the files on the secondary RCP with the version of the files on the primary RCP:

```
nx# copy sync all
```

For additional information about the secondary RCP, see [Chapter 7, “Cards and Interfaces.”](#)

## Working with Routing Modules

The software modules for each of the routing protocols must be activated before you can configure or use those protocols.

### Activating a Software Routing Module

To activate a routing module on the system, run the **load** command followed by the module name. The system activates the routing module from the software image file loaded on the system.

The following example activates the BGP module and displays information about the status of all routing modules available on the system:

```
nx# configure terminal
nx(config)# load bgp
@1073797121 Starting bgp server built Mon May 7 14:54:06 EDT 2001
@1073797121 bgp service registration
Start loading Module bgp.o ...
Module spf.o loaded at 0x1be5be0
Register: reactivation of nxSpf (0x1cb5ab0)
nx(config)# exit
nx# show modules
```

Module	Service	Version	Status	Last Change	Message
bgp	nxBgp		active	MAY-10-2001 14:45:37	
isi	nxIsi		active	MAY-10-2001 12:20:17	
pim	nxPim		on_disk		/pcmcia0/releases/current /pim.o
spf	nxspf		active	MAY-10-2001 12:25:17	

## Stopping a Software Routing Module

You can stop a routing module to make it inactive on the system, and preserve the configuration for the module. You can start the module again by issuing the **start** command. You must stop a module before unloading it.



Take care when unloading a routing module. Issuing the **unload** command after a module is installed clears the configuration for the specified module.

You unload a software module by issuing the **unload** command at the `nx(config)#` prompt. Issuing the **unload** command at the `nx#` prompt reloads (reboots) the system.

The following example stops an active BGP module and displays the resulting module status:

```
nx# configure terminal
nx(config)# stop bgp
nx(config)# exit
nx# show modules
Module      Service  Version  Status      Last Change      Message
bgp         nxBgp             stopping   JUN-14-2001 14:25:37
nx# show modules
Module      Service  Version  Status      Last Change      Message
bgp         nxBgp             inactive   JUN-14-2001 14:25:37
```

## Working with Files and Directories

The CLI provides standard file and directory commands to manage files on the PCMCIA card. These commands let you move files and directories on the system.

**Table 3-4. File and Directory Command Usage**

Command	File management	Directory management	Remote file access
cd		✓	
copy file	✓		
delete	✓		
dir		✓	
ftpget			✓
ftpput			✓
mkdir		✓	

**Table 3-4. File and Directory Command Usage**

Command	File management	Directory management	Remote file access
pwd	✓	✓	
rename	✓		
rmdir		✓	
show workdir		✓	
squeeze	✓	✓	
tftpget *			✓
tftpput *			✓
type	✓		
what	✓		
workdir		✓	

\* For information about limitations for the TFTP protocol, see [“Working with Files on the Network” on page 3-11](#).

## Working with Directories

Typically, you work with the existing directory structure, but you can change it if needed. The following example shows how to:

- Create a directory called newreleases
- Set the working directory to newreleases
- Displays the path of the current working directory

```

nx# mkdir newreleases
nx# dir /all
size          date          time          name
-----
512    MAY-09-2001    20:33:48    PCS          <DIR>
512    MAY-09-2001    20:33:48    NEWRELEASES  <DIR>
512    MAY-14-2001    10:14:52    RELEASES     <DIR>
nx# configure terminal
nx(config)# workdir /pcmcia0/newreleases
nx(config)# exit
nx# show workdir
nx# Default Working Directory: /pcmcia0/newreleases

```

## Working with Files

The system provides commands to move files on the PCMCIA card, and to copy them to and from a network server. The software provides support for the File Transfer Protocol (FTP) and the Trivial Transfer Protocol (TFTP) to work with remote files.

### Working with Files Locally

The following example copies the `newstart.cfg` file to the top-level directory on the PCMCIA card. It then deletes the `startup.cfg` file:

```
nx# dir
size          date          time          name
-----
512           MAR-01-2001    09:45:37      .              <DIR>
512           MAR-01-2001    09:45:37      ..             <DIR>
512           MAR-01-2001    09:45:37      ..             <DIR>
12189         MAR-01-2001    11:53:22      STARTUP.CFG
64241         MAR-01-2001    13:11:46      NEWSTART.CFG
nx# copy file newstart.cfg /pcmcia0/newstart.cfg
nx# del startup.cfg
nx# dir
size          date          time          name
-----
512           MAR-01-2001    09:55:24      .              <DIR>
512           MAR-01-2001    09:55:24      ..             <DIR>
512           MAR-01-2001    09:55:24      ..             <DIR>
12189         MAR-01-2001    12:17:43      STARTUP.DEL
64241         MAR-01-2001    13:54:18      NEWSTART.CFG
nx# squeeze /pcmcia0
nx# dir
size          date          time          name
-----
512           MAR-01-2001    10:22:57      .              <DIR>
512           MAR-01-2001    10:22:57      ..             <DIR>
512           MAR-01-2001    10:22:57      ..             <DIR>
12189         MAR-01-2001    13:07:21      NEWSTARTUP.CFG
```

### Working with Files on the Network

The system provides standard FTP and TFTP commands to copy files to and from another system on the network. Use these commands to obtain updated copies of software from an FTP or TFTP server, back up files to a server, or restore files from a network server. Typically, you use FTP to transfer files on the network, because it is better suited to handle large files and uses less memory.

- Due to protocol limitations for TFTP, the `tftpget` and `tftpout` commands have a file transfer limitation of 30 Mb, and cannot be used to transfer image files.

## Transferring Files from a Network Server

The **ftpget** command copies files from a remote server to the PCMCIA card on the active route control processor. The following example retrieves a configuration file, **newconfig.cfg** from an FTP server:

```
nx# ftpget ascii 192.0.24.2 username userpassword
/software/config/newconfig /pcmcia0/newconfig
Opening data connection!!!!!!!
FTP Transfer Started!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
8119548 bytes written to 192.0.24.2
Transfer complete
nx#
```

The following table explains each argument used in the preceding **ftpget** command example.

Argument	Description
<b>ascii</b>	Copies file in ASCII, or text, format.
<b>192.0.24.2</b>	IP address of the remote FTP server.
<b>username</b>	The username required to log into the FTP server.
<b>userpassword</b>	The password (associated with the username) required to log into the FTP server.
<b>software/config/newconfig.cfg</b>	The path and filename of the file to copy from the FTP server.
<b>/pcmcia0/newconfig.cfg</b>	The location of the newconfig.cfg file on the PCMCIA card in the active RCP.

## Transferring Files to a Network Server

The **ftpput** command copies files from the PCMCIA card in the active route control processor to a remote server. The following example shows how to copy a file named **abc** to an FTP server:

```
nx# ftpput octet 192.0.24.6 anonymous password /test/image/test.tar
/test/abc
Opening data connection!!!!!!!
FTP Transfer Started!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
8119548 bytes written to 192.0.24.6
Transfer complete
nx#
```

The following table explains each argument used in the preceding **ftpput** command example.

Argument	Description
<b>octet</b>	Copies files in binary, or octet, mode. The command copies binary files byte for byte.

Argument	Description
<code>192.0.24.6</code>	The IP address of the remote FTP server.
<code>anonymous</code>	The username required to log into the FTP server
<code>password</code>	The password (associated with the username) required to log into the FTP server.
<code>test/image/test.tar</code>	The path and filename of the file on the PCMCIA card in the active RCP to copy to the FTP server.
<code>test/abc/test.tar</code>	The destination path and filename on the remote FTP server.





## System Security and Access

The system enables you to configure users assigning password authentication and access levels. These access levels determine what commands a user is able to execute within the system. The system also provides support for Remote Access Dial In User Servers (RADIUS). RADIUS secures networks and network services against unauthorized access by using authentication and accounting server software running on a centralized computer. Client authentication software runs on the system.

Other ways of providing system security are:

- Access Control List (ACL) (see [Chapter 12, “Access List Configuration”](#) for more information).
- Routing protocol authentication, for more information see:
  - [Chapter 10, “Point-to-Point Protocol Configuration”](#)
  - [Chapter 16, “OSPF Configuration”](#)
  - [Chapter 18, “BGP Configuration”](#)
- SNMP community security (see [Chapter 6, “Setting up System Monitoring”](#) for more information).

## Key Features

This implementation supports:

- Password authentication  
Required for system access and for File Transfer Protocol (FTP) and Telnet authentication.
- Password encryption
- Telnet session management
- RADIUS authorization for remote access

► You must have RADIUS installed on your network to use RADIUS for system security.

## Technology Concepts

Basic to understanding of system security and access are:


- Password authentication
- Password encryption
- Telnet access
- RADIUS authorization

### Password Authentication

To set up password authentication, you must assign each user, who has to access the system, a username and password. The privilege level configured for a user, read-only or read-write, determines which commands a user can run. Read-only access lets a user get information about the system by running `show` commands. Read-write access lets users make configuration changes. The system prompt indicates the user access level:

- Read-Only—`nx>`
- Read-Write—`nx#`

When configuring a user, you can specify a privilege level for the user. A user's privilege level determines which commands the user has access to after logging into the system. Privilege levels 0 through 14 provide equivalent read-only access and level 15 provides read-write access provides full command access.

- ▶ Although there are 16 privilege levels (0-15) defined, it is recommended that you only use privilege levels 0 (non-privileged) and 15 (privileged).
- ▶ At initial system configuration, you must configure the first user with read-write access (privilege level 15).
-  If you define usernames but do not copy the running configuration in memory to the startup configuration file, those users will not have access to the system when the system reboots.
- ▶ In order to Telnet or FTP to the system from a remote host, a user must be locally configured with read-write privileges. The username and password required for Telnet and FTP access are configured using the `username` command.

It is important to avoid a configuration that includes one or more non-privileged users but no privileged users. With such a configuration, only non-privileged users are able to log into the system, and it is not possible to configure the system further. To avoid this situation, the `username` command enforces the following rules:

1. The first user added to the system must be configured with privilege level 15.

2. If no users are configured, any user that logs in is assigned the highest privilege level (15) by default. Username and password are not prompted for in this case. After you configure one or more users, all subsequent login attempts require the user to enter a valid username and password.
3. Attempts to delete the last privileged user configured on the system are rejected if one or more non-privileged users are still configured.
4. Attempts to change the privilege level of the last privileged user configured on the system are rejected.

## Password Encryption

If you use username-password authentication for system security, you should also enable password encryption so that the password appears in encrypted format in the configuration file.

- The service `password-encryption` command does not determine whether individual user passwords are encrypted. Instead, it enables the capability to encrypt individual passwords. You must enable password encryption before any encrypted username commands for passwords to be encrypted.

Using the service `password-encryption` command affects the behavior of the `username` command, as follows:

- With password encryption disabled, `username` commands are not allowed to request password encryption. All user passwords are stored in unencrypted (clear-text) form. With password-encryption disabled, all `username` commands that request password encryption fail.
- With password encryption enabled, you can configure a user name with or without password encryption. The `encryption-type` argument used with `username` command controls whether or not to use password encryption on a user-by-user basis. When password encryption is specified for a user, that user's password is stored in encrypted form.



If you enable service `password-encryption` and later disable it, any users configured with an encrypted password while password-encryption was enabled are lost on the next system reboot. If the system reboots with a startup configuration that specifies no `password-encryption` all previously configured users are lost.

Since user names and passwords appear in configuration files (and in the running-configuration, displayed using the `show running-config` command), you may want to enable password-configuration and configure each user with an encrypted password.

## Telnet Access

You can configure terminal lines at the configuration prompt level. The system has a maximum of 32 terminal lines that are used to accept incoming Telnet sessions. You can configure these lines to filter on the source addresses of incoming Telnet connections, either individually, or in aggregate, over ranges of these terminal lines.

## RADIUS Authorization

The system provides support for Remote Access Dial In User Servers (RADIUS). RADIUS servers secure networks and network services against unauthorized access by using authentication and accounting server software running on a centralized computer. Client authentication software runs on the system.

RADIUS servers authenticate user logins, store information pertaining to each user's authorization levels on the network, and collect session information, such as the amount of time spent online, and modem connection speeds.

When services are requested of a RADIUS-enabled remote access server, the RADIUS client sends a request to the RADIUS server for authentication of the party requesting the service, along with authorization for the services that were requested. In turn, RADIUS servers answer requests by accessing user authentication information that is stored on the network.

### Primary Authentication Server

To set a primary authentication server, use the `primary-auth-server` command to specify which server to use as the primary authentication server for RADIUS authentication requests. The authentication server maintains the user password database. Each time you execute this command, you overwrite the previously configured primary server.

### Secondary Authentication Server

To set a secondary authentication server, use the `secondary-auth-server` command to specify which server to use as the secondary authentication server for RADIUS authentication requests. The authentication server is the server that maintains the user password database. The secondary server comes up automatically if the primary server is removed (using the `no primary-auth-server` command) or is off-line. Each time you execute this command, you overwrite the previously configured secondary server.

### Authentication Database

Once you have enabled password encryption on your RADIUS server, you define which database the system checks first to authenticate user passwords by issuing the `auth-order` command. When a request is made, the system checks either its local database or the RADIUS authentication server database, depending on which one is the primary authentication server.

## Username and Password Configuration

When you configure users the system enters the information into a system database for login authentication. You can also change passwords or privilege levels of an existing user.

**Table 4-1** describes the commands used for configuring and validating users and passwords.

**Table 4-1. Username and Password Command Usage**

Command	Configuration	Verification
<code>username</code>	✓	
<code>service password-encryption</code>	✓	
<code>show running-config</code>		✓

**To establish initial user accounts on the system:**

1. Enable password encryption.
2. Set up a user with read-write access to the system.
3. Add other users, specifying either read or read-write access.

► If you assign an encryption-level 1 (password encryption requested) but password-encryption is not enabled the `username` command rejects the request.

## Configuring Usernames and Passwords

The following example enables password encryption and configures two users on the system. The first user, `admin`, is configured with full command access (level 15) and `visitor` is configured with read-only (level 0) access. Both users are configured to have password encryption and the passwords are set (`firstpass` and `guestpass`).

```
nx# configure terminal
nx(config)# service password-encryption
nx(config)# username admin privilege 15 password 1 firstpass
nx(config)# username visitor privilege 0 password 1 guestpass
```

The following table explains each syntax line used in the preceding example:

Configuration Line	Description
<code>configure terminal</code>	Puts the router into terminal configuration mode.
<code>service password-encryption</code>	Enables encryption of passwords.

Configuration Line	Description
<code>username admin privilege 15 password 1 firstpass</code>	Sets a username, privilege level (read-write) and password for a user named admin.
<code>username visitor privilege 0 password 1 guestpass</code>	Sets a username, privilege level (read-only) and password for a user named visitor.

## Verifying Username Configuration

The `show running-config` command shows the usernames configured. The 0 or 1 preceding the password string indicates whether it is encrypted or not. Zero (0) indicates that the password is not encrypted, 1 indicates that it is.

```
nx# show running-configuration
Current configuration:
service password-encryption
!
auth-order local
username admin privilege 15 password 1 t34hg39f
username visitor privilege 0 password 1 t46dh47e
```

Where 1 indicates password encryption and t34hg39f and t46dh47e are the encrypted passwords.

- If you configure usernames but do not copy the running configuration file to the startup configuration file, those users will not have access to the system when the system reboots.

## Telnet Access Configuration

Telnet access enables you to establish a remote connection to a system and issue commands via this remote login.

**Table 4-2** describes the commands used for Telnet access.

**Table 4-2. Telnet Command Usage**

Command	Configuration	Verification
<code>access-class</code>	✓	
<code>line</code>	✓	

**Table 4-2. Telnet Command Usage**

Command	Configuration	Verification
line vty	✓	
session-timeout	✓	
terminal length	✓	
show users		✓

## Configuring Telnet Access

Using the access-class command enables you to filter a specified terminal line on an incoming Telnet session. You need to choose a standard access-list identified by numbers between 1 and 99.

- The access-class command only applies to incoming Telnet sessions.

The following example applies access list 46 to an incoming Telnet session on line 3:

```
nx# terminal length 200
nx# configure terminal
nx(config)# line vty 3
nx(config)# session-timeout 60
nx(config-line)# access-class 46 in
```

The following table explains each syntax line used in the preceding example.

Configuration Line	Description
<code>terminal length 200</code>	Sets the number of lines that can be written to the active window.
<code>configure terminal</code>	Puts the router into terminal configuration mode so you can enter system commands.
<code>line vty 3</code>	Enters line configuration mode for a terminal line 3.
<code>session-timeout 60</code>	Sets the current inactivity limit to 60 minutes. When the time limit is exceeded, the connection is cancelled. To disable session time out, set the limit to zero (infinity).

Configuration Line	Description
<b>access-class 46 in</b>	Applies access list 46 to an incoming Telnet session on line 3.

To manage Telnet sessions, issue a **show users** command to display active and idle terminal line connection information:

```
nx# show users
Line      User      Idle      Location
0         con      00:00:26
3         vty                        135.17.241.46
```

## Clearing a Login Session

The following example shows how to display terminal-line information about current session login sessions, and then deleting an active line connection:

```
nx# show users
Line      User      Idle      Location
0   con  Admin
1   vty  users1 00:00:33 192.14.9.2
nx# clear line 1
nx# show users
Line      User      Idle      Location
0   con  Admin
nx#
```

The following table explains each syntax line used in the preceding example.

Command	Description
<b>show users</b>	Displays terminal-line information about the current user.
<b>clear line 1</b>	Deletes a remote Telnet session on a specified active line connection.

## RADIUS Configuration

Before you configure primary or secondary authentication servers, you must issue the **secret** command to set your password key. The **secret** command enables authentication and defines the password key used for authentication. The key validates and authorizes communication between the system and the RADIUS daemon. You can use any printable ASCII characters in the string except for the **Tab** key.



## Configuring RADIUS on the System

The following table describes the commands used for configuring and validating RADIUS:

**Table 4-3. RADIUS Configuration and Validation Commands**

Command	Authentication servers	Authentication Database	Access Server Attributes	Verification
auth-order		✓		
primary auth-server	✓			
router-ip			✓	
secondary-auth-server	✓			
secret	✓	✓	✓	
show radius				✓

**To configure RADIUS on your system:**

1. Enable RADIUS authentication using the `radius` command.
1. Enable password encryption by creating a password key using the `secret` command.

► The `secret` command requires quotation marks around a password key.

2. Assign your primary authentication server using the `primary auth-server` command.
3. Assign your secondary authentication server using the `secondary-auth-server` command.
4. Assign your authentication database using the `auth-order` command.
5. Set your Network Access Server IP (NAS-IP) attribute in the RADIUS authentication request using the `router-ip` command.

The following example sets a password key to `sysadmin` only, configures a primary authentication server and sets a UDP port for Radius requests, configures a secondary authentication server, and then configures the system to first check the local database and then check the RADIUS authentication server database.

```
nx# configure terminal
nx(config)# radius
nx(config-radius)# secret "sysadmin only"
nx(config-radius)# primary-auth-server 192.0.2.4 port 1100
nx(config-radius)# secondary-auth-server 192.0.2.6 port 1600
nx(config-radius)# exit
nx(config)# auth-order local radius
```

The following table explains each syntax line used in the preceding example that illustrates how to set your password key.

Configuration Line	Description
<code>configure terminal</code>	Accesses configuration mode.
<code>radius</code>	Provides access to RADIUS commands in the system. RADIUS commands are entered from this prompt level.
<code>secret "sysadmin only"</code>	Enables authentication and defines the password key used for authentication.
<code>primary-auth-server 192.0.2.4 port 1100</code>	Configures the IP address of the primary RADIUS authentication server as 192.0.2.4 and also sets the UDP port as 1100. This port is used by the RADIUS software to accept RADIUS requests.
<code>secondary-auth-server 192.0.2.6 port 1600</code>	Configures the IP address of the secondary RADIUS authentication server as 192.0.2.6 and also sets the UDP port as 1600. This port is used by the RADIUS software to accept RADIUS requests.
<code>auth-order local radius</code>	Configures the order RADIUS authentication is done in (first the local database and then check the RADIUS authentication server database).

The following example shows the output for the `show running-config` command:

```
nx# show running-config
Building configuration....
Current configuration:
service password-encryption
!
auth-order local radius
radius
  secret sysadmin only
  primary-auth-server 192.0.2.4
  secondary-auth-server 192.0.2.6
.
.
.
```

## Verifying RADIUS Configuration

The following table shows RADIUS commands used for verification.

**Table 4-4. RADIUS Commands to Verify Configuration**

Action	Command
Display current RADIUS settings.	<code>show radius</code>

**Table 4-4. RADIUS Commands to Verify Configuration**

Action	Command
Display current running configuration settings.	<code>show running-config</code>



## Working with the Log and Debug Utilities

The log utility manages messages generated in response to changes and error conditions on the system. Typically, you configure the log utility to work with a syslog server on the network. Alternatively, you can set up a file to record log messages.

The system also provides a debug utility that works in conjunction with the log utility. The debug utility enables the system to send debug messages for specified protocols to the log utility. You view debug messages as you would other log messages.

### Key Features

The log utility provides the following features:

- Support for standard message severity levels
- A log buffer and a log history buffer
- Support for syslog servers

The debug utility provides debugging support for specified modules.

### Technology Concepts

Working with the log and debug utilities on the system requires a basic understanding of the following:

- Message severity levels
- Log buffers
- Initial configuration for the log utility
- Module support for the debug utility

## Message Severity Levels

The log utility supports standard logging levels. The system stores messages for a configured severity level and all numerically lower levels. [Table 1-1](#) lists the logging levels and the syslog definition for each severity level:

**Table 5-1. Message Severity Levels**

Level	Severity	Description	Syslog Designations
0	emergencies	Makes the system unusable	LOG_EMERG
1	alerts	Requires immediate action	LOG_ALERT
2	critical	Requires attention	LOG_CRIT
3	errors	Indicates an error condition	LOG_ERR
4	warnings	Indicates a noncritical problem	LOG_WARNING
5	notifications	Notifies you of a significant condition	LOG_NOTICE
6	informational	Provides only informational messages	LOG_INFO
7	debugging	Indicates debugging condition	LOG_DEBUG

## Log Buffers

The system stores log messages in buffers. The log utility provides two types of buffers:

- A circular log buffer  
This buffer receives and stores system messages. The system replaces the oldest messages in the buffer with the newest ones when the buffer fills. By default, this buffer stores approximately 200 messages. The maximum capacity of the buffer is 1000 messages.
- A log history buffer  
By default, this buffer stores approximately 21 messages and has a severity level set to warning. The log history buffer has a maximum capacity of 500 messages.

You can view messages currently in the log buffer or the log history buffer from the CLI.

## Initial Configuration for the Log Utility

By default, logging is enabled on the system and supports:

- Sending log messages to the log buffer and sending more critical log messages (as specified) to the log history buffer
- Receiving log messages available through the route control processor (RCP), including those sent from the card control task to show card status  
You can configure logging for line cards to send all card messages to the RCP. The log buffer on each line card stores approximately 200 messages.

Typically, you set up the log utility to send messages to a syslog server or to a file, depending on your network configuration.

The default configuration sets severity-levels that can be changed. **Table 5-1** lists the default log level for the system logging:

**Table 5-2. Default Severity Levels for Logging Information**

Log Type	Default Log Level
Log to a file	Level 7: Debugging
Log to a syslog server	Level 6: Informational
Line card log messages	Level 7: Debugging
Log history buffer	Level 4: Warning

## Displaying Log Messages

The system stores log messages in a syslog server or from a file, as configured. You view messages from the console or from a Telnet session, or through the CLI. The following table lists the default severity levels:

**Table 5-3. Default Severity Levels for Message Display**

Display Type	Default Log Level
Display messages to the console	Level 6: Informational
Display messages to a Telnet session	Level 6: Informational

You can also display log messages inline as they enter the buffer. Inline logging:

- Ensures that all messages appear at the console.

In a situation where the log utility receives many messages, all messages may not appear on the screen between subsequent **show logging** commands run at the console.

- Lets you observe log messages when you issue a command. This lets you see whether running a CLI command initiates any system messages.

Typically, you enable inline logging for troubleshooting system problems.

- Inline logging is a high priority task and should be used with caution. Activating inline logging can slow system performance.

## Module Support for the Debug Utility

The debug utility sends all log messages (with a severity level of debugging) for a specified module to the log utility. It also supports logging debug messages identified by a message number. You view debug messages as you would other log messages.

The following modules provide debugging commands:

- ATM
- BGP (as IP BGP)
- Frame Relay
- IS-IS
- MPLS
- OSPF (as IP OSPF)
- PPP
- QoS
- RSVP

Use **debug** commands only in troubleshooting situations. These commands can increase system load and slow network performance. Turn **debug** commands off as soon as they are no longer needed.

## Logging Configuration

Typically, you set up logging to a network syslog server, but you can also configure logging to a file. In most cases, the default values for other log parameters can remain the same. The following sections describe how to set up logging to a syslog server or to a file, and how to change settings for the log utility. For information about the default configuration for the log utility, see [“Initial Configuration for the Log Utility” on page 5-2](#).

- The *NX64000 Command Reference* manual describes all commands referenced in this chapter.

The following table lists the logging commands.

**Table 5-4. Log Command Usage**

Command	Log messages to the syslog server	Log messages to a file	Set buffer size	Set severity level	View messages at a console	View messages from Telnet session	Display configuration information	Empty log buffer
clear logging								✓
logging buffered			✓					
logging console				✓	✓			
logging console inline				✓	✓	✓		



**Table 5-4. Log Command Usage**

Command	Log messages to the syslog server	Log messages to a file	Set buffer size	Set severity level	View messages at a console	View messages from Telnet session	Display configuration information	Empty log buffer
logging facility	✓							
logging file		✓		✓				
logging history				✓				
logging history size			✓					
logging linecard				✓				
logging monitor				✓		✓		
logging on	✓	✓						
logging source-interface	✓							
logging syslog-ip	✓							
logging trap	✓			✓				
show logging					✓	✓	✓	
terminal monitor				✓		✓		

## Setting up Logging to a Syslog Server

Many networks rely on syslog servers to collect log messages for network systems. The following example shows how to set up logging to a syslog server:

```
nx# configure terminal
nx(config)# logging syslog-ip 10.1.1.1 syslog-svr
nx(config)# logging source-interface ethernet0
nx(config)# logging facility syslog
nx(config)# logging trap errors
```

The following table describes the logging commands used in the preceding example:

Configuration Line	Description
<code>configure terminal</code>	Enter terminal configuration mode so you can enter system commands.
<code>logging syslog-ip 10.1.1.1 syslog-svr</code>	Set the IP address of the destination syslog server to 10.1.1.1 and the name of the server to syslog-svr.
<code>logging source-interface ethernet0</code>	Set the ethernet0 interface as the source port for messages sent to the syslog server. All messages destined for the syslog server show the address of this interface as the source address.
<code>logging facility syslog</code>	Set the logging facility type to syslog server.
<code>logging trap errors</code>	Send error, critical, alert, and emergency messages to the syslog server.

## Setting up Logging to a File

The following example shows how to create a log file:

```
nx# configure terminal
nx(config)# logging file /pcmcia0/logfile.log errors
```

The following table describes the commands used in the preceding example:

Configuration Line	Description
<code>configure terminal</code>	Enter terminal configuration mode so you can enter system commands.
<code>logging file /pcmcia0/logfile.log errors</code>	Send error, critical, alert, and emergency messages to the logfile.log errors file.

## Viewing Log Messages

This section describes the different ways you can view log messages from the log buffer. For more information about viewing log messages, see [“Displaying Log Messages” on page 5-3](#).

### Viewing Log Messages from the CLI

The `show logging` command displays messages from system buffers. The following table shows the command options to use to display different sets of messages:

To view messages	Use this command option
The contents of the log buffer	<code>none</code>
The contents of the log history buffer	<code>history</code>
Messages for a specified card (by identifying the slot in which the card resides)	<code>slot <i>slot-number</i></code>

The following example shows sample messages displayed from the log buffer:

```
nx# show logging
Logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  No Syslog Host configured
  Buffer logging: level INFO, 272 messages logged
Log Buffer (200 messages)
00:00:01.027    ALERT SOURCE    NML (tRootTask      )  SLOT  RCP_0
Network management library has initialized.
00:00:01.028    ALERT SOURCE    SNMP (tSnmpd        )  SLOT  RCP_0
SNMP Agent starting, built on Sat Oct 4 07:21:30 GMT-5:00 2000
00:00:01.035    ALERT SOURCE    CCT (tCCT          )  SLOT  RCP_0
Resetting cards:  action = 0x2.  slots = 0xcfffff
.
.
.
```

### Viewing Log Messages from a Console

Messages display to the console after you enable logging to the console. The following example displays messages to the console, then disables message display:

```
nx(config)# logging console alerts
nx(config)# no logging console
```

The following table describes the commands used in the preceding example:

Configuration Line	Description
<code>logging console alerts</code>	Display alert and emergency messages to the console.
<code>no logging console</code>	Turn off message display to the console.

## Viewing Log Messages from a Telnet Session

You can view system messages remotely from a Telnet session.

The following example displays log messages to a Telnet session and sets the severity level, then turns off message display:

```
nx# terminal monitor
nx# configure terminal
nx(config)# logging monitor critical
nx(config)# exit
nx# terminal unmonitor
```

The following table describes the commands used in the preceding example:

Configuration Line	Description
<code>terminal monitor</code>	View log messages from an active Telnet session.
<code>configure terminal</code>	Enter terminal configuration mode so you can enter system commands.
<code>logging monitor critical</code>	Enables the display of critical, alert, and emergency messages to a Telnet session.
<code>terminal unmonitor</code>	Turn off message display to a Telnet session.

- The `terminal monitor` and `logging monitor` commands can be entered in any order.

## Viewing Log Messages Inline

You can also display messages as the log buffer receives them by using inline logging. Inline logging is a high priority task and can slow system performance.

- You should turn inline logging off as soon as possible by using the `no logging console inline` command.

The following example displays critical, alert, and emergency messages inline to the console, then turns off inline message display:

```
nx# terminal monitor
nx# configure terminal
nx(config)# logging console inline
nx(config)# no logging console inline
```

## Getting Card Information

The log utility gathers basic information about system cards. The `logging linecard` command lets you collect log messages stored by each of the line cards. Because each card stores approximately 200 messages, you should send messages to a file or to a syslog server to make all of the messages available. If the messages are sent only to the buffer, messages will be lost as newer messages overwrite older ones in the log buffer.

The following example shows how to enable logging from all line cards:

```
nx# configure terminal
nx(config)# logging linecard
```

## Changing Buffer Sizes

The following example shows how to change the log buffer size to 500 messages, and the log history buffer to 50 messages:

```
nx# configure terminal
nx(config)# logging buffered 500
nx(config)# logging history size 50
```

## Verifying Log Configuration

To view information about the logging configuration for the system, run the `show logging` command. The initial lines in the command output display configuration information.

The following example shows logging of information messages enabled, and the size of the log buffer set to 200 messages. Logging to a file or syslog server is not configured:

```
nx# show logging
Logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
      No Syslog Host configured
      Buffer logging: level INFO, 272 messages logged
Log Buffer (200 messages)
.
.
.
```

The `show logging` command also provides information about the types of messages available for system modules, which modules are active on the system, and the number of messages logged for each module.

## Debugging Configuration

The software provides a comprehensive set of debugging commands to display debug messages for specified modules. You use the debug commands for troubleshooting system problems. [Table 5-5](#) lists the debug and related commands.

**Table 5-5. Debug Command Usage**

Command	Enable debug messages	View debug messages	Store debug messages in buffer	Disable all debug commands
debug atm arp	✓			
debug atm errors interfaces	✓			
debug atm errors vc	✓			
debug atm packet	✓			
debug frame-relay lmi	✓			
debug frame-relay pvc	✓			
debug ip bgp	✓			
debug ip ospf	✓			
debug isis	✓			
debug logging buffer			✓	
debug logging console		✓		
debug mpls	✓			
debug ppp	✓			
debug qos	✓			
debug rsvp	✓			
show debugging		✓		
show event-trace		✓		
show tech-support		✓		
undebug all				✓

## Enabling Debugging for a Protocol

All of the debug commands work in a similar way. The following example shows how to enable debugging for quality of service, then how to disable debugging for all debug commands enabled:

```
nx# debug qos
nx# undebug all
```

## Getting Information for Debugging

The following table lists the debugging-related **show** commands:

**Table 5-6. Show Commands for Debugging**

Action	Command
Display the status and setting for ATM debug commands.	<b>show debugging</b>
Display information about events that have an effect on system cards: <ul style="list-style-type: none"><li>• Card state</li><li>• Time the card enters a state</li><li>• Duration of the state</li><li>• Event that changes the card state</li></ul>	<b>show event-trace</b>
Display comprehensive system information to provide to Lucent Customer Support.  Run this command and save the output to a file before you contact Lucent Customer Support.	<b>show tech-support</b>





# Setting up System Monitoring

The Simple Network Management Protocol (SNMP) is a network monitoring tool used by network managers to monitor and manage the state of their network. The term SNMP refers to a set of standards that provide a framework for network management information, along with a protocol for exchanging that information. NX-IS supports SNMP versions 1 and 2.

This chapter explains how you can use SNMP to monitor your NX64000 IP Core Routers.

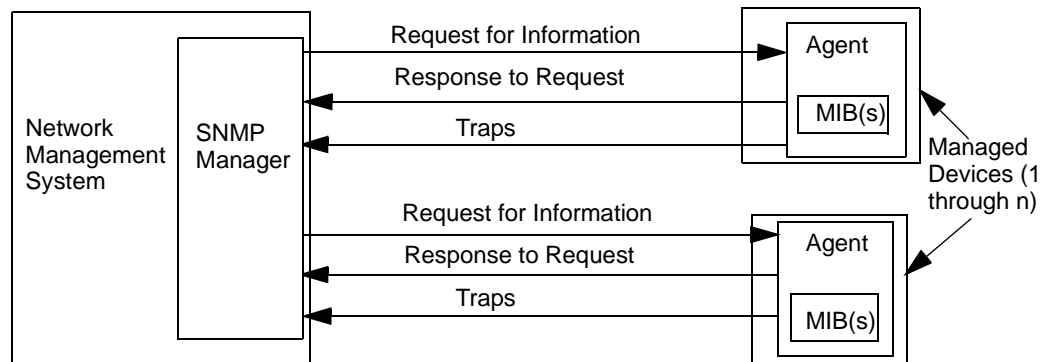
## Technology Concepts

The following concepts are basic to understanding how SNMP works:

- SNMP Manager and Agent Model
- SNMP Polls and Traps
- SNMP Communities
- Management Information Base (MIB)

### SNMP Manager and Agent Model

SNMP is based on the client/server model. This terminology, however, should not be applied to SNMP. In SNMP, the client is the network management software or manager that runs on the network management system (NMS), the server is referred to as the agent which runs on the device(s) being managed. A system can operate exclusively as a manager or as an agent, or it can be both. SNMP is the protocol the manager and agent(s) use for their communications.



**Figure 6-1. SNMP Manager and Agent Model**

## SNMP Manager

The SNMP network management software, or simply the manager, runs on the network management system (NMS) and, in addition to monitoring and controlling network elements (agents), it provides the interface between agents and users. An SNMP manager can manage multiple SNMP agents.

The manager performs two main functions in relation to the agent. One function is to monitor the agent's status and performance by periodically requesting (polling) the agent to provide this information.

The manager's second function is to manage the agent and the device. The manager accomplishes this by requesting the agent to change the value of the object in the MIB relating to the device function that the manager wants to change. For example, enabling or disabling ports or setting OSPF parameters.

## SNMP Agent

A managed device can be a PC, a workstation, a router (such as the NX64000 system), a file server, etc. The SNMP agent is a software module running on the managed device that is responsible for maintaining the information pertaining to the device's status, performance, and other functions and delivering that information to the manager when polled. The agent maintains this information in its Management Information Base(s) (MIBs).

An information exchange can be initiated by the manager through polling or by the agent through a trap. When the SNMP agent receives a request for information from the manager, the agent checks the appropriate MIB for the corresponding object identifier for which the manager is requesting an update. If the agent finds the object, it returns the value for that object to the manager. If the agent does not find the object, it returns a "no such instance" error.

## SNMP Polls and Traps

SNMP sessions are created when the SNMP manager requests information from an agent. These periodic requests for information are referred to as polls. Polling occurs at a rate determined by the manager.

A disadvantage of polling is that if an event occurs, for example, a link goes down or there is an authentication failure, the manager will not know about it until the next poll. To make possible real-time notification of the manager, SNMP implemented traps. Traps enable agents to send managers real-time notification about significant, pre-defined events outside of the normal polling sequence. Once the manager receives the trap, it notifies the network manager by setting an alarm or sending a message. It is then up to the network manager to resolve the problem. (See also [“Configuring SNMP Contact and Location Information” on page 6-7.](#))

Polls and traps can happen simultaneously since, while polling happens at a set frequency, traps happen whenever an agent identifies a problem or potential problem.

Traps are enabled with the `trapv1` or `trapv2` arguments of the `snmp-server community` command. (Refer to [“Configuring an SNMP Community” on page 6-7](#) for an example of how to enable traps.

## SNMP Communities

An SNMP community defines the relationship between an agent and a manager. Each community is given a community name (community string) which is equivalent to a password in that the manager must know the community name to have access to the agent. (If `snmp-server trap-authentication` is enabled, when an agent receives an SNMP packet with an invalid community string, it sends an authentication failure trap to all managers configured to receive traps.)

In addition, in configuring the community using the `snmp-server community` command (refer to [“Configuring an SNMP Community” on page 6-7](#)), the agent defines 3 important community characteristics that determine the level of access that the manager has to the agent. The characteristics are:

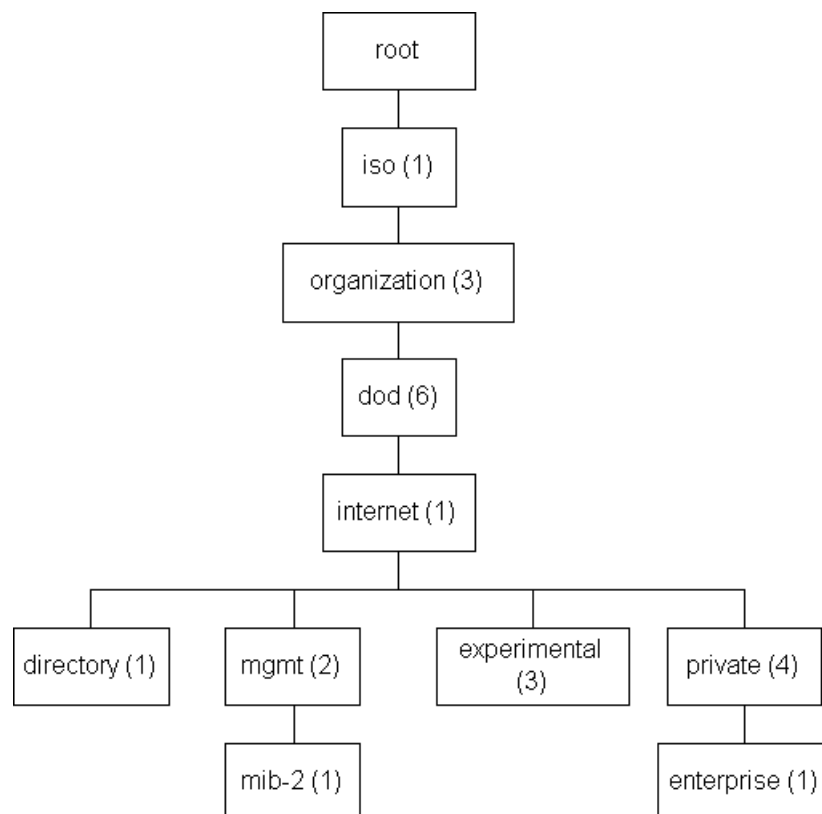
- **View** — a view serves as an access control mechanism to include or exclude portions of a MIB hierarchy tree from a community, therefore, defines what parts of the MIB hierarchy are visible to the manager. Views are configured with the `snmp-server view` command. (Refer to [“Management Information Base \(MIB\)” on page 6-4](#) and [“Configuring an SNMP Community” on page 6-7](#) for more information on views.)
- **Read-only or read-write privileges** — the default is read-only privileges which give the manager the capability of retrieving data from a MIB thus enabling the manager only to monitor the agent’s activity. To manage the agent, the manager must be able to change data in the MIB which requires read-write privileges. Read-only and read-write privileges are enabled with the `ro` and `rw` arguments of the `snmp-server community` command. In the NX64000 system, the default name for a community with read-only privileges is `public` and the name for a community with read-write privileges is `private`.
- **Traps** — traps are enabled with the `trapv1` and `trapv2` arguments of the `snmp-server community` command. If one of these arguments is not specified no traps are sent which is the default condition. `trapv1` sends SNMP version 1 compatible traps while `trapv2` sends SNMP version 2 compatible traps. The `snmp-server host` command lets you specify which manager is to receive traps by adding the manager to a particular community. (Refer to [“Configuring an SNMP Community” on page 6-7](#) for an example on using the `snmp-server host` command.)

Communities and community names are necessary for all SNMP operation between manager(s) and agent(s). An agent can establish a number of communities with different managers allowing each manager various levels of access. The agent can also add managers to a previously defined community using the `snmp-server host` command to give the new manager the privileges already defined for that community. Both managers and agents build tables to track communities to which they belong.

## Management Information Base (MIB)

A managed object is a piece of information about a managed device describing, for example, the state of an interface. The Management Information Base (MIB) is a database of managed objects that resides with the agent software on the managed device. Agents can implement numerous MIBs but all agents implement the *Management Information Base for Network Management of TCP/IP-based internets: MIB-II* (RFC 1213). The NX-IS software supports over 40 standard and proprietary MIBs (refer to the *NX64000 Command Reference* for a complete list of these MIBs). Collectively the MIBs describes the overall state of the managed device.

Managed objects are organized in a hierarchy that is defined by RFC 1155, *Structure and Identification of Management Information for TCP/IP-based Internets* and RFC 2578, *Structure of Management Information Version 2 (SMIv2)*. **Figure 6-2** gives an overview of the hierarchy.



**Figure 6-2. Object ID Tree**

Of the four branches shown:

- The directory branch is not used.
- The management (mgmt) branch is for the standard MIBs that are in general use in the industry, for example, RFC 1213 mib-2 and RFC 1850 OSPF.
- The experimental branch is to support new areas of network management software.
- The enterprise branch is for company specific MIBs, that is, those written exclusively by a company for its particular object(s), for example, nx64000chassis.mib.

As stated above, SNMP views, created with the `snmp-server view` command, are access control mechanisms to allow a community to view (include) or not view (exclude) portions of the MIB hierarchy tree.

### Managed Object ID

Each managed object in the MIB is identified by its name or objectID (OID). A managed objectID is of the form: 1.3.6.1.2.1.11.1.0 where:

1.3.6.1.2.1	These are identifiers from the OID tree from iso to mib-2 (see <a href="#">Figure 6-2</a> ).
11	Objects in MIBs are grouped by type. MIB-2 includes the following groups which also have corresponding number identifiers: System(1), Interfaces(2), Address translation(3), IP(4), ICMP(5), TCP(6), UDP(7), EGP(8), Transmission(10), and SNMP(11). Therefore 11 refers to an a managed object in the SNMP group.
1	Refers to the specific managed object in the SNMP group which, in this case, is <code>snmpInPkts</code> , number of incoming packets.
0	This is the Instance Identifier which specifies a specific instance of an object. What it identifies is based on what the object represents. For example, if the object is used to enable or disable a port, the instance identifier would specify the port number.

## SNMP Configuration

This section provides examples of basic SNMP configuration tasks, along with tables that explain how the commands are used to configure and maintain an SNMP network. [Table 6-1](#) lists the SNMP commands that comprise the current implementation and their application in configuring and monitoring SNMP.

**Table 6-1. SNMP Command Usage**

Command	Configures Access to Mgmt. Station	Sets SNMP Parameters	Deletes SNMP Statistics	verify SNMP Configuration
clear snmp-counters			✓	
show communities				✓
show snmp				✓
show views				✓
snmp-server community	✓			
snmp-server contact	✓			
snmp-server-host	✓	✓		
snmp-server-location		✓		
snmp-server-packetsize		✓		
snmp-server-trap-authentication		✓		
snmp-server-trap-source		✓		
snmp-server-view	✓			

► All commands discussed in this chapter are described in the *NX64000 Command Reference*.

SNMP is enabled at startup. If SNMP has already been configured on your NX64000 system, you can issue the `show communities` command to view SNMP configuration information for the system. The following section provide examples of the following configuration tasks:

- Configuring an SNMP view.
- Configuring an SNMP Community, giving the community read/write access, and enabling traps.
- Adding a host/manager to an SNMP community.
- Configuring contact and location information

## Configuring an SNMP Community

The following examples shows how to configure an SNMP view and an SNMP community and add a host to the community.

```
nx(config)# snmp-server view boston 1.3.6.1.4.1* exclude
nx(config)# snmp-server community dublin view boston rw trapv1
nx(config)# snmp-server host 172.21.2.2 dublin
```

The following table explains how the commands are used in the preceding example.

Configuration Line	Description
<code>snmp-server view boston 1.3.6.1.4.1* include</code>	Exclude communities assigned the SNMP view called <i>boston</i> access to everything below the enterprise MIB level, that is below 1.3.6.1.4.1 (see <a href="#">Figure 6-2</a> ).
<code>snmp-server community dublin rw trapv1</code>	Create an SNMP community named <i>dublin</i> giving read-write access to all managers in the community. The <i>trapv1</i> argument specifies that SNMP version 1 compatible traps are sent to the managers in the community.
<code>snmp-server host 172.21.2.2 dublin</code>	Add a manager/host (in this case 172.21.2.2) to the community <i>dublin</i> and, as a consequence the new host will receive traps sent to the community.

## Configuring SNMP Contact and Location Information

The following example shows how to add contact and server location information to the SNMP configuration. Both contact and location information can be up to 255 alphanumeric characters in length.

```
nx(config)# snmp-server contact System Admin Jane Doe @ ext 5678
nx(config)# snmp-server location New England Branch Office
```

The following table explains how the commands are used in the preceding example.

Configuration Line	Description
<code>snmp-server contact System Admin Jane Doe @ ext 5678</code>	Add system contact information (Jane Doe at extension 5678) to SNMP configuration information.
<code>snmp-server location New England Branch Office</code>	Add system/server location information to the SNMP configuration information. In this case, the location is the New England Branch Office.

## Clearing SNMP Counters

The NX64000 switch/router keeps detailed statistics about data traffic it has sent and received on its interfaces, bad data packets that it has seen, etc. Using this information you can determine if there have been problems on the network regarding router interface services.

The following table displays the SNMP counters that track this information and that are cleared with the `clear snmp-counters` command:

`nx# clear snmp-counters`

snmpInPkts	snmpInReadOnlys	snmpOutBadValues
snmpInBadVersions	snmpInGenErrs	snmpOutReadOnlys
snmpInBadCommunityNames	snmpInTotalReqVars	snmpOutGenErrs
snmpInBadCommunityUses	snmpInTotalSetVars	snmpOutGetRequests
snmpInASNParseErrs	snmpInGetRequests	snmpOutGetNexts
snmpEnableAuthenTraps	snmpInGetNexts	snmpOutSetRequests
snmpOutPkts	snmpInSetRequests	snmpOutGetResponses
snmpInBadTypes	snmpInGetResponses	snmpOutTraps
snmpInTooBig	snmpInTraps	snmpOutSilentDrops
snmpInNoSuchNames	snmpOutTooBig	snmpOutProxyDrops
snmpInBadValues	snmpOutNoSuchNames	

## Verifying and Monitoring SNMP Configuration

Use the following commands to verify and monitor your SNMP configuration. Also run the appropriate `show` command, such as `show communities`, to verify what parameters are already configured prior to making any configuration changes. All these commands are run at the global level.

**Table 6-2. SNMP Commands for Verification and Monitoring**

Action	Command
Display all the SNMP MIB counters on your network, along with listing contact name and location information.	<code>show snmp</code>
Display configured views for a host.	<code>show views</code>
Displays information about the configured SNMP communities on your network.	<code>show communities</code>



## Cards and Interfaces

This chapter describes how to configure the modules and interfaces in the NX64000 IP Core Router. The system provides high-port density, with interface speeds at OC-3, OC-12, OC-48, OC-192, Gigabit Ethernet and DS3. The system cards support Packet over SONET (POS), Asynchronous Transfer Mode (ATM), Ethernet and serial (DS3). Loopback, null, and tunnel interfaces are provided for use in special situations.

### System Interface Cards

The 40-slot system supports the following number of modules:

**Table 7-1. Module Description**

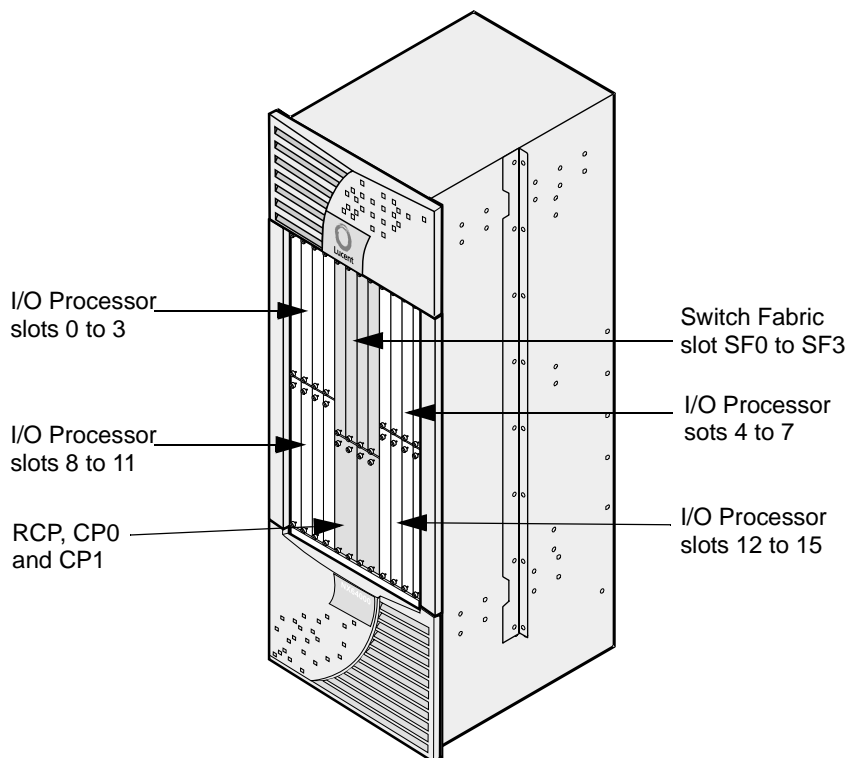
Number of Supported Modules	Module Name	Description
1-16	I/O Processing Engines (IOP)	The IOPs retain local copies of the routing table and handle control and signaling functions.
1-16	I/O Line Card Adapters (IOA)	The IOAs provide the physical network connectivity.
1-4	Switch Fabric Modules (SF)	The switch fabric stores packets from the network.
1-2	SONET/SDH Timing/Alarm Modules (STA)	The Chassis Timing and Alarm card (CTA). The CTA is also called SONET Timing/Alarm module (STA).  The interface uses either the default internal chassis clock or can synchronize with an external device (a line or external clock).
1-2	Route Control Processors (RCP)	The RCP maintains the forwarding engine, processes and updates route tables.

The system chassis has a mid-plane architecture design. The modules are inserted into both the front and the back of the chassis. **Table 7-2** lists the slot locations and corresponding card orientations.

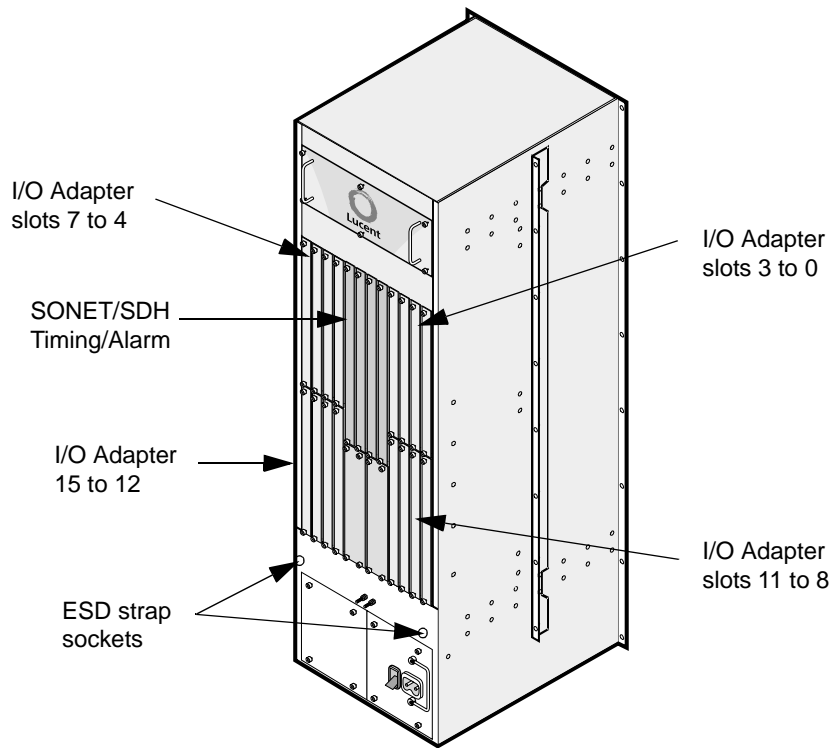
**Table 7-2. Slot Locations and Card Orientations**

Module Type	Slots	Slot Location in Chassis	Label and Slide Latch Orientation
Processor Engine (IOP)	0 through 7	Front	Top
Processor Engine (IOP)	8 through 15	Front	Bottom
Line Card (IOA)	0 through 7	Rear	Bottom
Line Card (IOA)	8 through 15	Rear	Top
Switch Fabric	SF0 through SF3	Front	Top
SONET Timing/Alarm	STA0 or STA1	Rear	Bottom
Route Control Processor	CP0 or CP1	Front	Install with PCMCIA slot at top

**Figure 7-1** and **Figure 7-2** on page 7-3 show the front and rear view of the system, and identify the locations of the system hardware components.



**Figure 7-1. Front View of Chassis**



**Figure 7-2. Rear view of Chassis**

## Interface Speeds

**Table 7-3** displays the available interface speeds.

**Table 7-3. Interface Speed**

Interface	Speed (Mb/sec.)
OC-3	155
OC-12	622
OC-48	2488
OC-192	9953
Gigabit Ethernet ENET 1000	1000
DS3	45

## Removing and Replacing Modules

The modules are inserted into the system when the router is installed. The *NX64000 Installation Guide* describes the system hardware modules and provides a list of field replaceable units (FRU) and associated order numbers.

When you remove an IOA or IOP from a slot and replace it with a different type of card, you must reconfigure the slot before the new card will function properly. For example, if you replace an OC-3 ATM IOP with an OC-3 POS IOP or an OC-3 ATM IOA with an OC-12 ATM IOA, then you must update the configuration.

- See the *NX64000 Troubleshooting Guide* for safety precautions and detailed information about removing and replacing system cards.

## Switch Fabric Modules

The switch fabric stores packets as they come from the network. Up to four switch fabrics can be configured in the system. Each switch fabric is completely independent and supports N+1 redundancy to prevent service disruption.

A single switch fabric module can support combinations of DS3s, OC-3s/STM-1s, OC-12s/STM-4s, Gigabit Ethernet, and OC-48s/STM-16s (1 port) line cards in all 16 slots. A second switch fabric is optionally added for redundancy.

Three switch fabric modules are required to support the 1-port OC-192/STM-64 and the 4-port OC-48/STM-16 cards. Because only three are needed, a fourth fabric provides redundancy. The switch fabric modules have LEDs on the front panel indicating boot status, and fan and PDU faults. See the section on the status LEDs in the *NX64000 Troubleshooting Guide* for more information.

## SONET/SDH Timing/Alarm Modules

Two SONET/SDH Timing/Alarm Modules (STA) (also referred to as the Clock Timing Adapter (CTA)) are installed in the back of the chassis. The second module is for redundancy.

The STA offers a Stratum 3 clock source that is used for high performance synchronization. The interface can be set to use either its internal chassis clock or to synchronize with an external device (a line or external clock). The default chassis clock setting is internal. See the *NX64000 Command Reference* for more information.

## Route Control Processor (RCP)

The RCP processes protocols, computes routing tables, and manages and updates the line card forwarding tables, signaling and control protocols, and SNMP. The NX64000 IP Core Router supports up to two RCPs. The second RCP is for redundancy.

The RCP has an EIA-232 service interface port for console command and diagnostic access, a 10/100 Base-T Ethernet port for system management, and a PCMCIA card reader that accommodates the PCMCIA card (see the **“PCMCIA Card”** section below).

When the system has two RCPs, the primary acts as the RCP for the system while the secondary periodically checks on the status of the primary. If the secondary determines that the primary has failed, it takes over and reboots the entire system, including the line cards. After the system reboots, the secondary RCP becomes the primary RCP. The failed RCP must be replaced; see the *NX64000 Troubleshooting Guide* for detailed instructions on removing and replacing field replaceable units.

The primary RCP is also responsible for copying files from the PCMCIA card on the primary RCP to the PCMCIA card on the secondary RCP. See the section on “[Synchronizing the RCP files](#)” for instructions on copying files from the primary RCP to the secondary RCP.

## PCMCIA Card

The PCMCIA is a standard 128 MB PC card. The PCMCIA card contains the operating system, the routing protocol code and configuration files. The main file on the PCMCIA card is the `rver.tar`, where `ver` indicates the version of the NX-IS software. The `rver.tar` file contains all the files that are necessary for booting.

## Synchronizing the RCP files

In a dual RCP configuration, when the primary RCP fails, the system reboots. The RCP functioning as secondary before the reboot becomes the primary RCP upon restart. Since each RCP has its own PCMCIA card, the images must be identical on both RCPs. The `copy sync` command synchronizes either a single file or the contents of the whole PCMCIA card between the primary and secondary RCPs.

To backup the entire contents of the PCMCIA card, in case the original card fails, run the `copy sync all` command. For example:

```
nx# copy sync all
```

Executing the `copy sync all` command reformats the PCMCIA card in the secondary RCP, overwriting all the files on the card. Although this command means “duplicate the card”, it does not copy the following:

- The `.del` files, the deleted files
- The “inet on ethernet” boot parameter

► You must reboot the secondary RCP when you run the `copy sync all` command to save the changes to the secondary RCP.

The following example reloads the secondary RCP controller.

```
nx# cards
nx(cards)# reload rcp
```

## Upgrading Software

For software updates, you normally receive a new `rver.tar` file. This file may be downloaded to the PCMCIA card in the primary RCP or provided on a new PCMCIA card. To copy the updated image file to the PCMCIA card on the secondary RCP use the `copy sync file` command. For example:

```
nx# copy sync file /pcmcia0/releases/current/r170.tar
```

Alternatively, you can run the `copy sync all` command, to copy everything from the primary PCMCIA card to the secondary PCMCIA card.

- To run `copy sync` commands, the capacity of the PCMCIA card in the secondary RCP must be of greater or equal size to the PCMCIA card in the primary RCP.

## Understanding the Inet on Ethernet Address

You assign an IP address to the management Ethernet port (Ethernet0) on the RCP. This IP address and the internal address set for RCP (inet on ethernet) cannot be set to the same value. The inet on ethernet value is preset to 10.0.100.74:ffffc000. See [Chapter 11, “Internet Protocol Configuration”](#) for information on changing the IP address for the management Ethernet port.

When two RCPs are installed in the chassis, The Ethernet0 IP address should be the same on RCP0 and RCP1, but cannot be the same as the IP address assigned to inet on ethernet (the internal address) on RCP0 or RCP1. [Table 7-4](#) summarizes the Ethernet0 and inet on the ethernet addressing using an example address for Ethernet0 of 10.10.10.10.

**Table 7-4. Ethernet0 and inet on ethernet addressing**

RCP0	RCP1	Comment
Ethernet0 10.10.10.10	Ethernet0 10.10.10.10	The Ethernet0 IP address should be the same on RCP0 and RCP1. The Ethernet0 IP address cannot be the same IP address as the inet on ethernet IP address.
inet on ethernet 10.0.100.74	inet on ethernet 10.0.100.74	The inet on ethernet IP address should be the same on RCP0 and RCP1. The inet on ethernet IP address cannot be the same IP address as the Ethernet0.  With the inet on the ethernet IP address the same for RCP0 and RCP1 you may see a duplicate IP address message at startup. Ignore the messages; the system will function normally.

## Input/Output Cards

The Input/Output Processor (IOP) is made up of a processor board and a daughter board (IOD). The daughter board gives an individual card its *personality*. For example, a POS card and an ATM card both share the same processor (IOP), but their daughter cards are different. Together, the IOP and IOD are a single field replaceable unit (FRU), typically referred to as an IOP. The daughter cards cannot be interchanged individually.

The IOA provides physical network connectivity. Removing and replacing an IOA affects only the traffic on that IOA; it does not affect the operation of the rest of the system. The IOA is a separate field-replaceable unit from the IOP. Each IOA connects to an IOP over the system's backplane. The IOP unit inserts into the front of the chassis while the IOA goes into the back. [Table 7-5](#) lists the IOP-to-IOA relationship.

**Table 7-5. IOP to IOA Relationship**

I/O Processors	I/O Adapters
OC-3c/STM-1 ATM, OC-12c/STM-4 ATM	<ul style="list-style-type: none"> <li>8-port OC-3c ATM</li> <li>2-port OC-12c ATM</li> </ul>
OC-3c/STM-1 POS, OC-12c/STM-4 POS	<ul style="list-style-type: none"> <li>8-port OC-3c POS</li> <li>4-port OC-12c POS</li> <li>8-port DS3 Frame POS</li> </ul>
OC-48c/STM-16 POS (for the 1-port IOA)	1-port OC-48c POS
OC-48c/STM-16 POS (for the 4-port IOA)	4-port QOC-48c POS
OC-192C/STM-64 POS	1-port OC-192 POS
Gigabit Ethernet	2-port Gigabit Ethernet

The *NX64000 Installation Guide* describes the system hardware modules and provides a list of field-replaceable units and associated order numbers.

► Three Switch Fabrics are required for the 4-port QOC-48 and the OC-192.

## Optical Interface Specifications

The system supports Standard SONET/SDH features. The table below provides the optical interface specifications.

**Table 7-6. Line Cards Signal Levels**

Line Card	Min. Output Power (dBm)	Max. Output Power (dBm)	Min. Input Power (dBm)	Max. Input Power (dBm)
8-port OC-3c ATM Short Reach, Multimode Maximum Distance: 2 km	-23.5	-14	-30	-14
8-port OC-3c POS Short Reach Multimode Maximum Distance: 15 km	-15	-8	-31	-14
2-port OC-12c ATM Intermediate Reach, Single-mode, Maximum Distance: 15 km	-15	-8	-31	-8

**Table 7-6. Line Cards Signal Levels**

Line Card	Min. Output Power (dBm)	Max. Output Power (dBm)	Min. Input Power (dBm)	Max. Input Power (dBm)
4-port OC-12c POS Intermediate Reach, Single-mode Maximum Distance: 15 km	-15	-8	-31	-8
1-port OC-48c POS Short Reach, Single-mode Maximum Distance: 2 km	-10	-3	-18	0
4-port OC-48c POS Short Reach, Single-mode Maximum Distance: 2 km	-10	-3	-18	-3
1-port OC-192c POS Short Reach, Single-mode, Maximum Distance: 12Km	-6	-0	-11	-1
1-port OC-192c POS Intermediate Reach, Single-mode Maximum Distance: 40 km	-1	2	-14	-3
2-port Gigabit Ethernet Short Reach, Multimode Maximum Distance: 500 meters	-9.5	-4	-17	0

See the *NX64000 Installation Guide* for information about SONET/SDH features.



## Working with System Cards

The following table lists the card-related commands supported by the system. Each listing indicates the general use for the command. Some of the commands are used and described in the configuration examples that follow. All the listed commands are documented in the *NX64000 Command Reference*.

**Table 7-7. Card-related Command Usage**

Command	Configuration	Monitor	Activation	Deactivation	Troubleshoot
<code>cards</code>	✓	✓	✓		
<code>cpu-monitoring</code>					✓
<code>failurecount</code>					✓
<code>reload</code>			✓		
<code>show</code>		✓			✓
<code>show buffers swfab-buffers</code>		✓			✓
<code>show environment</code>		✓			✓
<code>show process</code>		✓			✓
<code>shutdown</code>				✓	
<code>threshold</code>					✓
<code>unload</code>				✓	

The cards-related `show all` command displays the status of system cards. The command output displays the type of card in each slot, the status of a card—booting, quiescent, operational, reset, or failed—the number of interfaces configured on an IOP, the number of times the card failed since it was loaded, and how many times a card can fail before a manual reset is required. For example:

```
nx(cards)# show all
```

Slot	Type	State	No. Ifs	Failed Count	Failed Thresh
IOC: 0	OC12c-2 ATM	Operational	2	0	2
IOC: 1	OC192c-1 POS	Operational	1	0	2
IOC: 2	OC48c-1 POS	Operational	1	0	2
IOC: 4	OC12c-2 ATM	Operational	2	0	2
IOC: 6	OC3c-8 ATM	Operational	8	0	2
IOC: 8	unknown	Reset	0	0	2
IOC:12	OC192c-1 POS	Operational	1	0	2
IOC:14	unknown	Reset	0	0	2
SFC: 0	Switch Fabric	Operational	0	0	2
SFC: 1	Switch Fabric	Operational	0	0	2
SFC: 2	Switch Fabric	Operational	0	0	2
SFC: 3	Switch Fabric	Operational	0	0	2
TAC: 0	Clock Timing Adapter	Operational	0	0	2
TAC: 1	Clock Timing Adapter	Operational	0	0	2

```
nx(cards)#
```

## Setting the Threshold

The number in the `Failed Thresh` column represents the number of times a card can reload before the system locks the card. The default is 2 (indicating that on the third failure, the system locks the card). If set to 0, the system does not lock the card based on number of times the card reloads. Run the `threshold` command to change this value. For example, enter the following to set the failure threshold to 3 for the IOC in slot seven:

```
nx# cards
nx (cards)# threshold ioc 7 3
```

## Setting the Failurecount

Use the `failurecount` command to reset the card failure count, the number of times a specific card reloads (see the `Failed Count` column in the `show all` output above). Resetting the failure count to zero prevents it from reaching the card failure threshold (set with the `threshold` command). For example, enter the following to set the failure count to 0 on the IOC in slot seven:

```
nx# cards
nx (cards)# failurecount ioc 7 0
```

## Deactivating and Reactivating the Cards

You can deactivate or reset system cards.

Enter the `shutdown` command to perform an orderly shutdown of a card. The `shutdown` command resets the card and places it in a quiescent state. In this state, the card can be removed from the switch or reactivated. The `no` form of the `shutdown` command brings an administratively disabled interface back up.

The following example shuts down and then reloads interface pos15/0.

To shut down the POS interface in slot 15:

```
nx# cards
nx(card)# shutdown IOC 15
```

To activate a card that is in quiescent state:

```
nx# cards
nx(card)# no shutdown IOC 15
```

The following example resets the IOP in slot 12:

```
nx# cards
nx(cards)# reload IOC 12
```

The **unload** command holds the specified card in a reset state (performs an orderly shutdown but does not reboot). The following example sets the STA module (TAC is the card-type argument for the SONET Timing/Alarm Module) in slot 0 off-line:

```
nx# cards
nx(cards)# unload TAC 0
```

► See the *NX64000 Troubleshooting Guide* for detailed instructions on activating and deactivating the cards.

## Configuring Interfaces

You perform many system configuration tasks on a per-interface basis. The interface type determines the commands available for the interface.

The system supports a maximum of 128 interfaces (16 IOA cards x 8 ports per card). The types of cards installed on the system determine the type and number of interfaces that you can configure.

The system supports the following types of interfaces:

- POS—The Packet over SONET interface.
- ATM—The Asynchronous Transfer Mode interface.
- Serial—The DS3 interface.
- Gigabit Ethernet—Gigabit Ethernet
- Ethernet—Ethernet IEEE 802.3 management interface.
- Loopback—The virtual (software only) loopback interface that emulates an interface that is up continuously.
- MPLS (tunnel)—The logical interface for initiating and terminating MPLS label-switched paths
- Null interface—The interface used to discard unwanted packets.

When you enter an interface name at the command line for POS, ATM and Gigabit Ethernet Interfaces you use the format typeslot/port. The following table lists the interface types and provides example formats.

**Table 7-8. Interface Name Entry Format**

Encapsulation	Card Type and Available Ports	Example
ATM	8-port OC-3c ATM	<code>atm6/0</code>
<ul style="list-style-type: none"><li>• Frame Relay</li><li>• PPP</li></ul>	8-port OC-3c POS	<code>pos5/0</code>
<ul style="list-style-type: none"><li>• Frame Relay</li><li>• PPP</li></ul>	8-port DS3	<code>serial17/4</code>
ATM	2-port OC-12c ATM	<code>atm6/0</code>
<ul style="list-style-type: none"><li>• Frame Relay</li><li>• PPP</li></ul>	4-port OC-12c POS	<code>pos5/3</code>
<ul style="list-style-type: none"><li>• Frame Relay</li><li>• PPP</li></ul>	1-port OC-48c POS	<code>pos5/0</code>
<ul style="list-style-type: none"><li>• Frame Relay</li><li>• PPP</li></ul>	4-port OC-48c POS	<code>pos8/2</code>
<ul style="list-style-type: none"><li>• Frame Relay</li><li>• PPP</li></ul>	1-port OC-192c POS	<code>pos8/0</code>
Ethernet	2-port Gigabit Ethernet	<code>gigabitethernet15/0</code>

## Ethernet, Loopback, Tunnel, and Null Interfaces

The syntax used to configure Ethernet, loopback, tunnel, and null interfaces, depends on the type of interface.

Two broadcast interfaces are predefined on the system `loopback0` and `ethernet0`. The `loopback0` interface is a software-defined interface, used for routing and testing; `ethernet0` is the management Ethernet interface. Ethernet0 can reach the network to download configuration files and software upgrades.

Use null interfaces to discard unwanted packets. Null0 is the only valid null interface name. Use the format *typeport* for the Ethernet, loopback, and null interfaces. For example:

```
loopback0
ethernet0
null0
```

Logical interfaces, called tunnel interfaces, transmit traffic on the ingress and egress routes of label-switched paths. You configure tunnel interfaces as you would system interfaces, but you must also bind them to physical interfaces to transmit traffic. See [Chapter 14, “Multi-protocol Label Switching \(MPLS\) Configuration”](#) for more information.

Use the format *tunnel tunnel-number* to configure a tunnel interface. For example:

```
tunnel100
```

## Quality of Service for Traffic Management

Quality of Service (QoS) guarantees forwarding services for specified types of traffic. It prioritizes traffic and allocates bandwidth to give preferred treatment to priority traffic while decreasing jitter and delay. You configure QoS on a per-interface basis. For more information about using QoS, see the [Chapter 13, “Quality of Service Configuration for Traffic Management,”](#).

- QoS is not available on Gigabit Ethernet interfaces

## Interface Configuration Commands

The interface-related commands affect or display the state of a specified interface. [Table 7-9](#) lists useful commands for configuring an interface. See the *NX64000 Command Reference* for a detailed description of the commands.

**Table 7-9. Interface Configuration Command Usage**

Command	Interoperability	Basic Interface Configuration	Verification	Clear Counters	Gigabit Ethernet	Interface Management
bandwidth		✓				
clear counters				✓		
clock-source						✓
description		✓				
encapsulation		✓				
fcs	✓					
feac						✓
interface		✓				
laser						✓
load-interval						✓
mtu						✓
payload-scrambling	✓					

**Table 7-9. Interface Configuration Command Usage**

Command	Interoperability	Basic Interface Configuration	Verification	Clear Counters	Gigabit Ethernet	Interface Management
pos flags c2	✓					
pos-scrambling atm	✓					
show ds3			✓			
show interfaces			✓			
show sonet			✓			
shutdown						✓
arp					✓	
arp-timeout					✓	
autonegotiation					✓	

## Basic Interface Configuration Tasks

The following sections show how to perform various interface configuration tasks. These tasks include:

- Creating the interface
- Setting the encapsulation type
- Setting operating compatibility
  - Setting the CRC values (optional)
  - Setting the C2 byte (optional)
  - Managing scrambling (optional)
  - Configuring framing (ATM only, optional)
- Configuring subinterfaces
- Configuring Gigabit Ethernet interfaces
  - Adding static MAC addresses to the ARP table
  - Setting the MTU
  - Setting autonegotiation
- Verifying the configuration

## Creating the Interface

To configure an interface you enter the type and slot number of the interface you want to configure and parameters specific to a particular interface, for example, an IP address. To configure an interface with an IP address, see [Chapter 11, “Internet Protocol Configuration.”](#)

Enter the interface in the proper format *for the type of interface*. See [Figure 7-8. “Interface Name Entry Format” on page 7-12](#). You must specify the encapsulation type when you configure a permanent virtual circuit for ATM. For more information, see [Chapter 9, “Asynchronous Transfer Mode \(ATM\) Configuration”](#). The following example enters the interface configuration mode for an ATM interface in slot 2:

```
nx# configure terminal
nx(config)# interface atm2/0
```

When configuring either a POS or serial interface, you must set an encapsulation method of Frame Relay or Point-to-Point Protocol (PPP). No traffic is forwarded across the interface until the encapsulation method is set. For more information configuring encapsulation, see [Chapter 8, “Frame Relay Configuration”](#) and [Chapter 10, “Point-to-Point Protocol Configuration”](#). The following example configures a POS interface on slot number 5, port number 0 and sets the encapsulation method to Frame Relay:

```
nx# configure terminal
nx(config)# interface pos5/0
nx(config-if)# encapsulation frame-relay
```

## Setting Operating Compatibility

To assure the system interoperates with other devices, the CRC value, C2 byte, scrambling and the framing settings must be set to the same values on both sides of the link. This section provides the system default values and the configuration methods.

### Setting the Cyclic Redundancy Check (CRC) Values

The CRC is an algorithm that checks for transmission errors for OC-12 POS, OC-48 and OC-192 POS interfaces. The system has a default CRC value of 32, but it can be set to 16.

The following example sets the CRC to 16 on interface pos5/0 and reboots the card.

```
nx# configure terminal
nx(config)# interface pos5/0
nx(config-if)# fcs 16
nx(config-if)# exit
nx(config)# exit
nx# cards
nx(cards)# reload IOC 12
```

### Setting the C2 Byte

SONET devices use the SONET path-label byte, or C2 byte, to identify the type of traffic being sent over the link. The C2 byte must be set to the same value on both sides of the link to interoperate with other POS devices. The default for the system is 0x16 (22) for POS or 0x13 (19) for ATM.

You must enter the value of the path-label byte, in hexadecimal digits. For example, to get the byte value 22, enter 0x16.

```
nx# configure terminal
nx(config)# interface pos5/0
nx(config-if)# pos flags c2 0x16
```

### Disabling Scrambling

The system enables scrambling on a POS interface by default. You can disable scrambling to interoperate with other devices. The following example disables payload scrambling on interface pos5/0:

```
nx# configure terminal
nx(config)# interface pos5/0
nx(config-if)# no payload-scrambling
```

### Configuring Framing

SONET and SDH are a set of standards for synchronous data transmission over fiber optic networks. Framing enables the correct physical-layer standard for SONET-based interfaces, either North America SONET or rest-of-the-world SDH. By default, system interfaces use SONET framing. The following example configures the connection for European standards, SDH, on interface atm2/0:

```
nx# configure terminal
nx(config)# interface atm2/0
nx(config-if)# framing sdh
```

### Configuring Subinterfaces

Subinterfaces are logical interfaces on a physical port. ATM PVCs and Frame Relay DLCIs can be configured on subinterfaces. Frame Relay subinterfaces support only routing. For information about configuring Frame Relay DLCIs on subinterfaces, refer to [Chapter 8, “Frame Relay Configuration”](#). For information about configuring ATM PVCs, see [Chapter 9, “Asynchronous Transfer Mode \(ATM\) Configuration.”](#)

► The Gigabit Ethernet cards do not support subinterfaces.

### Configuring Gigabit Ethernet Interfaces

The Gigabit Ethernet cards have specialized settings to configure the interfaces and parameters. This section describes the use of these Ethernet commands.

#### Adding Static MAC Addresses to the ARP Table

The Gigabit Ethernet interface requires a hardware MAC address be associated with an IP address, in order to transmit a packet to its destination. Run the `arp` command to add static MAC addresses to the Address Resolution Protocol (ARP) table. The ARP table enables the system to translate between IP addresses and physical (MAC) addresses. When a host wants to communicate with a peer, the host looks in the ARP table for the peer address.



The following example adds an entry to the ARP table; associating IP address 192.0.2.100, to MAC address 00:00:0d:2c:06:0c on interface gigabitethernet12/0. It then sets the time limit that ARP entries remain in cache to 1200 seconds (20 minutes):

```
nx# configure terminal
nx(config)# interface gigabitethernet12/0
nx(config-if)# arp 192.0.2.100 00:00:0d:06:2c:0c static
nx(config-if)# arp-timeout 1200
```

Use the `show ip arp` command to verify the entry is in the ARP table.

### Setting the MTU

Gigabit Ethernet interfaces provide support for extended frames, or jumbo packets, larger than the 1500 byte size of standard Ethernet. This eliminates the need to do fragmentation of packets larger than 1500 bytes and makes efficient use of larger payload sizes. Since the overhead (size and processing time) for larger packets is the same as the overhead for small packets, the use of these extended packets can increase data throughput across devices overloaded by small packet overhead.

Valid MTU byte values are from 64 to 8000 bytes. The default Gigabit Ethernet MTU value is 1500 bytes. If a Gigabit Ethernet interface is to support jumbo packets, you must specify an appropriate MTU value when you configure the logical interface. For example:

```
nx# config terminal
nx(config)# interface gigabitethernet7/0
nx(config-if)# mtu 8000
nx(config-if)# exit
```

### Setting Autonegotiation

When a connection is established to a network device, autonegotiation detects the various performance-compatible modes in the device on the other side of the link, announces its own abilities and automatically configures the highest performance mode of interoperation.

Once the highest performance common mode is determined, autonegotiation passes control to the appropriate device and becomes transparent until the connection is broken. Autonegotiation is enabled by default.

Change both sides of the link to disable autonegotiation. For example:

```
nx(config)# interface gigabitethernet7/0
nx(config-if)# no autonegotiation
```

Run the `show interface` command to confirm the change was made.

► The line protocol is always up on a Gigabit Ethernet interface.

## Verifying Cards and Interface Configuration

Run the following **show** commands to verify which parameters are configured before making any configuration changes. The following table lists the type of information you can view from each **show** command:

**Table 7-10. Card and Interface Commands for Verifying Configuration**

Action	Command
Displays configuration and status for your system cards. This command is executed from the <code>nx(cards)#</code> prompt.	<b>show</b>
Display statistics for a specified DS3 serial interface.	<b>show ds3</b>
Display configuration settings, statistics, and laser settings for all or specified interfaces configured on the system.  Note: The line protocol is always up on a Gigabit Ethernet interface.	<b>show interfaces</b>
Display interface parameter and configuration settings currently in running memory. Compare the output from this command and the <b>show startup-config</b> command to isolate configuration changes made since the last time the running configuration was saved to the startup configuration.	<b>show running-config</b>
Display the contents of the Address Resolution Protocol (ARP) table.	<b>show ip arp</b>
Display statistics and errors pertaining to the SONET signals on a specified interface.	<b>show sonet</b>
Display the parameter and configuration settings saved to the startup configuration file. Compare the output from this command and the <b>show running-config</b> command to isolate configuration changes that were made since the last time the running configuration was saved to the startup configuration.	<b>show startup-config</b>

# Frame Relay Configuration

Frame Relay is a synchronous network that evolved from the Integrated Services Digital Network (ISDN) and the X.25 protocol suite. Frame Relay transfers information over a WAN by implementing permanent virtual circuits (PVCs) which are logical circuits that appear as dedicated point-to-point connections. Packets, known in Frame Relay as frames, are transferred from endpoint to endpoint over the PVCs. Each PVC is identified by a data link connection identifier (DLCI). By having multiple PVCs a system can simultaneously communicate with multiple sites. Because the Frame Relay network is made up of virtual/logical rather than physical connections between endpoints, it is most often represented as a cloud to indicate that connections are not fixed but set up as needed. Bandwidth on these logical connections/paths is allocated on a virtual circuit (VC) basis.

Since Frame Relay passes frames from element to element within the network without error detection, the transfer of information from source to destination is very fast. This is in contrast to X.25. In addition to speed, Frame Relay provides a number of circuit parameters, such as Burst Size and Committed Information Rate (CIR), to define the Quality of Service (QoS) parameters.

## Key Features

The NX-IS Frame Relay implementation supports the following features:

- Frame Relay switching
- Frame Relay routing
- Link management interface (LMI) protocol
- FRF2.1/Link access procedure for Frame Relay (LAPF)
- Quality of Service

## Technology Concepts

The following sections give an overview of concepts that are basic to understanding Frame Relay:

- Virtual circuits
- Data link connection identifier (DLCI)
- Local management interface (LMI)
- Link Access Procedure for Frame Relay (LAPF)
- Encapsulation
- Frame Relay routing and switching
- Frame Relay dynamic and static address mapping
- Quality of service and traffic management
- Cyclic redundancy check (CRC)
- Frame Relay network interfaces/devices

## RFCs and Standards

The NX implementation of Frame Relay is based in part on the following RFCs and standards:

Standard	Title
RFC 1604	Frame Relay Service
RFC 2115	Frame Relay Management Information Base
RFC 1490	Multiprotocol Interconnect over Frame Relay
ANSI T1.617 Annex D	Digital Subscriber Signaling System No. 1 (DSS1) — Signaling Specification for Frame Relay Bearer Service
ITU-T Q.933 Annex A	Integrated Services Digital Network (ISDN) Digital Subscriber Signalling System No. 1 (DSS 1) — Signalling Specifications for Frame Mode Switched and Permanent Virtual Connection Control and Status Monitoring
FRF2.1	Frame Relay Network-to-Network Interface (NNI) Implementation Agreement

## Virtual Circuits

The NX64000 implementation uses the concept of virtual circuits (VCs) to transfer data between two endpoints. VCs are bidirectional, software-defined data paths/connections between two network endpoints. The endpoints can be IP assigned devices or Frame Relay endpoints known as DTEs (data termination equipment). Each endpoint is assigned one or

more unique identifiers, known as Data-Link Connection Identifiers (DLCIs). As shown in **Figure 8-5**, virtual circuits can extend through any number of intermediate data communication equipment (DCE) or network-to-network interface (NNI) devices within the Frame Relay network.

Frame Relay supports both permanent virtual circuits and switched virtual circuits (PVCs and SVCs) but the NX-IS implementation supports only permanent virtual circuits.

PVC are defined as logical links between endpoints set up via the command line interface (CLI) and network management system as required, for example, when a new endpoint is connected to the network or when existing endpoints need to talk to one another. Although PVCs act as fixed paths, the actual path through the network can be changed. However, the endpoints of the circuit do not change. In this way, the PVC acts like a dedicated point-to-point circuit.

## Data Link Connection Identifier (DLCI)

Each source and destination device port in the Frame relay network is assigned a DLCI number which corresponds to the address of that particular source or destination device.

DLCIs need not be unique in a Frame Relay network because they have only local significance. As shown below, DLCI numbers that refer to the same connection can change within the connection (DLCI 40 and DLCI 70 in NX2) but need to be the same between devices (DLCI 40 between NX1 and NX2 and DLCI 70 between NX2 and NX3).



**Figure 8-1. Data Link Connection Identifier**

DLCI supports either Frame Relay switching or Frame Relay routing.

## Local Management Interface (LMI) Protocol

LMI is a protocol that performs periodic status updates on the link and PVCs configured on it to ensure that the link is operating correctly. This is especially useful when there is no traffic on the link. The status updates provide the following type of information for both the network and the user end:

- Keepalives
- Status inquiries sent and received
- Errors reported
- Asynchronous updates sent and received

For more information on the status updates, refer to the `show frame-relay lmi` command.

The NX-IS implementation supports the LMI specifications and LMI types listed in the following table:

LMI Specification and Type	On DLCI	Packet Flow	Applies to
ANSI T1.617 Annex D (this is the default)	DLCI 0	Unidirectional and bidirectional	U.S. UNI and NNI
ITU-T Q.933 Annex A	DLCI 0	Unidirectional and bidirectional	International UNI and NNI
Annex2-1	DLCI 0	Unidirectional and bidirectional	Any NNI
Cisco (Stratacom)	DLCI 1023	Unidirectional	Any UNI
None (No LMI type is specified. User must configure all DLCIs.)	N/A	N/A	International UNI and NNI

The LMI type can be specified on a per-interface basis with the `frame-relay lmi-type` command or, if the `frame-relay auto-lmi` command is executed, the switch learns the LMI type at the other end of the link and sets itself appropriately. The command `frame-relay auto-lmi` can be executed when a given POS interface is configured to support Frame Relay encapsulation and the interface type is either DCE or NNI.

- Before executing the `frame-relay lmi-type` command, disable auto LMI sensing. This prevents possible race conditions in the event that the auto-LMI sensing option is set at both ends of the link.

The frequency of the LMI status updates and the type of information provided is determined by certain LMI characteristics, such as duration of keepalive, error threshold, and polling interval, which are set to default values. However, the user can override these default values by using the following commands if, for example, the system at the user end must be re-configured to match the speeds of the system at the other end of the link.

- `frame-relay lmi-n391dte`
- `frame-relay lmi-n392dc`
- `frame-relay lmi-n392dte`
- `frame-relay lmi-n393dce`
- `frame-relay lmi-n393dte`
- `frame-relay lmi-t392dce`

See the example in **“Local Management Interface (LMI)”** on page 8-16.

## Link Access Procedure for Frame Relay (LAPF)

LAPF makes possible more efficient frame transmission. It is enabled by executing the `frame-relay lmi-type` command with the `annex2-1` argument. LAPF provides a reliable transmission layer, therefore, annex2-1 is mode event driven as compared to periodic updates in all other LMIs.

Enabling the LASPF functionality also enables the following LAPF-related commands which, if required, allow you to configure LAPF parameters on a specific NNI interface:

- `frame-relay annex2-1`
- `frame-relay lapf k`
- `frame-relay lapf n200`
- `frame-relay lapf n201`
- `frame-relay lapf n201`
- `frame-relay lapf t200`
- `frame-relay lapf t203`

Refer to “[LAPF Configuration](#)” on page 8-17 for examples of LAPF command execution.

## Encapsulation

Encapsulation is the process of enclosing or wrapping frames or packets of one type of protocol within another type of protocol. Since a Frame Relay network accepts only frames that are formatted for Frame Relay, the end device, known as a Frame Relay Access Device (FRAD) is responsible for wrapping protocol-specific frames into Frame Relay frames so that they will be accepted by the Frame Relay network for transmission to their destination.

## Frame Relay Routing and Switching

The basic description of Frame Relay routing and switching may be stated as follows:

- In routing, the decision of where to send the frame is made at the IP layer (layer 3) by looking at destination address. The link from source to destination must be established before any data transfer can take place. The routing table is maintained by the routing protocol, which queries the status of neighbor routers to update the tables.
  - In Frame Relay switching, the decision is made at the interface layer (layer 2). In this case, the frame is switched from the input DLCI on the router's interface to an output DLCI before being forwarded to its destination.
- 
- To allow both routing and switching on a POS or serial interface, the subinterfaces are used exclusively for routing and the main interfaces for switching.
  - You must configure IP addresses for subinterfaces (for example, `pos2/0.1`) but not for main interfaces (for example, `pos2/0`).

## Frame Relay Dynamic and Static Address Mapping

Frame Relay provides two methods for mapping an IP protocol address to a DLCI, dynamic and static.

### Dynamic Address Mapping

Dynamic address mapping is achieved with the Inverse Address Resolution Protocol (Inverse-ARP) which, in the NX-IS implementation, is enabled by default for those interfaces on which IP is configured. Inverse-ARP discovers the IP address of a peer connected at the other end of the DLCI. Thus, it provides a physical address (DLCI)-to-IP address association.

When an interface receives the first LMI packet, it discovers which DLCIs are active. The router then sends an inverse-ARP request which contains its own source IP address. Responses to inverse-ARP requests are entered in an IP protocol address-to-DLCI mapping table. This information is then used to supply outgoing traffic with the next hop IP protocol address or DLCI number.

Although inverse-ARP is enabled by default, it can be disabled for any IP protocol address and DLCI pair. This allows the user to have dynamic mapping for some IP protocol address and DLCI pairs and static mapping for others.

If dynamic mapping is disabled, you can enable it by executing the `frame-relay inverse-arp` command.

- To do dynamic mapping, you need inverse-ARP on both systems. You must use static mapping if the router at the other end of the link does not support inverse-ARP.

### Static Address Mapping

Static address mapping in Frame Relay requires that you define the connection between the IP protocol address and the destination DLCI. This connection is made in a Frame Relay map (created with the `frame-relay map` command). A static map associates a specified next hop IP address and the DLCI connecting to the destination.

Static mapping eliminates the need for inverse-ARP requests. Therefore, once you supply a static map, inverse-ARP is automatically disabled on the specified DLCI.

## Quality of Service (QoS) and Traffic Management

Quality of Service generally refers to a network's ability to deliver to all users the necessary system resources at the level specified by the QoS parameters. Therefore by managing system resources and traffic requirements, QoS provides support for networks that participate in the service-level agreements between service providers and their clients. For an example of QoS and traffic management configuration, refer to [“Quality of Service \(QoS\) and Traffic Management Configuration” on page 8-18](#).

For a detailed description of QoS implementation in NX-IS, refer to [Chapter 13, “Quality of Service Configuration for Traffic Management”](#).



## Cyclic Redundancy Checking (CRC)

CRC is an error checking mechanism used by Frame Relay and implemented in the NX hardware. With this method the originating device calculates the CRC value based on the content of the frame before it is transmitted. The destination device recomputes the value of the frame and compares it to the original value. If the values match, the frame is OK, otherwise it is discarded.

Error correction is a function of higher layer protocols running on top of Frame Relay. Thus, by implementing error checking rather than error correction, Frame Relay reduces network overhead.

Since CRC checking is done in the hardware, no performance deterioration results from it.

## Frame Relay Network Interfaces/Devices

A Frame Relay network supports three types of interfaces or devices at the end of the link:

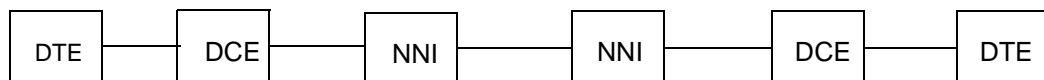
- **Data Terminal Equipment (DTE) interface:** behaves like the interface of a router and must be connected to a DCE, such as a Frame Relay switch.
- **The Data Communication Equipment (DCE) interface:** behaves like the interface of a switch; routers (DTE) can be connected to it.

The connection between a DTE and a DCE consists of a physical and a link-layer component. The physical-layer interface defines the mechanical, electrical, and functional specifications of the connection while the link-layer defines the protocol that establishes the connection between the DTE and the DCE.

► You cannot have a DTE connected to another DTE or a DCE connected to another DCE. For proper operation, if one end has a DTE connection, the other end must have a DCE connection.

- **Network-to-Network Interface (NNI):** behaves like the intermediate node of a Frame Relay connection. In the NX-IS implementation, the default Frame relay interface is NNI.

The following illustration shows a general Frame Relay network interfaces/devices connection.



## Debugging and Logging Facilities

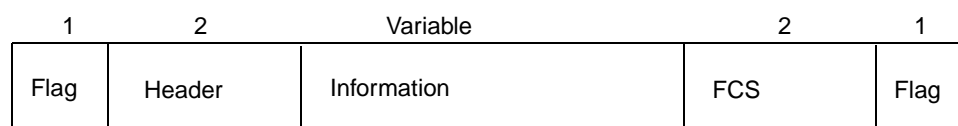
The NX-IS implementation provides debugging and message logging facilities to help you troubleshoot Frame Relay problems. In addition the `frame-relay country-code` and the `frame-relay network-id` commands provide additional information to help isolate the problem to a geographical area. Also when `lmi-type annex2-1` is enabled, the `show frame-relay lmi` command output indicates reasons for DLCI failure. For more information on troubleshooting Frame Relay problems, refer to the *NX64000 Troubleshooting Guide*.

## Frame Relay Frame Formats

The following sections describe the Frame Relay frame, the Frame Relay frame header, and the LMI Frame formats.

### Frame Relay Frame

Frame Relay uses the synchronous HDLC frame format which can be up to 8 kbytes in length. The major difference between the two frame formats is that the Frame Relay frame has no control field. **Figure 8-2** shows the basic format (and the length of each field in 8-bit bytes) of the Frame Relay frame. **Figure 8-3** shows the content of the header portion of the frame.



**Figure 8-2. Frame Relay Frame**

The following descriptions summarize the Frame Relay frame fields.

1. **Flag:** These two fields indicate the start and the end of the Frame Relay frame.
2. **Header:** The header field is 2 to 4 bytes long (see below for more details).
3. **Information field:** This is the actual data (payload) being carried by the frame.
4. **FCS (Frame Check Sequence):** FCS checks the integrity of transmitted data. This value is computed by the source element and verified by the receiving element. If the values match, it indicates a good frame. If the values do not match, the frame is discarded. However, the Frame Relay network components do not deal with these errors. The errors are left to the originator and destination components to deal with.

### Frame Relay Frame Header

**Figure 8-3** gives a more detailed view of the Frame Relay frame header and the length in bits of each field.



**Figure 8-3. Frame Relay Frame Header**

The following descriptions summarize the 2-byte Frame Relay header fields.

1. **DLCI (Data Link Connection Identifier):** The main component of the Frame Relay header is the 10-bit DLCI which indicates the frame destination. (Note that there is no source address in the header.)
2. **EA (Extended Address):** The EA bit indicates whether the extended address capability is being used. If it is, an additional 1 or 2 bytes would be added to the header to allow for the extended addressing scheme.

3. **C/R (Command/Response):** Used in LAPF. Set to 0 to indicate a command and 1 to indicate a response.
4. **Congestion Control (FECN, BECN, and DE bits):** Congestion control is not implemented in the NX-IS Frame-Relay, therefore, these bits are not used.

## LMI Frame Format

Figure 8-4 shows the format of an LMI frame and the length of each field in 8-bit bytes.

1	2	2	Variable	2	1
Flag	Header	Control	Information	FCS	Flag

**Figure 8-4. LMI Frame Format**

The following descriptions summarize the LMI frame fields.

1. **Flag:** These two fields indicate the start and end of a frame.
2. **Header:** LMI messages travel on either DLCI 0 or DLCI 1023 depending on the LMI type.
3. **Control:** Contains zeros. This field is not used.
4. **Information:** This field contains one of the following:
  - A status inquiry message which is a request from a user device for status about a specific interface or all interfaces.
  - A status message which provides the status information in response to the status inquiry message.

For detailed information on the content of the status-inquiry message and the status message, refer to the `show frame-relay lmi` command in the *NX64000 Command Reference*.

5. **Frame Check Sequence (FCS):** Checks the integrity of transmitted data.

## Frame Relay Configuration

This section provides examples of basic Frame Relay configuration tasks. Table 8-1 lists the Frame Relay commands that comprise the current implementation and their application in configuring and/or maintaining a Frame Relay network.

- All commands referenced in this chapter are described in the *NX64000 Command Reference* manual.

**Table 8-1. Frame Relay Command Usage**

Command	Routing	Switching	DLCI Configuration	LMI Configuration	Interoperability	OoS/Traffic Management	Verification and Debug
clear arp-cache	✓						
clear frame-relay-inarp	✓						
frame-relay annex2-1					✓		
frame-relay auto-lmi				✓			
frame-relay bc			✓				
frame-relay be			✓				
frame-relay cir			✓				
frame-relay class			✓				
frame-relay country-code							✓
frame-relay interface-dlci	✓		✓				
frame-relay interface-type	✓	✓					
frame-relay inverse-arp	✓						
frame-relay lapf k				✓			
frame-relay lapf n200				✓			
frame-relay lapf n201				✓			
frame-relay lapf t200				✓			
frame-relay lapf t203				✓			
frame-relay keepalive				✓			
frame-relay lmi-n391dte				✓			
frame-relay lmi-n392dce				✓			
frame-relay lmi-n392dte				✓			

**Table 8-1. Frame Relay Command Usage**

Command	Routing	Switching	DLCI Configuration	LMI Configuration	Interoperability	QoS/Traffic Management	Verification and Debug
frame-relay lmi-n393dce				✓			
frame-relay lmi-n393dte				✓			
frame-relay lmi-t392dce				✓			
frame-relay lmi-type				✓			
frame-relay local-dlci		✓	✓				
frame-relay map	✓		✓				
frame-relay network-id							✓
frame-relay priority						✓	
frame-relay route			✓				
frame-relay switching		✓					
frame-relay traffic-rate						✓	
map-class frame-relay						✓	
show frame-relay class						✓	
show frame-relay lmi							✓
show frame-relay map							✓
show frame-relay pvc							✓
show frame-relay route							✓
show frame-relay summary							✓
show frame-relay traffic							✓

## Basic Frame Relay Configuration Tasks

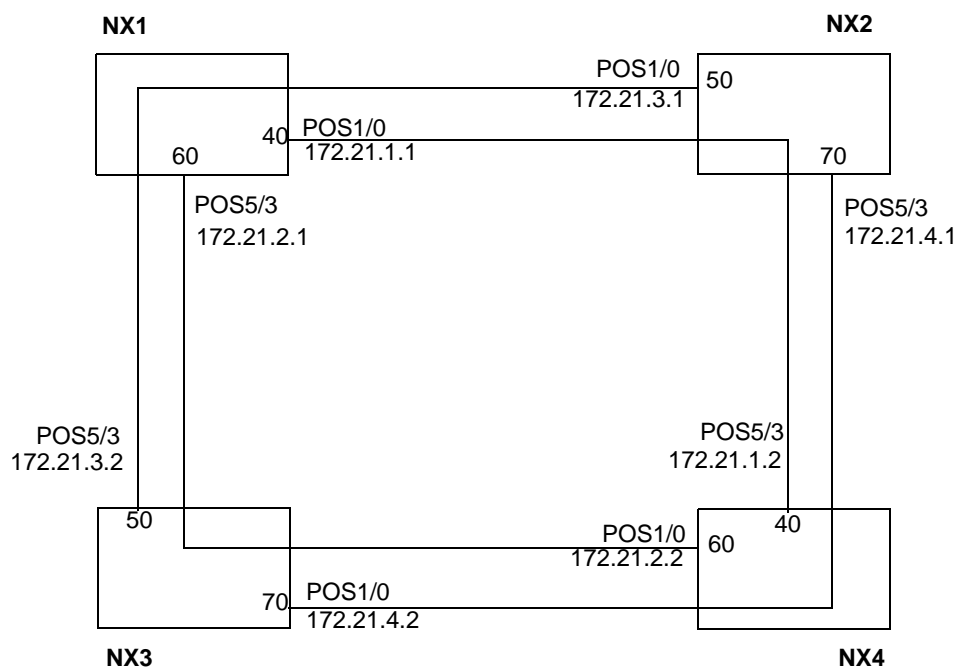
The following sections show how to perform various Frame Relay configuration tasks. The tasks in the basic configuration include:

- Configuring POS interfaces
- Configuring DLCIs
- Configuring interfaces and LMI types
- Creating a Frame Relay map
- Creating a DLCI through a router

Other examples include:

- Configuring address mapping
- Changing LMI statistics values
- Enabling LAPF and changing LAPF configuration parameters
- Configuring QoS and Traffic Management
- Configuring traffic shaping parameters

**Figure 8-5** illustrates a hypothetical Frame Relay network used as the basis for the sample configuration tasks.



**Figure 8-5. Frame Relay Configuration Sample Network**

The illustration shows:

- DLCI 40 going from NX1 through NX2 to NX4.

- DLCI 50 going from NX2 through NX1 to NX3.
- DLCI 60 going from NX4 through NX3 to NX1.
- DLCI 70 going from NX3 through NX4 to NX2.

In this sample network, therefore, the systems do both routing (for example, packets being delivered from NX1 to NX4) and switching (for example, packets going through NX2).

Although the examples indicate packet flow in a clockwise direction, from NX1 to NX4, NX2 to NX3, the network is bidirectional so realistically the flow can also be from NX1 to NX4 and from NX4 to NX1

## Basic Frame Relay Configuration

### Configuring POS1/0 on NX1

```
nx1# configure terminal
nx1(config)# interface pos1/0
nx1(config-if)# encapsulation frame-relay
nx1(config-if)# frame-relay intf-type nni
nx1(config-if)# frame-relay lmi-type ansi
nx1(config-if)# exit
nx1(config)# interface pos1/0.1 point-to-point
nx1(config-subif)# ip address 172.21.1.1 255.255.255.252
nx1(config-subif)# frame-relay interface-dlci 40
nx1(config-subif)# frame-relay map ip 172.21.1.2 40
```

The following table explains the commands used in the example above and below the table on configuring interfaces on NX1.

Configuration Line	Description
<b>configure terminal</b>	Puts the router into terminal configuration mode so you can enter system commands.
<b>interface pos1/0</b>	Assigns the interface type to the interface (in this case, POS for Frame Relay) and enters interface configuration mode. With the interface type defined, you can also create subinterfaces on the interface.
<b>encapsulation frame-relay</b>	Sets Frame Relay as the encapsulation mode. No traffic can be forwarded across the interface until encapsulation is set.

Configuration Line	Description
<b>frame-relay intf-type nni</b>	Specifies the kind of equipment that can be connected to the interface. In the example, execution of this command is redundant because the router sets the interface type to NNI by default. Because the routers in the example do both routing and switching, NNI is the correct interface type. In other cases, the interface type might have to be set to DTE or DCE. Note that: <ol style="list-style-type: none"> <li>1. You cannot have DTEs or DCEs at both ends of the interface. For proper operation, if one end has a DTE connection, the other must have a DCE connection.</li> <li>2. If Frame Relay switching, which is enabled by default, was disabled for some reason using the <code>no frame-relay switching</code> command, you cannot define <code>dce</code> or <code>nni</code> as an interface type nor can you define a frame-relay route unless you re-enable Frame Relay switching.</li> </ol>
<b>frame-relay lmi-type ansi</b>	Enables LMI type <code>ansi</code> to match the LMI type on the destination router.
<b>exit</b>	Returns to the previous prompt level. This is necessary because when configuring an IP address on a Frame Relay link (below), you must create a subinterface and configure the IP address and associated DLCI on the subinterface.
<b>interface pos1/0.1 point-to-point</b>	Creates subinterface 1 on interface <code>pos1/0</code> .
<b>ip address 172.21.1.1 255.255.255.252</b>	Specifies the IP address and the subnet mask for the interface and enables IP on the source interface.
<b>frame-relay interface-dlci 40</b>	Specifies the source DLCI.
<b>frame-relay map ip 172.21.1.2 40</b>	Creates a static ARP entry in the routing table which associates the IP address with DLCI 40.

#### Configure POS5/3 on NX1

```

nx1(config)# interface pos5/3
nx1(config-if)# encapsulation frame-relay
nx1(config-if)# frame-relay lmi-type ansi
nx1(config-if)# exit
nx1(config)# interface pos5/3.1 point-to-point
nx1(config-subif)# ip address 172.21.2.1 255.255.255.252
nx1(config-subif)# frame-relay interface-dlci 60
nx1(config-subif)# frame-relay map ip 172.21.2.2 60
nx1(config-subif)# exit
nx1(config)# frame-relay route pos5/3 50 pos1/0 50

```



```
nx1(config-if)#
```

The use of the `frame-relay route` command is explained in the following table.

Configuration Line	Description
<b>frame-relay route pos5/3 50 pos1/0 50</b>	Configures the static route entry for DLCI 50 through NX1. Identifies pos5/3 50 as the incoming interface and associates pos1/0 50 as the outgoing interface.

The same configuration tasks are performed for the other NX routers but the IP addresses, interfaces, and DLCIs are changed as indicated in [Figure 8-5](#).

## Configuring Dynamic or Static Address Mapping

As stated above, Frame Relay provides two methods for mapping an IP protocol address to a DLCI:

- Dynamic mapping which is achieved with the Inverse Address Resolution Protocol (Inverse-ARP).
- Static mapping which requires the user to define the connection between the IP protocol address and the destination DLCI.

Following are examples of dynamic and static address mapping.

### Dynamic Address Mapping

Because Frame Relay inverse-ARP is enabled by default, dynamic mapping is enabled by default as well. However, if inverse-ARP was disabled on a DLCI using the `no frame-relay inverse-arp` command or by mapping a static address, it can be re-enabled as shown in the following example.

```
nx# configure terminal
nx(config)# interface pos5/3
nx(config-if)# encapsulation frame-relay
nx(config-if)# exit
nx(config)# interface pos5/3.1 point-to-point
nx(config-subif)# ip address 172.21.2.1 255.255.255.0
nx(config-subif)# frame-relay inverse-arp ip 400
```

You can clear the ARP and Inverse ARP caches using the following commands at the global (nx#) level.

```
clear frame-relay arp
```

and

```
clear frame-relay-inarp
```

## Static Address Mapping

You must use static mapping if the router at the other end of the link does not support inverse-ARP. The following example maps the destination IP address 172.21.2.2 to DLCI 60 on interface pos5/3.1.

```
nx# configure terminal
nx(config)# interface pos5/3
nx(config-if)# encapsulation frame-relay
nx(config-if)# exit
nx(config)# interface pos5/3.1
nx(config-if)# frame-relay map ip 172.21.2.2 60
DLCI 60 added on interface pos5/3.1
```

## Local Management Interface (LMI)

Frame Relay LMI commands listed in [Table 8-1 on page 8-10](#) check the status of the link at regular intervals. The commands:

- Establish the keepalive sequence which is a polling procedure for passing status information between user and network.
- Set the full status polling counter to obtain status on new and existing PVCs when interface is configured as DTE or NNI.
- Set the LMI error threshold for DCE, DTE, or NNI interfaces. When the set number of errors has occurred, the connection is considered to be down.
- Set the monitored event count for DCE, DTE or NNI interface. When the set number of errors has occurred, the link is considered to be down.
- Configure the polling verification timer for a DCE or NNI interface.

The commands are enabled by default and only need to be executed if you wish to change the default values.

The following example shows how to change the keepalive interval and the full status polling counter for a PVC on an interface.

```
nx# configure terminal
nx(config)# interface pos1/0
nx(config-if)# encapsulation frame-relay
nx(config-if)# frame-relay lmi-type ansi
nx(config-if)# frame-relay keepalive 6
nx(config-if)# frame-relay lmi-n391dte 10
```

The following table explains the commands used in the above example:

Configuration Line	Description
<b>frame-relay lmi-type ansi</b>	Sets the LMI type before the parameters can be changed. You may have to disable auto-lmi before you can set the LMI type. In each case, the system prompts you with an error message.
<b>frame-relay keepalive 6</b>	Enables the keepalive sequence.

Configuration Line	Description
<b>frame-relay lmi-n391dte 10</b>	Sets the full status polling interval.

- The LMI error threshold (**frame-relay lmi-n392dte/dce**) should be the same on the systems at both ends of the link as should be the LMI monitored event count (**frame-relay lmi-n393dte/dce**).

## LAPF Configuration

LAPF is enabled by executing the **frame-relay lmi-type** command with the **annex2-1** argument. The following example illustrates two configurable LAPF parameters:

```
nx# configure terminal
nx(config)# interface pos1/0
nx(config-if)# encapsulation frame-relay
nx(config-if)# frame-relay lmi-type annex2-1
nx(config-if)# frame-relay lapf t200 20
```

or

```
nx# configure terminal
nx(config)# interface pos1/0
nx(config-if)# encapsulation frame-relay
nx(config-if)# frame-relay lmi-type annex2-1
nx(config-if)# frame-relay annex2-1 ciscointerop-on
```

The following table explains the commands in the two LAPF examples above:

Configuration Line	Description
<b>frame-relay lmi-type annex2-1</b>	Enables LMI type annex 2-1 and LAPF functionality on the system. If LAPF is already enabled, you do not have to execute this command.
<b>frame-relay lapf t200 20</b>	Sets the retransmission LAPF T200 timer on interface pos1/0 to 2 seconds (10 = 1 second).
<b>frame-relay annex2-1 ciscointerop-on</b>	Activates LAPF packet padding and thereby enable the NX system to interoperate with non-Lucent systems that may require frames that are at least 10 bytes long.

## Quality of Service (QoS) and Traffic Management Configuration

The following example illustrates how to create a map class for specifying traffic rate and priority parameters.

```
nx# configure terminal
nx(config)# map-class frame-relay boston
nx(frl-map-config)# frame-relay priority 3
nx(frl-map-config)# frame-relay cir 1
nx(frl-map-config)# frame-relay be 2
nx(frl-map-config)# frame-relay bc 2
```

The following table explains the commands used in the QoS example above:

Configuration Line	Description
<code>map-class frame-relay boston</code>	Defines a class of QoS parameters. After specifying the map class, you can assign QoS parameters to that class.
<code>frame-relay priority 3</code>	Sets the priority level for the PVC endpoint in the egress (outbound) direction.
<code>frame-relay cir 1</code>	Configures the committed information rate (CIR) for a PVC.
<code>frame-relay be 2</code>	Configures the outgoing excess burst size (Be) for a PVC.
<code>frame-relay bc 2</code>	Configures the outgoing committed burst size (Bc) for a PVC.

Alternatively, after you have executed the `map-class frame-relay` command, you can execute the `frame-relay traffic-rate` command, as shown below, which configures traffic shaping characteristics for CIR, Be, and Bc with one command. The first number specifies the average traffic rate in mbps while the second number, which is optional, specifies the peak rate in mbps.

```
nx# configure terminal
nx(config)# map-class frame-relay boston
nx(frl-map-config)# frame-relay priority 3
nx(frl-map-config)# frame-relay traffic-rate 3 4
```

When working with QoS parameters on Frame Relay interfaces, be aware of the following:

- DLCIs configured on an interface (or subinterface) inherit the parameters (including any QoS parameters set) of the map class assigned to the interface (subinterface), unless you assign a different map class to the DLCI.
- A default-control map class is created at system initialization for the control DLCIs (DLCI 0 and DLCI 1023) to ensure that they have guaranteed bandwidth. The control DLCIs do not use the value set by a map class for an interface or subinterface. The `show frame-relay pvc summary` command shows the control DLCI settings.

- Use the `frame-relay priority` command to change a priority queue for the assigned map class or to create a new map and then reassign the map class to cross connected DLCIs.



The priority queue should be changed only during quiet periods. Because a change in priority queue assignment is effective immediately, it can result in out-of-order frame delivery for ongoing traffic.

## Verifying and Monitoring Frame Relay Connections

Use the commands in [Table 8-2](#) to verify and monitor Frame Relay connections. Also run the appropriate show command, such as `show frame-relay route`, to verify which parameters are already configured prior to making any configuration changes. All show commands are run at the global level.

**Table 8-2. Frame Relay Commands for Monitoring Connections**

Action	Command
Display the parameter and configuration settings currently in running memory.	<code>show running-config</code>
Send an echo request packet to a specified device and listen for an answer.	<code>ping</code>
Display configuration settings, statistics, and laser settings for all or specified interfaces configured on the system.	<code>show interfaces</code>
Display map class information.	<code>show frame-relay class</code>
Display LMI statistics and line parameter values. Also when lmi-type annex2-1 is enabled, display reasons for DLCI failure.	<code>show frame-relay lmi</code>
Display current map entries and related information	<code>show frame-relay map</code>
Display PVC-related values and its statistics for Frame Relay interfaces.	<code>show frame-relay pvc</code>
Display statistics based on DLCI status.	<code>show frame-relay pvc state</code>
Display a summary of PVC statistics.	<code>show frame-relay pvc summary</code>
List all configured Frame Relay routes and their status.	<code>show frame-relay route</code>
Display a summary of the configuration and status of Frame Relay interfaces.	<code>show frame-relay summary</code>
Display Frame Relay traffic statistics.	<code>show frame-relay traffic</code> (run only after ping is done)

## Implementation Differences

**Table 8-3** identifies pertinent implementation differences between the NX system and other vendors' systems that must be considered when configuring a heterogeneous network.

**Table 8-3. Implementation Differences**

Feature	Implementation Differences	Required Action
Frame Relay interface type	<p>Default setting:</p> <ul style="list-style-type: none"><li>• Cisco — DTE</li><li>• Juniper — DTE</li><li>• NX64000 — NNI</li></ul> <p>Frame Relay interoperability requires the Cisco interface set to DTE.</p> <p>Other configurations can generate error messages when the NX system has subinterfaces configured on a Frame Relay interface.</p>	On the NX system, set the interface type to DCE.
Frame Relay LMI type	<p>Default setting:</p> <ul style="list-style-type: none"><li>• Cisco — auto-sense</li><li>• NX system — auto-lmi</li></ul> <p>Two systems on a link using the <code>auto</code> setting can have difficulty negotiating the LMI type. In that case, <code>auto</code> setting must be disabled.</p>	Set the LMI type on the Cisco system to be the same type as the NX system.
Frame Relay Keepalive messages	<p>Default setting:</p> <ul style="list-style-type: none"><li>• Juniper — disabled</li><li>• NX64000 — enabled</li></ul>	Enable keepalive messages on the Juniper system.
Cyclic Redundancy Check (CRC)	<p>Default setting:</p> <ul style="list-style-type: none"><li>• Juniper — 16</li><li>• NX64000 — 32</li></ul>	The CRC value on both systems must be the same value, either 16 or 32.
Maximum Transmission Unit (MTU)	<p>Default setting:</p> <ul style="list-style-type: none"><li>• Juniper — 4470</li><li>• NX System — 4096</li></ul>	Set the protocol MTU and media MTU on the Juniper system to the NX system value.
Clock source	<p>Default setting:</p> <ul style="list-style-type: none"><li>• Juniper — internal system clock source</li><li>• NX Systems — internal system clock source</li></ul>	Configure the Juniper system to use an external clock source, which will then cause it to use the NX system clock source.

# Asynchronous Transfer Mode (ATM) Configuration

Asynchronous Transfer Mode (ATM) is a connection-oriented protocol in which the information is organized into cells. It is asynchronous in the sense that the recurrence of cells containing information from an individual user is not necessarily periodic.

The switch/router routes IP packets and terminates ATM connections, it does not switch cells. An ATM ingress line card takes cells from its network interface, reconstructs packets, and sends them through the switch fabric to the egress line card according to the IP header information in the ATM stream. When a system ATM line card is the egress card, it segments an IP packet into ATM cells and sends them out a predetermined virtual channel (VC).

## Key Features

The ATM implementation on the system supports:

- Support for Inverse ARP
- ATM Logical Link Control (LLC) and ATM Subnetwork Attachment Point (SNAP): places IP packets into an AAL5 frame and translates the IP address
- ATM Adaptation Layers: AAL5MUX (only IP is supported), AAL5 SNAP
- Support for BGP, OSPF, MPLS, and IS-IS routing protocols on ATM interfaces and subinterfaces

## Technology Concepts

Basic to understanding of ATM are:

- Virtual Channels (VCs) (sometimes called virtual circuits) and Virtual Paths (VPs)
- Permanent Virtual Circuits (PVCs)
- Virtual Path Identifiers (VPIs) and Virtual Channel Identifiers (VCIs)
- ATM Adaptation Layer AAL5

## Standards

This implementation of ATM is based on the following standards:

Document	Title
ATM Forum Specification af-sig-0061.000	ATM User-Network Interface Signalling Specification v4.0
RFC 2684	Multiprotocol Encapsulation over ATM Adaptation Layer 5
RFC 2390	Inverse Address Resolution Protocol
RFC 2558	Definitions of Managed Objects for the SONET/SDH Interface Type
RFC 2514	Definitions of Textual Conventions and OBJECT-IDENTITIES for ATM Management
RFC 2515	Definitions of Managed Objects for ATM Management

## Virtual Circuits and Virtual Paths

An ATM virtual circuit (VC) (also referred to as a virtual channel) is a logical point-to-point connection between an end system and an end point in the ATM network. Each virtual circuit is identified by a pair of numbers, the virtual path identifier (VPI) and the virtual channel identifier (VCI). The cells of a virtual circuit are identified in the ATM cell headers by the VPI/VCI pair.

A virtual path (VP) is a logical construct over a physical circuit. A physical circuit can carry multiple virtual paths. The virtual path can carry multiple virtual channels.

You can statically configure VCs as permanent virtual circuits (PVCs) or dynamically control them via signaling as switched virtual circuits (SVCs). The NX64000 supports only PVCs.

## Permanent Virtual Circuits (PVCs)

A PVC is a logical connection across a physical path. Multiple PVCs can share the same physical path. The PVC is statically mapped at every point in the ATM network. A failure of any link that a PVC crosses causes the PVC to fail. The PVC can be either point-to-point or point-to-multipoint. PVCs can be assigned a quality of service based on the amount of bandwidth allocated for their use.

Generally, PVCs are assigned to subinterfaces. Subaggregating numbers of PVCs to a few subinterfaces makes better use of system resources than does allocating fewer PVCs to a larger number of subinterfaces.

## Configuring PVCs

A PVC is typically assigned to a subinterface and is defined by a VPI/VCI pair and an encapsulation protocol. Optionally, you can enable inverse ARP.



The subinterface is configured to be either point-to-point (one VC on the subinterface) or multipoint (multiple VCs on the subinterface), and has an IP address. The system supports routable IP addresses assigned to endpoints on the system (addresses assigned to an endpoint such as a port interface or a subinterface). This does not refer to route addresses obtained via dynamic routing.

### Using VCDs and VPI/VCI Pairs

The virtual circuit descriptor (VCD) represents the VPI/VCI pair for the PVC. Remember these rules as you assign VCDs and VPI/VCI:

1. Every VCD needs to be unique per interface. You use a VCD number one time per system. To avoid confusion and configuration errors, use a unique VCD for each PVC on a system. The system supports VCD numbers 1 through 65535.
2. Do not use VCD 0.
3. The VPI/VCI pair needs to be unique per port.
4. The VPI/VCI pair needs to be unique per system IP address, use a VPI/VCI once for PVCs assigned under the same subinterface.

► You must plan carefully when assigning VCDs and VPI/VCI pairs. The CLI does not return an error message if you break the rules listed above.

### Using Sub-aggregation

PVCs are often sub-aggregated to subinterfaces, that is, large numbers of PVCs are assigned to a few subinterfaces. This practice makes efficient use of system resources, including IP addresses.

### Deleting or Changing a PVC

If you need to change parameters on a currently configured PVC, consider deleting the subinterface and then recreating it with the new values. If a PVC that is to be deleted is the only one on a subinterface, then it is better to delete and re-create the subinterface than just to remove the PVC. Deleting the interface also removes associated entries that are no longer needed in system tables and removes leftover entries from the configuration file.

## Virtual Path Identifiers (VPIs) and Virtual Channel Identifiers (VCIs)

Identifying a specific virtual channel (VC) on a physical circuit is done through the VPI/VCI fields of the ATM cell header. The virtual path identifier (VPI) is an eight-bit UNI (for UNI format) field in the ATM cell header that identifies a specific virtual path on a physical circuit. The virtual channel identifier (VCI) is a 16-bit field in the ATM cell header that, along with the VPI, identifies a specific virtual channel on a physical circuit.

The ATM layer uses the VPI/VCI pair to asynchronously interleave (multiplex) cells from multiple connections. VCIs and VPIs are not addresses, and only have local significance on an ATM network. They are explicitly assigned at each link between ATM nodes when a connection is established and remain for the duration of the connection.

VPI	VCI
0-255	1-1023 The range you can assign to the VPI depends upon the <code>atm vc-per-vp</code> and <code>atm max vpi-bits</code> settings.

Each bidirectional link in the ATM network is created with two PVCs, one in each direction (transmit/receive). You configure the same VPI/VCI in both directions to make the connection. Bandwidth allocated on one direction of the link can differ from that allocated in the other direction if traffic flows are not alike. The system automatically configures a transmit and receive PVC using the VPI/VCI combination specified by the `atm pvc` command.

### VPI/VCI Pairs and the Virtual Circuit Descriptor (VCD)

On the system each PVC is configured with a VCD and a VPI/VCI pair. When referenced, the VCD is linked to that VPI/VCI pair. The system supports VCD numbers 1 through 65535, VCD 0 is unavailable.

- For troubleshooting and debugging purposes it is suggested that you ensure that every VCD on a router has a unique VCD number.

### Determining VCs and VPs

To determine the number of VCs and VPs for the port you base it on the following process. This process defines what you can use for valid VPI settings when you configure using the `atm vc` command.

Configurable Parameter	Command to Set	Default	Range
VCs on an interface	<code>atm maxvc</code>	OC-3 is 2048 OC-12 is 8192	OC-3: 16, 32, 64, 128, 256, 512, 1024, 2048  OC-12: 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192
VPis per interface	<code>atm maxvpi-bits</code>	8	0 to $(2^n - 1)$ where $n=0$ to 8  See “ <b>VP Bit Range</b> ” on page 9-11.
# of VCs	<code>atm vc-per-vp</code>	OC-3 is 1024 OC-12 is 8192	OC-3: 16, 32, 64, 128, 256, 512, 1024, 2048  OC-12: 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192

The following formula represents the interrelation among the three components:

$$\text{Maximum total number of VCs per interface} = (\text{VCs per VP}) \times (\text{VPs per interface})$$

For example, given that the physical interface supports 2048 VCs by default, and you want to use eight different VPs (VPI 0 through 7), then you can configure up to 2048/8 or 256 VCs per VP. Here is the formula for this case:

$$\begin{aligned} \text{atm maxvc} &= \text{atm vc-per-vp} \times \text{number of VPs} \\ 2048 \text{ vc} &= 256 \text{ vc/vp} \times 8 \end{aligned}$$

You need to specify 3 bits in `atm maxvpi-bits` to allow the VPI 0–7 range. The `atm maxvc` command is not needed since the default value is used. Here are the commands to configure this case:

```
nx(config)# interface atm1/2
nx(config-if)# atm vc-per-vp 256
nx(config-if)# atm maxvpi-bits 3
```

First you find the maximum number of circuits per PVC. If you plan to use 256 rather than the default, you would divide 256 by 4, thus 64 VCs per VP. To enable the range of VPI numbers, you need to specify 8 bits. Here are the commands to configure this:

```
nx(config)# interface atm1/2
nx(config-if)# atm maxvc 256
nx(config-if)# atm vc-per-vp 64
nx(config-if)# atm maxvpi-bits 8
```

The following table explains each syntax line used in the preceding example:

Configuration Line	Description
<code>interface atm1/2</code>	Enters interface configuration mode for interface atm1/2.
<code>atm maxvc 256</code>	Specifies the maximum number of VCs supported on an ATM interface.
<code>atm vc-per-vp 64</code>	Specifies the maximum number of supported VCIs per VPI, that is, how many VCs can use the same VPI number.
<code>atm maxvpi-bits 8</code>	Sets the uppermost limit to the number of VPIs allowed on this interface.

## ATM Adaptation Layer

An ATM adaptation layer is an encapsulation type that formats a packet before segmentation and is recognized during reassembly. The system implements AAL5.

Each type of adaptation layer is designed to support specific service types. AAL5 supports high-speed, connection oriented data services.

## AAL and Encapsulation

The system supports two types of AAL5 encapsulation:

- Virtual channel multiplexed encapsulation, AAL5mux. When the `aal5mux` keyword is specified, a protocol is required. Only IP is supported.
- Logical Link Control/Subnetwork Access Protocol (LLC/SNAP) encapsulation, AAL5snap.

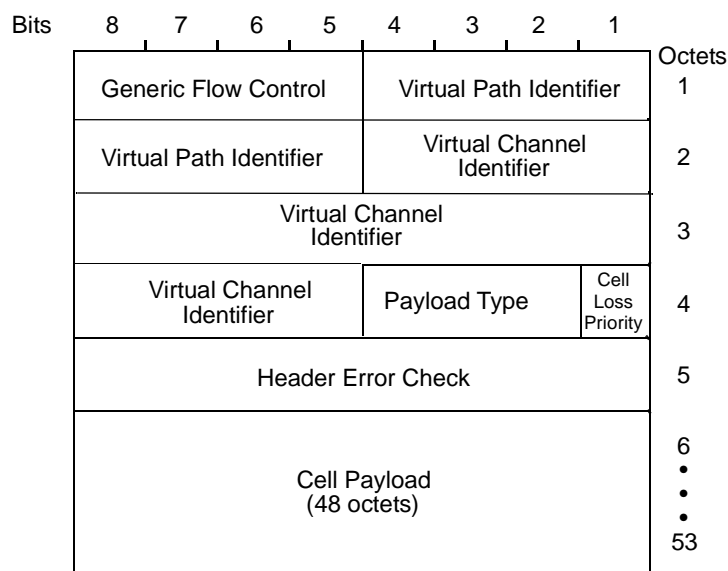
MTU enforcement (by either fragmenting a packet or refusing to forward a packet) is a function of the outgoing interface. For example, if an 8k packet comes into a POS interface and is forwarded out an ATM interface, it is the MTU setting of that egress ATM interface that dictates what MTU size to use to fragment the packet.

If the Don't Frag (DF) bit is set in the IP header and the packet size is greater than the MTU size of egress interface, then the packet is dropped. In addition, an ICMP message is generated and sent back to the source of the packet.

On system ATM line cards, the default is 4144 bytes.

## ATM UNI Cell Format

ATM is a packet-oriented transfer mode based on asynchronous time division multiplexing and the use of fixed length cells. A cell consists of a 5-byte header and a 48-byte data payload.



**Figure 9-1. Fields in an ATM cell**

The following table describes the ATM cell header fields:

**Table 9-1. ATM Cell Header Fields**

Field	Description
Generic Flow Control 4 bits	This is typically set to 0.
Virtual Path Identifier 8 bits	Identifies the Virtual Path.
Virtual Channel Identifier 16 bits	Identifies the Virtual Channel.
Payload Type	Differentiates between a cell payload carrying user information or one carrying management information.
Cell Loss Priority	Cell Loss Priority (CLP): 0 - indicates cells of highest priority 1 - indicates cells of lowest priority (selectively discarded during congestion)
Header Error Check	Provides error checking.
Cell Payload 48 bytes	Contains the data.

## Debugging and Logging Facilities

The NX-IS software provides logging and debug commands for ATM. In addition, you can use show commands to troubleshoot ATM. For more information about troubleshooting ATM problems, see in the *NX64000 Troubleshooting Guide* .

## ATM Configuration

This table lists commands used for the configuration and verification of ATM interfaces and features. Each command listing indicates the general functions for which you use the command. Some commands are included and described in the configuration examples that follow. All commands listed are documented in the “*NX64000 Command Reference*.”

**Table 9-2. ATM Command Usage**

Command	Defining interfaces	ATM ARP	QoS and bandwidth	Verification and debug logging
atm maxvc			✓	
atm maxvpi-bits			✓	
atm pvc	✓			
atm-vc	✓			
atm vc-per-vp	✓		✓	
clock-source	✓			
debug atm arp		✓		✓
debug atm errors interfaces	✓			✓
debug atm errors vc	✓			✓
debug atm packet	✓			✓
debug logging buffer atm				✓
debug logging console atm				✓
description	✓		✓	
framing	✓			
interface	✓			
loopback				✓
map-group	✓			

**Table 9-2. ATM Command Usage**

Command	Defining interfaces	ATM ARP	QoS and bandwidth	Verification and debug logging
map-list	✓			
mtu	✓			
protocol-overhead	✓			
show atm interfaces	✓			✓
show atm map				✓
show atm vc	✓			✓
show atm vc-summary	✓			✓
show debugging	✓			✓
show ip arp		✓		✓
show ip interface	✓			✓
show ip route	✓			✓
show logging messages atm				✓
show logging registered				✓
show sonet				✓

## Basic ATM Configuration Tasks

- This chapter generally uses the term interfaces when referring to interfaces as well as subinterfaces. When subinterfaces is used, the reference is limited to subinterfaces. Subinterfaces are logical interfaces on a physical port. For more information on cards and interfaces, see [Chapter 7, “Cards and Interfaces”](#).

Basic ATM configuration requires:

1. Setting up physical interfaces.
2. Managing virtual channels.
3. Configuring ATM PVCs.
4. Creating map lists for static routing.
5. Verifying ATM configurations with `show` commands.
6. Controlling message logging with `debug` commands.

- The system supports only the IP protocol over ATM. The system requires AAL5snap encapsulation for interfaces that run IS-IS or that use inverse ARP because both use non-IP frames.

Depending on the network configuration, you can also configure the following:

- Loopback mode for a specified interface.
- A maximum number of VCs, and a corresponding number of VPIs and VCIs.

## Setting Up Port (Physical) Interfaces

When you configure a physical ATM interface on the system, the hardware/physical-layer settings are made at the primary interface level and are inherited by its subinterfaces.

- It is usually not necessary to assign an IP address to a port interface. The system supports configurable IP addresses. These are endpoints on the router, not routes obtained from dynamic routing. Depending on your requirements, it may be more useful to reserve addresses for subinterfaces.

The following example shows interface settings on a system located in Europe that have the interface recover its clock from the other endpoint and support an ATM VC multiplexed connection:

```
nx# configure terminal
nx(config)# interface atm4/0
nx(config-if)# description "path to mn_reg_net"
nx(config-if)# mtu 4144
nx(config-if)# clock-source line
nx(config-if)# framing sdh
nx(config-if)# protocol-overhead 8
nx(config-if)#
```

The following table explains each syntax line used in the preceding example:



Configuration Line	Description
<code>interface atm4/0</code>	Names primary interface on port 0.
<code>description "path to mn_reg_net"</code>	Describes connecting customer network.
<code>mtu 4144</code>	Adjusts MTU for egress traffic. The MTU reflects what can be handled at the other end of the ATM link, and thus is configurable at the subinterface level.
<code>clock-source line</code>	Sets the clock source for the primary interface. Ports can operate with an internal or external clock, depending upon link requirements.
<code>framing sdh</code>	Enables the required framing for SDH. Ports need a framing option according to geographical location.
<code>protocol overhead 8</code>	Set the encapsulation overhead value. Encapsulation and associated overhead depends on connection protocol type.

## Managing Virtual Channels

A virtual channel (VC) is associated with a virtual path (VP). Each VPI supports some number of VCI. Numbers of VCs and VPs are managed and configured per port (physical interface). These circuit components are configurable within system ranges, and their values are set using the `atm maxvc`, `atm maxvpi-bits`, and `atm vc-per-vp` commands.

- ▶ Virtual channel parameters are configured on a per port basis, at the physical interface level, rather than at the subinterface level.

### VP Bit Range

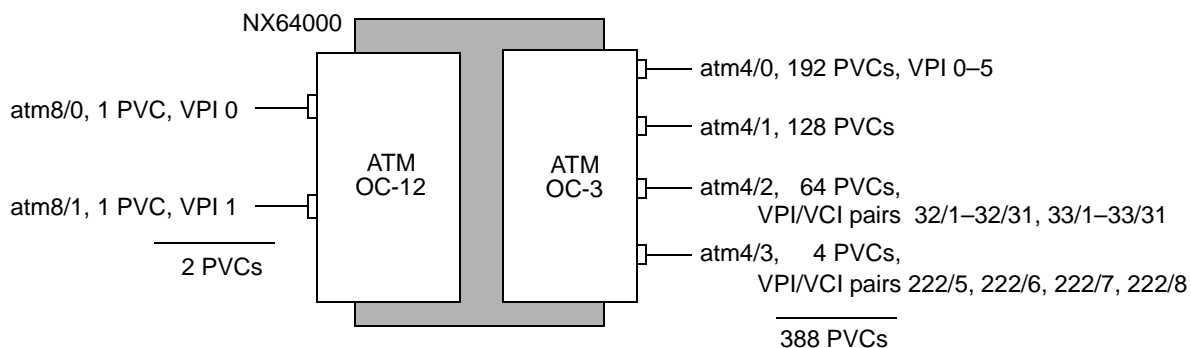
The value you designate in the `atm maxvpi-bits` command sets the uppermost limit of VPIs allowed on the interface the system supports a maximum of 256. VPI values range from 0 to  $(2^n - 1)$ , where  $n$  is a value from 0 to 8. For example, if  $n$  is 3, you can specify VPIs numbered 0 through 7, inclusive. The following table shows the range of values and VPI numbers:

0 = VPI 0	5 = VPIs 0–31
1 = VPIs 0–1	6 = VPIs 0–63
2 = VPIs 0–3	7 = VPIs 0–127
3 = VPIs 0–7	8 = VPIs 0–255 (default)

4 = VPIs 0-15	
---------------	--

## Configuring VC and VP Parameters

**Figure 9-2** shows configured VC and VP parameters on primary interfaces on the ATM OC-3 and ATM OC-12 cards.



**Figure 9-2. Example Allocating Numbers of VCs and VPs**

atm4/0

Interface atm4/0 requires the use of VPIs 0 - 5. The most appropriate VPI range to specify is 0 - 7, so the `atm maxvpi-bits` value is bit 3. The interface is required to support 192 PVCs. Using the following formula for allotting VCs and VPs you get:

```
atm maxvc = atm vc-per-vp x #VPIs
192 vc    = ?? vc/vp x 6
192 vc    = 32 vc/vp x 6
```

A setting of 32 is an allowed value for the `atm vc-per-vp` command. However, 192 is not a legal value for the `atm maxvc` command and so the next higher value (256) is used in the actual command.

► By configuring this interface for 256 VCs note that all 256 are consumed against the maximum available on the card and router.

This example configures the channel parameters on interface atm4/0:

```
nx# configure terminal
nx(config)# interface atm4/0
nx(config-if)# description "path to mn_reg_net"
nx(config-if)# mtu 4144
nx(config-if)# atm maxvc 256
nx(config-if)# atm vc-per-vp 32
nx(config-if)# atm maxvpi-bits 3
```

The following table explains each syntax line used in the preceding example:

Configuration Line	Description
<b>interface atm4/0</b>	Enters interface configuration mode for interface atm4/0.
<b>description "path to mn_reg_net"</b>	Describes connecting customer network.
<b>mtu 4144</b>	Adjusts MTU for egress traffic.
<b>atm maxvc 256</b>	Sets the number of virtual channels configured on this physical interface to a maximum of 256. Because the intended number, 192, is not a legal value for this command, the closest or next higher allowed value is used.
<b>atm vc-per-vp 32</b>	Enables up to 32 VCs to use the same VPI number. On this interface, VCI numbers can range between 0 and 31.
<b>atm maxvpi-bits 3</b>	Bit 3 enables you to use eight VPI numbers (VPI 0–7). See <b>"VP Bit Range"</b> on page 9-11.

### Verify Channel Parameters

To verify that the parameters are configured correctly, use the **show interfaces** command and specify the target interface:

```

nx# show interfaces atm4/0
atm4/0 is up, line protocol is up (ifindex is 6)
  Framing SONET, Clock-source Chassis, Laser is On
  Loopback not set
  Hardware is atm4/0, Maker OC-3
  MTU 4144 bytes, BW 150336 Kbit
  Encapsulation(s): AAL5, PVC Mode
  Max VCCs: 256, VCs per VP: 32, Max VPI bits: 3
  .
  .

```

### atm8/0 and atm8/1

Interfaces atm8/0 and atm8/1 require VPI 0 and 1, respectively, and therefore, the **atm maxvpi-bits** values are 0 and 1. Because 1 is not a legal value for **atm maxvc** or **atm vc-per-vp**, the closest values allowed (16 and 16) are used in the actual commands. This example configures the channel parameters for atm8/0 and atm8/1:

```

nx# configure terminal
nx(config)# interface atm8/0
nx(config-if)# description "path to east_net1"
nx(config-if)# mtu 4144
nx(config-if)# atm maxvc 16
nx(config-if)# atm vc-per-vp 16
nx(config-if)# atm maxvpi-bits 0
nx(config-if)# interface atm8/1

```

```
nx(config-if)# description "path to east_net2"
nx(config-if)# mtu 4144
nx(config-if)# atm maxvc 16
nx(config-if)# atm vc-per-vp 16
nx(config-if)# atm maxvpi-bits 1
```

atm4/1

Interface atm4/1 requires 128 PVCs and VPIs 0–5 are to be used, the atm maxvpi-bits value is 3 (enables VPI 0–7). Using the formula, 128 divided by 6 does not equal one of the values allowed for atm vc-per-vp, the closest value is 32. This example configures the channel parameters for atm4/1:

```
nx# configure terminal
nx(config)# interface atm4/1
nx(config-if)# description "path to west_net1"
nx(config-if)# mtu 4144
nx(config-if)# atm maxvc 128
nx(config-if)# atm vc-per-vp 32
nx(config-if)# atm maxvpi-bits 3
```

atm4/2

Interface atm4/2 requires 64 PVCs on VPI 32 and VPI 33, the atm maxvpi-bits value is 6 bits (enables VPI 0–63). This example configures the channel parameters for atm4/2:

```
nx# configure terminal
nx(config)# interface atm4/2
nx(config-if)# description "path to south_net2"
nx(config-if)# mtu 4144
nx(config-if)# atm maxvc 64
nx(config-if)# atm vc-per-vp 32
nx(config-if)# atm maxvpi-bits 6
```

atm4/3

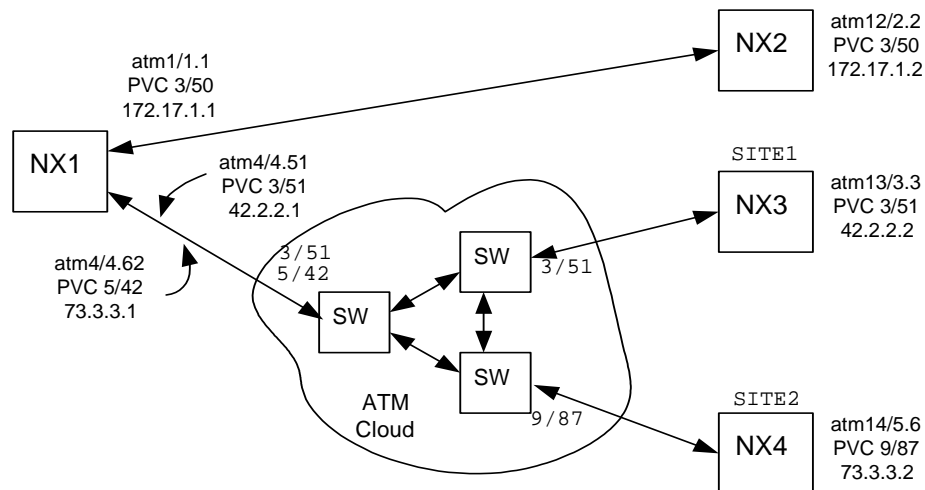
Interface atm4/3 requires 4 PVCs to be assigned on VPI 222 and VCI 5–8, the atm maxvpi-bits value is 8 bits, the default value (enables VPI 0–255). Minimum values of 16 are used for atm maxvc and atm vc-per-vp since there are only 4 channels. This example configures the channel parameters for atm4/3:

```
nx# configure terminal
nx(config)# interface atm4/3
nx(config-if)# description "path to mn_reg_net"
nx(config-if)# mtu 4144
nx(config-if)# atm maxvc 16
nx(config-if)# atm vc-per-vp 16
```

## Configuring Point-to-Point PVCs

This example demonstrates two different methods of using point-to-point PVCs. In the simple example of using point-to-point PVCs, systems NX1 and NX2 in [Figure 9-3](#) are directly connected to each other and use a single subnet on their physical ports.

In the more complex example, there is a virtual direct connection between system NX1 and systems NX3 and NX4 via an ATM cloud. System NX1 aggregates the disparate networks of SITE1 and SITE2 using a single physical port.



**Figure 9-3. Point-to-Point PVC Configuration Examples**

Connecting PVCs need to match or be coordinated as follows:

- VPI/VCI numbers must match unless they are separated by a switch/router. For example, NX1 (5/42) and NX4 (9/87).
- Encapsulation protocols must match.
- All devices must be inverse ARP-capable.
- Only one side of a link can supply clock timing, not both. While both ends of a PVC can use internal clocking, only one side can be configured for line (external) clocking.

Configurations for the Simple Example (NX1 to NX2)

This example establishes a PVC pair between two directly connected systems using inverse ARP. (The NX2 machine does not need to be an NX64000 but would have to be inverse ARP capable.) Assume NX2 is in Europe (where SDH framing is typical). For an example of interfacing to a non-InARP capable device, see [“Using Map Lists for Static Routing” on page 9-23](#).

NX1 Configuration

The following example creates the primary (or parent) interface for physical port 1 of the ATM line card in slot 1, then creates the atm1/1.1 subinterface under the primary interface, and on the subinterface it creates one PVC that links to NX2.

Default settings for atm maxvc, atm maxvpi-bits, and atm vc-per-vp are assumed. Framing needs to be specified as SDH for this configuration. NX1 recovers its clock from NX2.

The PVC between NX1 and NX2 uses VPI/VCI 3/50. On the NX1 box, the 3/50 circuit is referenced by VCD 350. The method for selecting a VCD is not dictated, but using some type of systematic approach aids in avoiding total randomness; in this case, a scheme is used that matches the VCD to the VPI/VCI.

```
nx1# configure terminal
nx1(config)# interface atm1/1
nx1(config-if)# description "path to NX2"
nx1(config-if)# mtu 4083
nx1(config-if)# clock-source line
nx1(config-if)# framing sdh
nx1(config-if)# exit
nx1(config)# interface atm1/1.1 point-to-point
nx1(config-subif)# ip address 172.17.1.1 255.255.0.0
nx1(config-subif)# atm pvc 350 3 50 aal5snap inarp 2
nx1(config-subif)# description "PVC to NX2 12/2.2"
nx1(config-subif)# exit
```

The following table explains each syntax line used in the preceding example:

Configuration Line	Description
<code>interface atm1/1</code>	Enters interface configuration mode for interface atm1/1.
<code>description "path to NX2"</code>	Attaches a text description to the interface.
<code>mtu 4083</code>	Adjusts the MTU for egress traffic.
<code>clock-source line</code>	Sets clock timing to be recovered from the line. NX2 acts as the clock source.
<code>framing sdh</code>	Sets frame mode to SDH.
<code>interface atm1/1.1 point-to-point</code>	Creates atm1/1.1 subinterface on atm1/1 and selects point-to-point mode.
<code>ip address 172.17.1.1 255.255.0.0</code>	Assigns the subinterface an IP address and mask.
<code>atm pvc 350 3 50 aal5snap inarp 2</code>	Configures a PVC on the subinterface with a VCD of 350. This VCD represents the 3/50 VPI/VCI pair.  Assigns encapsulation protocol and specifies Inverse ARP update every 2 minutes.
<code>description "PVC to NX2 atm12/2.2"</code>	Identifies the endpoint of the PVC.

## Verifying Interface and PVC Configuration for NX1

After entering the configuration, use the **show atm vc**, **show interface**, and **show atm vc vcd #** commands to check that configuration settings are correct.

```

nx1# show atm vc

```

Interface	VCD	VPI	VCI	AAL/ Encapsulation	Peak Kbps	Avg. Kbps	Burst cells	Status
atml/1.1	350	3	50	AAL5-SNAP	0	0	0	ACTIVE

```

End of VC-list
nx1# show interfaces atml/1
atml/1 is up, line protocol is up (ifindex is 79)
  Framing SDH, Clock-source Line, Laser is On
  Loopback not set
  Hardware is atml/1, Maker OC-3
  Description:
  MTU 4083 bytes, BW 150336 Kbit
  Encapsulation(s): AAL5, PVC Mode
  Max VCCs: 2048, VCs per VP: 1024, Max VPI bits: 8
  Interface up time 00:00:00
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  Input: 0:0 packets, 0:0 bytes, 0 errors, 0 drops
    Local input: 0:0 packets, 0:0 bytes, 0 drops
    Local 5 minute input rate 0 bits/sec, 0 packets/sec
  Output: 0:0 packets, 0:0 bytes, 0 errors, 0 drops
    Local output: 0:0 packets, 0:0 bytes
    Local 5 minute output rate 0 bits/sec, 0 packets/sec
nx1# show atm vc 350
atml/1.1      VCD: 350, VPI 3 VCI 50
Encaps: AAL5-SNAP
Peak rate:      0 Avg rate:      0 Burst:      0
OAM interval 0 second(s)
InArp interval 2 minute(s)
State ACTIVE
In Packets 0:0, Out Packets 0:0
In Octets 0:0, Out Octets 0:0
End of VC-list
nx1#

```

If the PVC state is **ENABLED**, it usually means that the link optics are not yet registered with each other, or it may mean that the VC at the other end of the link is not yet configured. If the state does not go to **ACTIVE**, check that the configurations at both ends of the link match.

The fields **Peak rate**, **Avg rate**, and **Burst rate** are not for current statistics, but indicate the configured traffic rates. The system does not currently support traffic shaping, so the rates display zero. Similarly, **OAM** is not currently supported in the system, even though **OAM interval** appears in the **VCD** printout.

## NX2 Configuration

This series of commands creates subinterface atm12/2.2, and then creates one PVC to NX1 on the subinterface.

For this example, the VCD to represent VPI/VCI 3/50 was randomly chosen to be 133.

```
nx2# configure terminal
nx2(config)# interface atm12/2
nx2(config-if)# description "path to NX1 atm1/1"
nx2(config-if)# mtu 4083
nx2(config-if)# clock-source internal
nx2(config-if)# framing sdh
nx2(config-if)# exit
nx2(config)# interface atm12/2.2 point-to-point
nx2(config-subif)# ip address 172.17.1.2 255.255.255.0
nx2(config-subif)# atm pvc 133 3 50 aal5snap inarp 10
nx2(config-subif)# description "PVC to NX1 atm1/1.1"
```

Verify the interface and PVC configuration using the `show atm vc`, `show interface`, and `show atm vc vcd #` commands. Also, verify the Inverse ARP operation using the `show ip arp` command. The output from the NX2 system should be similar to the following:

```
nx2#show ip arp
Host          Interface    MAC Address      Type    ARP Type    Age
-----
172.17.1.1    atm12/2.2    04:01:00:66:00:00  PTPT
Dynamic Pt-Pt          358
172.17.1.2    atm12/2.2    00:00:00:00:00:00  PTPT    Local
Pt-Pt
172.17.143.0   ethernet0    ff:ff:ff:ff:ff:ff  LAN     Local
Broadcast
172.17.143.1   ethernet0    00:c0:80:8f:93:3e  LAN
Dynamic Router          359
172.17.143.45   ethernet0    08:00:3e:2a:cc:93  LAN     Local
172.17.143.255 ethernet0    ff:ff:ff:ff:ff:ff  LAN     Local
Broadcast
nx2#
```

As the display shows, the NX2 machine (IP address 172.17.1.2) found the IP address of its NX1 peer (IP address 172.17.1.1) using Inverse ARP and has installed the path as a dynamically ascertained point-to-point connection.

## Configurations for the More Complex Example (NX1 to NX3,NX4)

In **Figure 9-3**, NX3 and NX4 might be two previously independent companies that are being tied into the NX1 system at corporate headquarters. This example uses an ATM network to communicate with the two disparate systems using a single physical port. All systems in the example are Inverse ARP capable. Neither the NX3 nor NX4 in this example needs to be an NX64000, but would have to be Inverse ARP capable. Assume all systems shown are in the United States (where SONET framing is the standard).



Before detailing the configurations for each of the systems, note the following:

- Since an ATM network is connecting the systems, the VPI/VCI do not have to remain constant for the entire VC; the VPI/VCI only need to remain constant along a segment. From the system perspective, this means the VPI/VCI need only match their segment's entry into the ATM network. For NX1 to NX3, VPI/VCI 3/51 is allocated to both ends of the PVC. For NX1 to NX4, the segment from NX1 to the ATM network uses VPI/VCI 5/42, while the segment from NX4 to the ATM network uses 9/87. Mechanisms within the ATM network internally map NX1's 5/42 with NX4's 9/87 to create an end-to-end circuit.
- To connect NX1 to the ATM network, a single fiber is used from port 4 of the ATM card in slot 4 of NX1. But that single fiber carries two bidirectional point-to-point PVCs that handle full-duplex traffic from distinct systems on the other end of the ATM network.
- Logical extension of this example would allow a single physical port to directly access a large number of disparate networks.

## NX3 and NX4

The configurations of NX3 and NX4 are very similar to NX2, only instead of directly communicating with another NX, the devices are really linking up to the ATM network. Both configurations use the defaults for framing, clock source, mtu, atm maxvpi-bits, and atm vc-per-vp.

This series of commands for NX3 creates the primary interface for physical port 3 of the ATM line card in slot 13, creates the atm13/3.3 subinterface, and then creates one PVC to NX1 on the subinterface.

```
nx3# configure terminal
nx3(config)# interface atm13/3
nx3(config-if)# description "path to NX1 atm4/4"
nx3(config-if)# exit
nx3(config)# interface atm13/3.3 point-to-point
nx3(config-subif)# ip address 42.2.2.2 255.255.255.0
nx3(config-subif)# atm pvc 82 3 51 aal5snap inarp 10
nx3(config-subif)# description "PVC to NX1 atm4/4.51"
```

This series of commands for NX4 creates subinterface atm14/5.6, and then creates one PVC to NX1 on the subinterface.

```
nx4# configure terminal
nx4(config)# interface atm14/
nx4(config-if)# description "path to NX1 atm4/4"
nx4(config-if)# mtu 4083
nx4(config-if)# exit
nx4(config)# interface atm14/5.6 point-to-point
nx4(config-subif)# ip address 73.3.3.2 255.255.255.0
nx4(config-subif)# atm pvc 83 9 87 aal5snap inarp 10
nx4(config-subif)# description "PVC to NX1 atm4/4.62"
```

Use the show atm vc, show interface, and show atm vc vcd# commands to check that the configuration settings are correct.

## NX1

This series of commands configures physical port 4 of NX1 to connect with NX3 and NX4, as shown in [Figure 9-3](#). The configuration uses the defaults for framing, clock source, and mtu size. However, it is not possible to use the system default for atm maxvpi-bits or atm vc-per-vp. NX1 must communicate with VPIs 3 and 5; that means a value of 3 for atm maxvpi-bits, giving a VPI range of 0-7. Using the default atm maxvc of 2048, the value for atm vc-per-vp needs to be set to 256. These values need to be set on the primary (parent) interface since they affect all subinterfaces. Once the parent interface is set up, the configuration of subinterfaces follows the same pattern as in the simple method example of directly connected boxes.

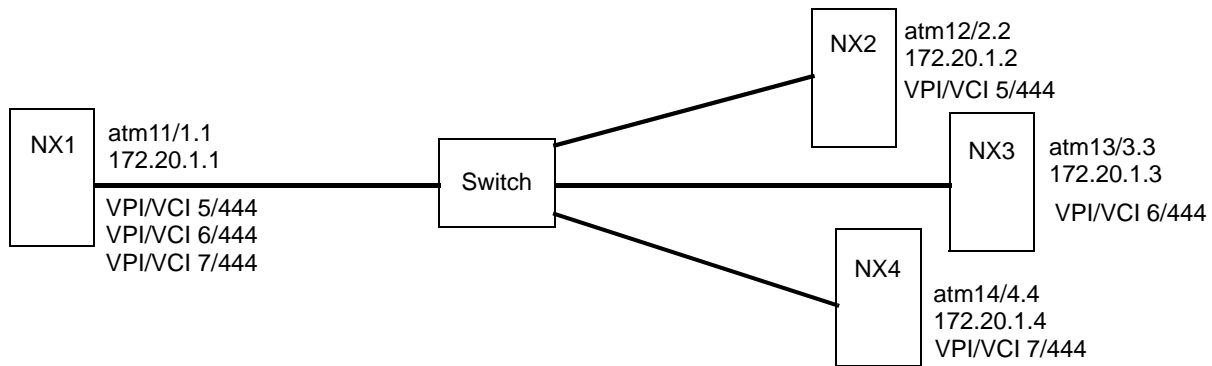
```
nx1# configure terminal
nx1(config)# interface atm4/4
nx1(config-if)# description "aggregation of SITE1 and SITE2"
nx1(config-if)# atm max-vpibits 3
nx1(config-if)# atm vc-per-vp 256
nx1(config-if)# exit
nx1(config)# interface atm4/4.51 point-to-point
nx1(config-subif)# ip address 42.2.2.1 255.255.255.0
nx1(config-subif)# atm pvc 351 3 51 aal5snap inarp 10
nx1(config-subif)# description "PVC to SITE1 NX3 atm13/3.3"
nx1(config-subif)# exit
nx1(config)# interface atm4/4.62 point-to-point
nx1(config-subif)# ip address 73.3.3.1 255.255.255.0
nx1(config-subif)# atm pvc 542 5 42 aal5snap inarp 10
nx1(config-subif)# description "PVC to SITE2 NX4 atm14/5.6"
```

Use the `show atm vc`, `show interface`, and `show atm vc vcd #` commands to check that configuration settings are correct. Verify the Inverse ARP operation using the `show ip arp` command. Confirm connectivity by using the `ping` command from NX1. If you have implemented the network shown in [Figure 9-3](#), you should be able to ping NX2, NX3, or NX4 from NX1. The results should look similar to the following ping of NX4:

```
nx1# ping 73.3.3.2
100 bytes from 73.3.3.2: icmp_seq=0, time=2 ms
100 bytes from 73.3.3.2: icmp_seq=1, time=2 ms
100 bytes from 73.3.3.2: icmp_seq=2, time=2 ms
100 bytes from 73.3.3.2: icmp_seq=3, time=2 ms
100 bytes from 73.3.3.2: icmp_seq=4, time=2 ms
----73.3.3.2 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 2/2/2
```

## Configuring Point-to-Multipoint PVCs

This example establishes PVCs between one interface on one local router and several remote systems located in the U.S. (i.e., SONET framing). Only one physical port is needed at the local router. Each uses internal clocking. All systems in this example use Inverse ARP. For an example of connecting to a non-InARP capable device, see [“Using Map Lists for Static Routing” on page 9-23](#).



**Figure 9-4. Point-to-Multipoint PVC Configuration Example**

Connecting PVCs need to match or be coordinated as follows:

- VPI/VCI numbers must match (only with point-to-point and no switch in between).
- Encapsulation protocol must match.
- Both PVCs can use internal clock, but only one side of the link can supply the clock (line).

NX1

This example creates subinterface atm11/1.1 and on it, three PVCs to remote systems. All routers use Inverse ARP so the IP addresses of remote systems do not need to be statically configured. The configuration uses the defaults for atm maxvc, atm maxvpi-bits, atm vc-per-vp, clock-source and framing.

```

nx1# configure terminal
nx1(config)# interface atm11/1
nx1(config-if)# description "path to metro switch"
nx1(config-if)# mtu 4083
nx1(config-if)# exit
nx1(config)# interface atm11/1.1 multipoint
nx1(config-subif)# ip address 172.20.1.1 255.255.255.0
nx1(config-subif)# description "PVCs to metro"
nx1(config-subif)# atm pvc 5444 5 444 aal5snap inarp 12
nx1(config-subif)# atm pvc 6444 6 444 aal5snap inarp 11
nx1(config-subif)# atm pvc 7444 7 444 aal5snap inarp 10
  
```

The following table explains each syntax line used in the preceding example:

Configuration Line	Description
<code>interface atm11/1</code>	Enters interface configuration mode for interface atm11/1.
<code>description "path to metro_switch"</code>	Attaches a text description to the interface.
<code>mtu 4083</code>	Adjusts the MTU for egress traffic.

Configuration Line	Description
<b>interface atm11/1.1 multipoint</b>	Creates atm11/1.1 subinterface and enables multiple VCs to be configured. A subinterface must be set to either multipoint or point-to-point.
<b>ip address 172.20.1.1 255.255.0.0</b>	Assigns the subinterface an IP address and mask.
<b>atm pvc 5444 5 444 aal5snap inarp 12</b>	Configures a PVC on the subinterface with a VCD of 5444. This VCD represents the 5/444 VPI/VCI pair. Assigns encapsulation protocol and inverse ARP/interval.
<b>description "PVCs to metro"</b>	Describes destination end of this interface.
<b>atm pvc 6444 6 444 aal5snap inarp 11</b>	Configures a PVC on the subinterface with a VCD of 6444. This VCD represents the 6/444 VPI/VCI pair. Assigns encapsulation protocol and inverse ARP/interval.
<b>atm pvc 7444 7 444 aal5snap inarp 10</b>	Configures a PVC on the subinterface with a VCD of 7444. This VCD represents the 7/444 VPI/VCI pair. Assigns encapsulation protocol and inverse ARP/interval.

### Verifying Interface and PVC Configuration

After you have entered the configuration, use **show interfaces** to check that configuration settings for the interface are correct.

The **show atm vc** command lists all the PVC VCDs and the settings for connecting links:

```

nx2# show atm vc

```

Interface	VCD	VPI	VCI	AAL/ Encapsulation	Peak Kbps	Avg. Kbps	Burst cells	Status
atm11/1	5444	5	444	AAL5-SNAP	0	0	0	ENABLED
atm11/1	6444	6	444	AAL5-SNAP	0	0	0	ACTIVE
atm11/1	7444	7	444	AAL5-SNAP	0	0	0	ACTIVE

```

End of VC-list

```

The next section provides the series of configuration commands for the remote systems.

### NX2

This series of commands creates subinterface atm12/2.2 and one PVC to NX1. The PVCs between NX2 and NX1 use VPI/VCI 5/444.

```

nx2# configure terminal
nx2(config)# interface atm12/2
nx2(config-if)# description "path to metro switch"
nx2(config-if)# mtu 4083
nx2(config-if)# exit

```

```
nx2(config)# interface atm12/2.2 point-to-point
nx2(config-subif)# ip address 172.20.1.2 255.255.0.0
nx2(config-subif)# description "PVC to metro1"
nx2(config-subif)# atm pvc 5444 5 444 aal5snap inarp 10
```

### NX3

This series of commands creates subinterface atm13/3.3 and one PVC to NX1. The PVCs between NX3 and NX1 use VPI/VCI 6/444.

```
nx3# configure terminal
nx3(config)# interface atm13/3
nx3(config-if)# description "path to metro switch"
nx3(config-if)# mtu 4083
nx3(config-if)# exit
nx3(config)# interface atm13/3.3 point-to-point
nx3(config-subif)# ip address 172.20.1.3 255.255.0.0
nx3(config-subif)# description "PVC to metro1"
nx3(config-subif)# atm pvc 6444 6 444 aal5snap inarp 10
```

### NX4

This series of commands creates subinterface atm14/4.4 and one PVC to NX1. The PVCs between NX4 and NX1 use VPI/VCI 7/444.

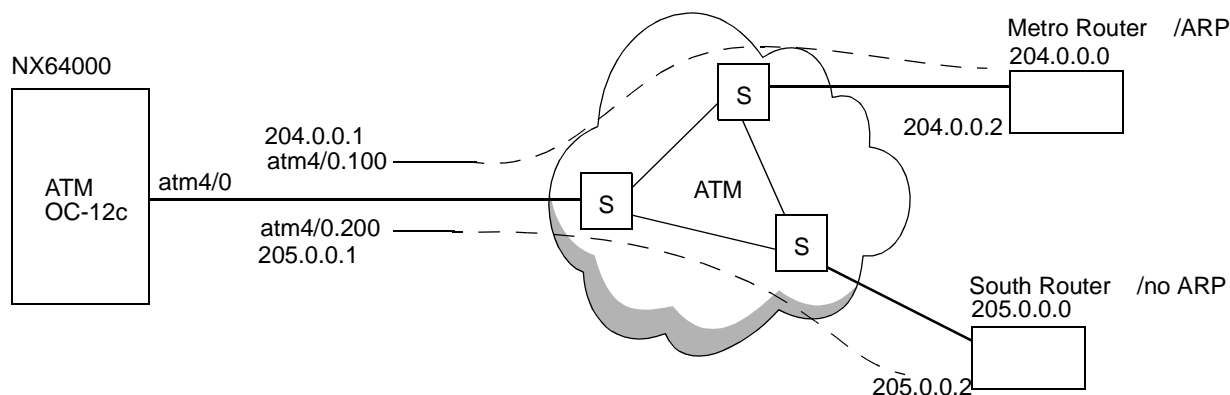
```
nx3# configure terminal
nx3(config)# interface atm14/4
nx3(config-if)# description "path to metro switch"
nx3(config-if)# mtu 4083
nx3(config-if)# exit
nx3(config)# interface atm14/4.4 point-to-point
nx3(config-subif)# ip address 172.20.1.4 255.255.0.0
nx3(config-subif)# description "PVC to metro1"
nx3(config-subif)# atm pvc 7444 7 444 aal5snap inarp 10
```

## Using Map Lists for Static Routing

You can use a map list to associate the IP address of a remote system with an interface and PVC configured on an ATM interface. This is useful especially when the destination system does not support ARP.

In this example, the system has an ATM connection into an ATM cloud. Two subinterfaces have routes to the remote systems shown. A PVC on atm4/0.100 forwards to the Metro network. The Metro router supports Inverse ARP so the PVC is configured with the inverse ARP parameter.

Forwarding traffic to the South network is done via PVC atm4/0.200. The South router does not support inverse ARP. To provide the PVC with its destination IP address, it is assigned a map group. The map-list entry maps the destination IP address on South to the VCD of this PVC.



**Figure 9-5. Map Group Configuration Example**

This example configures the two atm4/0 interfaces, a PVC on each, and the map group:

```
nx# configure terminal
nx(config)# interface atm4/0.100 point-to-point
nx(config-subif)# description "metro_link"
nx(config-subif)# ip address 204.0.0.1 255.255.255.0
nx(config-subif)# atm pvc 100 0 100 aal5snap inarp 12
nx(config-subif)# exit
nx(config)# interface atm4/0.200 point-to-point
nx(config-subif)# description "south_link"
nx(config-subif)# ip address 205.0.0.1 255.255.255.0
nx(config-subif)# atm pvc 200 0 200 aal5snap
nx(config-subif)# map-group south
nx(config-subif)# exit
nx(config)# map-list south
nx(map-list)# ip 205.0.0.2 atm-vc 200
nx(map-list)# exit
```

The following table explains each syntax line used in the preceding example:

Configuration Line	Description
<code>interface atm4/0.100 point-to-point</code>	Enters interface configuration mode and creates subinterface 0.100 on a card. The net type is set to point-to-point.
<code>description "metro_link"</code>	Attaches a text description to the interface.
<code>ip address 204.0.0.1 255.255.255.0</code>	Assigns the new interface an IP address and mask.
<code>atm pvc 100 0 100 aal5snap inarp 12</code>	Creates a PVC and enables inverse ARP capability on it.
<code>interface atm4/0.200 point-to-point</code>	Enters interface configuration mode and creates subinterface 0.200 on the card.
<code>description "south_link"</code>	Attaches a text description to the interface.

Configuration Line	Description
<code>ip address 205.0.0.1 255.255.255.0</code>	Assigns the interface an IP address and mask.
<code>atm pvc 200 0 200 aa15snap</code>	Creates a PVC on the interface.
<code>map-group south</code>	Includes this interface in the map group named "south".
<code>map-list south</code>	Enters map-list configuration mode and enables the map group named "south".
<code>ip 205.0.0.2 atm-vc 200</code>	A map list statement, maps destination IP address 205.0.0.2 to local circuit 200.

## Verifying a Map List

Use the `show atm map` command to verify map group name, IP address, and which VC the IP address maps to.

```

nx# show atm map
Map List 200:
    ip 205.0.0.2 maps to VC 200

```

## Verifying ATM Configuration

The NX-IS software provides a number of show commands that let you verify configuration and status. The following tables lists the type of information you can view from each show command:

**Table 9-3. ATM Commands for Verifying Configuration**

Action	Command
Displays interface characteristics, circuit identification, and traffic counts.  Use the <code>details</code> option to show error and exception counts for input and output packets.	<code>show interfaces</code>
Displays interface characteristics, circuit identification, and traffic counts for ATM interfaces.	<code>show atm interfaces</code>
Displays the static route maps configured for ATM VCs to remote hosts on the network. The information includes the map list name and the destination IP address for local VCs.	<code>show atm map</code>

**Table 9-3. ATM Commands for Verifying Configuration**

Action	Command
Displays configuration and traffic shape settings and current state for all VCs in the system.	<code>show atm vc</code>
Displays ATM VC information. With the <code>vcd</code> option, the command adds packet and octet traffic statistics for the target VC.	<code>show atm vc vcd</code>
Displays the total number of PVCs configured throughout the system and then breaks that number down into the number of PVCs in each of the six possible states. If no PVCs are in a particular state, no count is shown for that state.	<code>show atm vc-summary</code>
Displays on/off status for <code>debug atm arp</code> and <code>debug atm packet</code> commands. It reports the severity settings that are currently in effect for the <code>debug atm errors</code> commands.	<code>show debugging</code>
Displays the contents of the Address Resolution Protocol (ARP) table.	<code>show ip arp</code>
Displays configuration information and status of IP interfaces listed in the system routing table. An IP interface is created using an <code>ip address</code> command.	<code>show ip interface</code>
Displays all ATM-related messages in the log buffer if you specify <code>show logging messages atm</code> . Displays just the messages in the log buffer that pertain to the ATM interface you specify, for example, <code>show logging messages atm4/1</code> .	<code>show logging messages atm</code>
Displays the list of logging categories and the number of messages logged for each.	<code>show logging registered</code>
Displays statistics pertaining to the SONET signals on a specified interface. This command also enables you to view SONET Line, Section, and Path errors that are not logged to the event log or displayed on the console.	<code>show sonet</code>



# Point-to-Point Protocol Configuration

Point-to-Point protocol (PPP), as described in RFC 1661, provides a standard method for transporting a variety of network-layer protocols over point-to-point links. PPP is designed for simple links that transport packets between two peers. The PPP connections are defined on a per-interface basis and set the encapsulation method and the Link Control Protocol (LCP) for connection management.

Relative to the Open Systems Interconnection (OSI) reference model, PPP provides layer-2 (data-link layer) service. PPP is a full-duplex protocol that can be used on various physical media. PPP runs on POS and serial interfaces and uses a variation of High Speed Data Link Control (HDLC) for packet encapsulation.

## Key Features

The PPP implementation in the system supports the following Network control protocols:

- IPCP for IP
- OSICP for IS-IS
- MPLSCP for MPLS

## Technology Concepts

Basic to the understanding of PPP are:

- Encapsulation
- Link Control Protocol
- Network Control Protocols

## Standards

The NX64000 system implementation of PPP is based in part on the following RFC:

RFC	Title
RFC 1661	The Point-to-Point Protocol (PPP)

## Encapsulation

PPP encapsulation multiplexes different network-layer protocols, simultaneously, over the same link. This encapsulation requires framing to indicate the beginning and end of the encapsulation. PPP encapsulation is compatible with most commonly used devices.

## Link Control Protocol

In order to be portable over a wide variety of environments, PPP provides a Link Control Protocol (LCP). The LCP automatically agrees upon the encapsulation format options, handles varying limits on sizes of packets, and detects routing loops and other common configuration errors. LCP is used to bring lines up, set negotiating options, test lines, and bring lines down when they are no longer needed.

## Network Control Protocols

PPP provides a family of Network Control Protocols (NCP) for establishing and configuring different network-layer protocols. If the link also uses a network protocol that also has an associated NCP, the associated network control protocol must be up for the link to pass route data. The supported NCPs are listed in [Table 10-1](#).

**Table 10-1. Supported Network Control Protocols**

Network-Layer Protocol	PPP Network Control Protocols
An IP address or IP unnumbered	IPCP — Internet Protocol (version 4) Control Protocol
IS-IS	OSICP — Network control protocol for IS-IS
MPLS	MPLSCP — MPLS Control Protocol

## Debugging and Logging Facilities

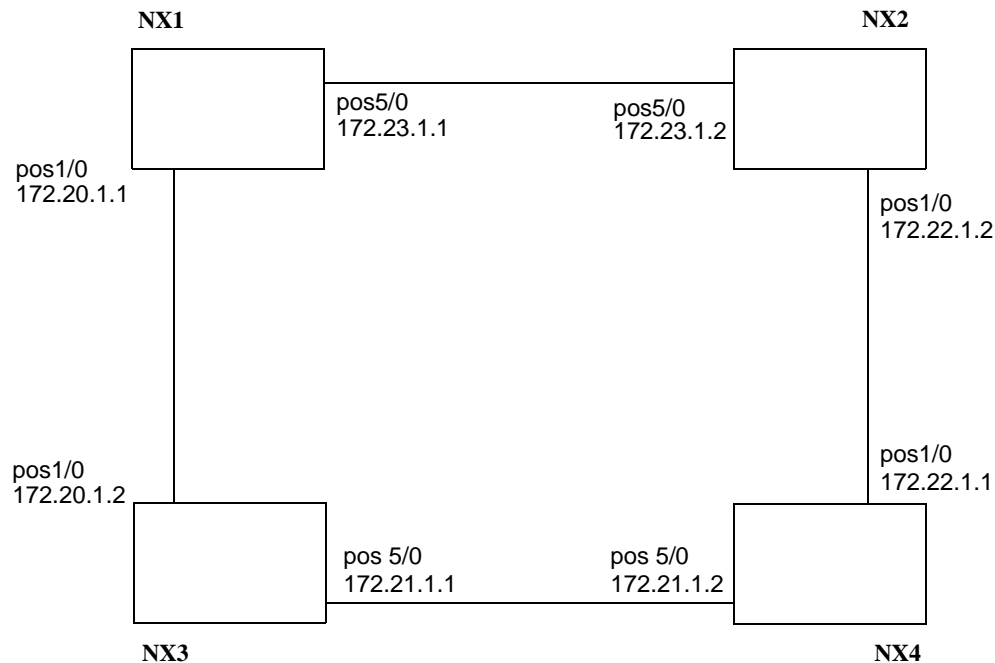
The NX-IS software implementation provides debugging and debug message-logging facilities to help you troubleshoot PPP problems. In addition, the `show ppp` command provides information to help locate a problem. For more information on diagnosing problems with PPP, see the *NX64000 Troubleshooting Guide*.

## PPP Interface Configuration

This section provides examples of basic PPP configuration options on the system. Basic PPP configuration requires:

- Accessing the interface and specifying the encapsulation type as PPP. Encapsulation must be specified before any packets can be transmitted.
- Configuring an IP address and network mask on the interface, or setting the interface to IP unnumbered to transmit data.

**Figure 10-1** illustrates a hypothetical PPP network that is used as the basis for the sample configuration tasks. The figure shows the sample setup of four NX64000 switch/routers, with the host names NX1, NX2, NX3, and NX4.



**Figure 10-1. PPP Sample Network**

Using this sample network, the following examples demonstrate how to create two PPP connections, one to each adjacent NX64000 switch/router.

NX1:

```
nx1# configure terminal
nx1(config)# interface pos5/0
nx1(config-if)# encapsulation ppp
nx1(config-if)# ip address 172.23.1.1 255.255.0.0
nx1(config-if)# exit
nx1(config)# interface pos1/0
nx1(config-if)# encapsulation ppp
nx1(config-if)# ip address 172.20.1.1 255.255.0.0
nx1(config-if)# exit
```

Perform the same configuration tasks for the other NX64000 routers in the sample network, but use the corresponding IP addresses and submasks as shown in **Figure 10-1 on page 10-3**.

The following table explains the use of the commands in the example above on configuring pos5/3 and pos1/0 on NX1:

Configuration Line	Description
<code>configure terminal</code>	Put the router into terminal configuration mode so you can enter system and interface settings.
<code>interface pos5/0</code>	Enter interface configuration mode for the POS interface on slot five, port 0.
<code>encapsulation ppp</code>	Set PPP as the encapsulation method on the interface. Traffic is not forwarded across the interface until the encapsulation method is set.
<code>ip address 172.23.1.1 255.255.0.0</code>	Specify the IP address and the subnet mask for the interface and enables IP on the source interface.
<code>exit</code>	Return to the previous prompt level (config) so you can access the pos1/0 interface and specify the encapsulation type.
<code>interface pos1/0</code>	Enter interface configuration mode for the POS interface on slot 1, port 0.
<code>encapsulation ppp</code>	Set PPP as the encapsulation method on the interface. No traffic is forwarded across the interface until the encapsulation method is set.
<code>ip address 172.10.1.1 255.255.0.0</code>	Specify the IP address and the subnet mask for the interface and enables IP on the source interface.
<code>exit</code>	Return to the previous (config) prompt level.

## Configuring PPP on an Unnumbered Interface

An unnumbered PPP interface does not have any network prefix associated with it and therefore, network interfaces connected to an unnumbered PPP interface do not have IP addresses.

When the unnumbered interface forwards a packet, it uses borrowed address as its source address. This method saves IP addresses by allowing a single address to represent multiple interfaces. The following example enables PPP encapsulation on interface pos2/1 and defines the interface as unnumbered:

```
nx# configure terminal
nx(config)# interface pos2/1
nx(config-if)# ip unnumbered loopback0
nx(config-if)# ppp peer-ip-route
```

The following table explain the use of the commands in the example above on configuring PPP on an unnumbered interface:

Configuration Line	Description
<code>configure terminal</code>	Put the router into terminal configuration mode so you can enter system and interface settings.
<code>interface pos2/1</code>	Enter interface configuration mode for the POS interface on slot two, port one.
<code>ip unnumbered loopback0</code>	Configure interface pos2/1 to use the loopback interface's IP address.
<code>ppp peer-ip-route</code>	Add router IDs of unnumbered interfaces to the routing table.

## Verifying and Monitoring PPP Connections

Use the following commands to verify and monitor the PPP connections.

**Table 10-2. PPP Commands for Verifying Configuration**

Action	Command
Display the parameter and configuration settings currently in running memory. Compare the output from this command and the output from the <code>show startup-config</code> command to isolate configuration changes made since the last save.	<code>show running-config</code>
Display the parameter and configuration settings saved to the startup configuration file. Compare the output from this command and the output from the <code>show running-config</code> command to isolate configuration changes made since the last save.	<code>show startup-config</code>
Verify connectivity by sending an echo request packet to a specified device and listening for an answer.	<code>ping</code>
Display summary information about the interfaces and protocols for a specific interface configured on the system.	<code>show interfaces</code>
Display PPP-related state machine information such as the link phase, the link control protocol (LCP) status, and the network control protocol status on all interfaces or a specified interface.	<code>show ppp</code>



# Internet Protocol Configuration

Internet Protocol (IP) is the network layer protocol that supports communication between different networks. IP provides addressing and control support for the routing of data packets. IP routing is the process of determining where to forward IP packets so that they can arrive at the destination address.

IP provides a connectionless, unreliable, best-effort packet delivery service. IP is considered to be connectionless because it does not guarantee delivery, which is the responsibility of the higher-layer protocols, such as Transmission Control Protocol (TCP).

IP includes information for fragmentation/reassembly and Time-To-Live and identifies the encapsulated protocol (for example, TCP). For TCP traffic, IP routers can drop packets to handle congestion (the sending router is supposed to detect the congestion by the high rate of retransmit requests from the receiver and slow down). The system provides quality of service support to manage congestion.

IP encompasses interior and exterior gateway protocols that route data across area networks. The system supports the following routing protocols:

- Border Gateway Protocol (BGP)
- Intermediate System-to-Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)
- Internet Protocol Multicast

IP executes two basic functions:

- Addressing
- Fragmentation

## Key Features

The NX64000 IP Core Router provides IP (version 4) core routing and backbone transport.

## Technology Concepts

Basic to the understanding of IP are:

- IP addressing
- Internet control protocols
  - Domain Name System
  - Address Resolution Protocol
- IP routing protocols
  - Border Gateway Protocol
  - Intermediate System-to-Intermediate System
  - Open Shortest Path First
  - IP Multicast Routing
- Differentiated Services

## IP Addressing

The system communicates with other devices using IP. You should be familiar with standard IP addressing and with the network's addressing scheme before configuring IP addresses for the interfaces.

Internet-connected computers use a numeric addressing system to locate other computers on the network. This numeric addressing system requires that each computer connected to the Internet have its own, unique, 32-bit numeric address—an Internet Protocol (IP) address.

An IP address is composed of two parts: the network number and the host number. It is expressed as four decimal numbers separated by periods, such as “123.123.123.123”. In this dotted decimal format, each section represents 8 bits of the IP address. Valid addresses range from 0.0.0.0 to 255.255.255.255, a total of about 4.3 billion addresses.

IP addresses are divided into five different classes, A, B, C, D and E. You can determine the class of any IP address by examining the first 4 bits of the IP address, the first octet.

**Table 11-1** lists the IP-address ranges and status by Class.

**Table 11-1. Reserved and Available IP Addresses**

Class	Address or Range	Status
A	0.0.0.0	Reserved
	1.0.0.0 through 126.0.0.0	Available
	127.0.0.0	Reserved
B	128.0.0.0	Reserved
	128.1.0.0 through 191.254.0.0	Available
	191.255.0.0	Reserved



**Table 11-1. Reserved and Available IP Addresses**

Class	Address or Range	Status
C	192.0.0.0	Reserved
	192.0.1.0 through 223.255.254	Available
	223.255.255.0	Reserved
D	224.000. through 239.255.255.255	Reserved for multicast traffic
E	240.0.0.0 through 247.255.255.255	Reserved for future use

## Subnet Bit Masks

A subnet is a segment of a network. Subnetting allows you to divide one physical network into smaller logical networks to control the flow of traffic, reduce network congestion and to improve manageability, performance and security. You can set up multiple logical networks that exist within a single Class A, B, or C network.

You use subnet masks to create subnets. The subnet mask identifies which portion of the IP address is the network ID and which portion is the host ID.

A subnet address is created by “borrowing” bits from the host field and designating them as the subnet field. The number of borrowed bits is specified in the subnet mask. Subnet masks use the same format and representation technique as IP addresses. The subnet mask has binary 1s in all bits specifying the network and subnetwork fields, and binary 0s in all bits specifying the host field.

Subnet masks allow devices to determine if traffic should be routed out of the network or kept within it. Using a mask saves the router from having to handle the entire 32-bit address. Instead, the router looks at the bits selected by the mask and ignores the host portion of the address.

The easiest way to view subnet masks is in the binary form where 1's represent which portion of the IP address is the network ID and 0's indicate which portion of the IP address is the host ID. For example:

### Class A Address

Mask	Net ID.Host ID.Host ID.Host ID
Binary	11111111.00000000.00000000.00000000
Decimal	255.0.0.0

### Class B Address

Mask	Net ID.Net ID.Host ID.Host ID
Binary	11111111.11111111.00000000.00000000
Decimal	255.255.0.0

### Class C Address

Mask	Net ID.Net ID.Net ID.Host ID
Binary	11111111.11111111.11111111.00000000
Decimal	255.255.255.0

## Access Lists

The NX64000 system supports standard and extended access lists. Standard access lists feature packets by the source IP address specified in the message header. The extended access lists allow the system, or associated interface, to selectively deny access to IP packets from a specified source address or addresses. Extended access lists establish a set of rules that classify IP packets based on the values contained in the IP and layer 4 headers. Creating or modifying an extended access list has no effect on the operation of the router until that particular list is referenced by part of the system configuration. For more information on configuring access lists, see [Chapter 12, “Access List Configuration.”](#)

## Internet Control Protocols

In addition to IP, which provides the packet transfer, the system provides the following control protocols.

- Domain Name System
- Address Resolution Protocol

### Domain Name System (DNS)

The Domain Name System is an Internet directory service, used to translate between domain names and IP addresses. The DNS allows clients and servers to communicate with each other.

DNS allows names or text to be used as IP addressees. The benefit is that it's easier for people to remember addresses if they are familiar, relational, or hierarchical, rather than strings of numbers. When an Internet user types in the domain name Lucent.com the DNS translates the domain name Lucent to the corresponding IP address. DNS name resolution is running on the system by default. You must specify one to six domain name servers for the system to resolve hostnames and IP addresses.

The system stores the names of the name servers in the same order in which you configure them. The system then queries the various servers in this order. That is, the system first queries the server configured first, next the server that was configured second and so forth.

### Address Resolution Protocol (ARP)

For two routers on a network to communicate, they must know the other's physical (or MAC) addresses. ARP defines a method of dynamically discovering the MAC-layer address corresponding to a particular IP network-layer address.

ARP broadcasts a packet, which contains the IP address of the destination, to all hosts attached to the network. Most hosts ignore the packet. The intended router, recognizing that the IP address in the packet matches its own, returns an answer.

After learning a MAC-layer address, IP devices store the recently acquired IP-to-MAC address mapping in the ARP table. When a host wants to communicate with a peer, the host first looks in the ARP table for the peer address. If an address is found, it is sent directly, otherwise ARP broadcasts. When the packet is sent, the intended device must respond within a specified period or the cache entry is discarded.

## IP Routing

IP uses the IP subnet mask and a routing table to route data to a destination. The system uses IP routing tables to decide where to send a packet. The routing table contains both dynamic and static routes.

A dynamic route is one that is learned. Dynamic routing occurs when routers talk to adjacent routers, using a common communication protocol, about the networks to which they are connected. The system searches its routing table, looking for host routes, network routes, and default routes. The information is updated in the routing table as routes change, instead of coming from route commands in bootstrap files. Dynamic routing allows the router to reroute packets around network failures by maintaining alternate routes.

A static route is manually configured. Static routes are maintained in the routing table during shut downs, restarts, and software reloads.

When defining a static route for a multi-access segment you must use the next-hop IP address. If it is point-to-point, then you may use the interface name instead of the next-hop IP address. When defining a static route for a Gigabit Ethernet interface you must specify the next-hop IP address. This is because Gigabit ethernet uses ARP to map a known IP address to a MAC address.

For example, you can specify a static route as:

```
nx# configure terminal
nx (config)# ip route 1.1.1.0 255.255.255.0 2.2.2.0
```

but not as:

```
nx# configure terminal
nx (config)# ip route 1.1.1.0 255.255.255.0 gigabitethernet1/1
```

## Routing Protocols

The NX-IS supports the following IP routing protocols.

- IP Unicast Routing
  - BGP4, OSPF v2, IS-IS
- IP Multicast Routing

## Border Gateway Protocol (BGP)

BGP is commonly accepted as the replacement to the now-defunct Exterior Gateways' Protocol (EGP). BGP bridges the gap between networks running different interior gateway protocols (for example, OSPF and IS-IS). For more information on configuring BGP, see [Chapter 18, "BGP Configuration."](#)

## Open Shortest Path First (OSPF)

The OSPF protocol is a link-state routing protocol that runs within an Autonomous System (AS). Link-state means that when there is a topology change, for example, a link goes down or a router is added to the network, a link-state advertisement (LSA) is sent to all directly connected routers. The routers update their tables and flood LSAs to their neighbors to ensure that all routers have the same topology information in their LSA database(s). For more information about configuring OSPF protocols, see [Chapter 16, “OSPF Configuration.”](#)

## Intermediate System-to-Intermediate System (IS-IS)

IS-IS protocol is an interior gateway protocol (IGP). IS-IS is designed to allow a single system to route both IP and Open Systems Interconnection (OSI) traffic. IS-IS is a single, integrated protocol that can route, by area, either IP, OSI, or “dual” traffic, and uses a shared backbone for the routing domain. Packets are forwarded without any type of encapsulation. On the NX64000 IP Core Router, only IP routing is supported. For more information on configuring the IS-IS protocol, see [Chapter 17, “IS-IS Configuration.”](#)

## Internet Protocol Multicast

Internet Protocol Multicast is a multicast routing method with control mechanisms for building multicast data distribution trees and forwarding multicast data down these trees. As the name implies, it is not dependent on the type of unicast protocol in use in the network, although it uses information from the unicast routing table. IP Multicast uses the following protocols:

- Protocol Independent Multicast (PIM)
- Internet Group Management Protocol (IGMP)
- Multicast Source Discovery Protocol (MSDP)

For more information about Internet Protocol Multicast, see [Chapter 15, “Internet Protocol \(IP\) Multicast Configuration.”](#)

## Differentiated Services

IP supports traffic prioritization by allowing packets to be labeled with an abstract type of service to define per-hop forwarding behaviors. For information about configuring Quality of Service on IP interfaces see [Chapter 13, “Quality of Service Configuration for Traffic Management.”](#)

## Debugging and Logging Facilities

This implementation provides debugging facilities to help you troubleshoot IP problems. The ping and the traceroute commands provide information to help locate problems. For more information on troubleshooting IP problems, refer to the *NX64000 Troubleshooting Guide*.

# IP Header Format

An IP datagram consists of two parts: header and text. The header is a 20 byte fixed part and a variable length optional part. The IP packet header consists of 20 bytes of data. The full header is shown in **Figure 11-1**.

Version	IHL	Type of Service	Total Length	
Identification			Control Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options				Padding

**Figure 11-1. The IP Header Format**

The following descriptions summarize the IP header fields.

**Table 11-2. Description of IP Header Fields**

Field	Description
Version 4 bits	Sets the IP version, the system supports version 4.
Internet Header Length (IHL) 4 bits	Contains the length of the Internet header. The minimum value is 5, which means no options are present. The maximum value is 15, which limits the header to 60 bytes.
Type of Service (TOS) 8 bits	Specifies the type of service. It is used by Differentiated Services, also called the Differentiated Services Code Point (DSCP). See <b>“Differentiated Services” on page 11-6</b> .
Total Length 16 bits	Defines the packet length (in bytes) including the header. The maximum size of an IP datagram is 65,535 octets.
Identification 16 bits	Specifies the unique number, assigned by the sender, to all fragments of a single datagram. The Identification field facilitates the reassembly of the fragmented datagrams.

**Table 11-2. Description of IP Header Fields**

Field	Description
Control Flags 3 bits	Consists of three flags set to control the fragmenting of a packet and to indicate the parts of a fragmented packet to the receiver.  Bits 0, 1, 2 are flags <ul style="list-style-type: none"><li>• bit 0 is reserved, it must be set to 0</li><li>• bit 1 DF: 0 = Fragment, 1 = Don't fragment</li><li>• bit 2 MF: 0 = Last fragment, 1 = More fragments</li></ul>
Fragment Offset 13 bits	Specifies a value for each data fragment in the reassembly process. Indicates the correct order of the datagram fields.
Time to Live (TTL) 8 bits	Limits the time that a packet is allowed to travel. Each IP packet contains a TTL field, which is decremented every time a router handles the packet. If the TTL field reaches zero, the packet is discarded, preventing looping.
Protocol 8 bits	Defines the type of transport protocol for the packet being carried. 1 = ICMP; 2= IGMP; 6 = TCP; 17= UDP
Header Checksum 16 bits	Verifies the header only. The header is verified at each point that the Internet header is processed, because at least one field (TTL) always changes.
Source Address 32 bits	Contains the IP address of the original packet sender.
Destination Address 32 bits	Contains the IP address of the final packet destination.
Options variable length	Provides features to allow the sender of a packet to set requirements on the path it takes through the network, trace the route a packet takes and labels packets with security features.
Padding variable length	Ensures that the Internet header ends on a 32-bit boundary. The padding is zero.

# Internet Protocol Configuration

This section provides examples of basic IP configuration tasks. The IP configuration commands affect or display the state of a specified component in the system. **Table 11-3** lists the IP commands that comprise the current implementation and their application in configuring and maintaining an IP network.

- ▶ Before you change an existing configuration, make sure you have a good working knowledge of the current configuration. Run the associated **show** commands to view information about the configuration.  
To preserve the current configuration, save the running configuration to the startup configuration file (**copy running-config startup-config**) before making changes.
- ▶ The *NX64000 Command Reference* manual describes all commands referenced in this chapter.

**Table 11-3. IP Commands Usage**

Command	Address Resolution	Configuration	IP addressing	Interoperation	IP Routing	Verification
access-group					✓	
access-list					✓	
clear arp-cache					✓	
clear ip route					✓	
ip address			✓			
ip domain-name	✓					
ip domain-lookup	✓					
ip mtu		✓				
ip name-server	✓					
ip route					✓	
ip unnumbered		✓				
nslookup	✓					

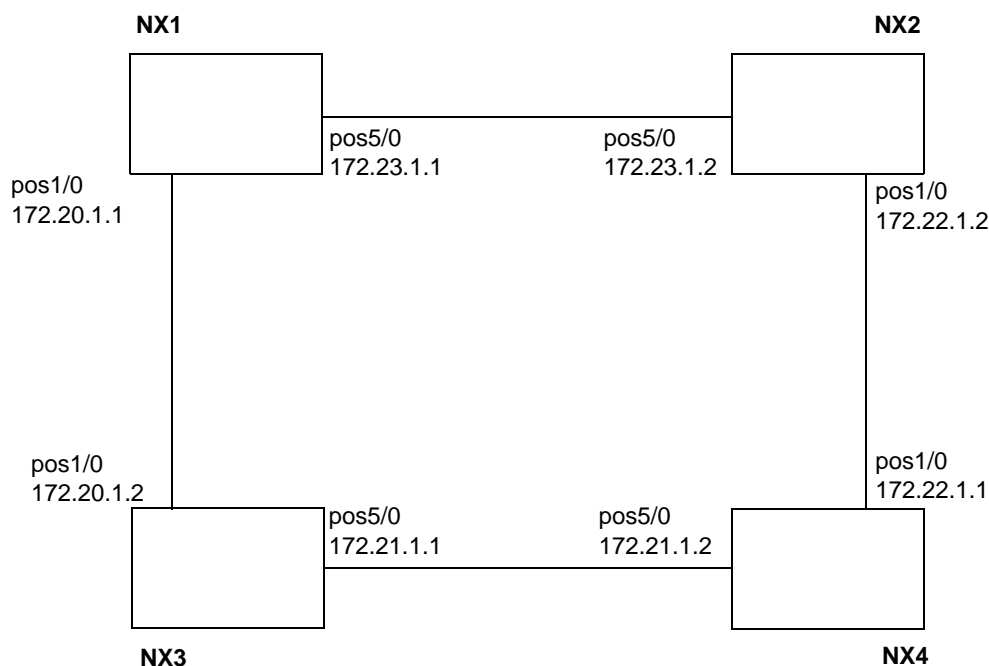
Command	Address Resolution	Configuration	IP addressing	Interoperation	IP Routing	Verification
ping						✓
show access-list						✓
show ip arp						✓
show ip as-path-access-list						✓
show ip interface						✓
show ip route						✓
show ip route summary						✓
show ip route supernets-only						✓
show ip rstats						✓
show ip traffic						✓
tracert						✓

## Configuration Overview

To configure IP you must assign an IP address to system interfaces, or configure an interface as IP unnumbered. Refer to the specific protocol chapters for detailed explanations of operation.

**Figure 11-2** illustrates an IP network that is used as the basis for the sample configuration tasks. The figure shows the sample setup of four systems, with the host names NX1, NX2, NX3, and NX4.





**Figure 11-2. IP Sample Network**

The following examples create two IP connections, one to each adjacent system. The examples use POS interfaces with PPP encapsulation. You specify IP addresses in the same manner for other interface types and for other POS interfaces using a different encapsulation type.

1. Configure pos1/0 and pos5/0 on NX1.

```

nx1# configure terminal
nx1(config)# interface pos5/0
nx1(config-if)# encapsulation ppp
nx1(config-if)# ip address 172.23.1.1 255.255.0.0
nx1(config-if)# exit
nx1(config)# interface pos1/0
nx1(config-if)# encapsulation ppp
nx1(config-if)# ip address 172.20.1.1 255.255.0.0
nx1(config-if)# exit
  
```

2. Perform the same configuration tasks for the other systems in the sample network, but use the corresponding IP addresses as shown in [Figure 11-2 on page 11-11](#). The following table explains each command line of the POS1/0 and POS5/0 configuration for NX1.

The following table explains the use of the commands in the example above:

Configuration Line	Description
<code>configure terminal</code>	Enter into terminal configuration mode. From here you can enter system commands and interface settings.
<code>interface pos5/0</code> <code>interface pos1/0</code>	Enter interface configuration mode for the specified interface on slot number/port number.
<code>encapsulation ppp</code>	Set PPP as the encapsulation method on the interface. Traffic is not forwarded across the interface until the encapsulation method is set.
<code>ip address 172.23.1.1 255.255.0.0</code>	Specify the IP address and the subnet mask for the interface and enable IP on the source interface.
<code>exit</code>	Return to the previous prompt level, the (config) prompt in this case.

## DNS Server IP Address Configuration Example

You can specify all six domain-name servers at one time, or you can configure them one at a time. The following example specifies two name servers that the system will query for name resolution:

```
nx# configure terminal
nx(config)# ip name-server 172.0.2.10 172.24.2.10
```

## Assigning an IP Address to the Ethernet Port

Two broadcast interfaces are predefined on the system loopback0 and ethernet0. The loopback0 interface is a software-defined interface for routing and testing; ethernet0 is the management Ethernet interface. For management purposes, an IP address is assigned to the Ethernet interface on the NX64000 IP Core Router during installation. You can change this IP address if needed.

The following example sets an IP address and network mask for the management Ethernet port:

```
nx# configure terminal
nx(config)# interface ethernet0
nx(config-if)# ip address 192.2.0.1 255.255.255.0
```

The following table explains the use of the commands in the example above:

Configuration Line	Description
<code>configure terminal</code>	Enter into terminal configuration mode.
<code>interface ethernet0</code>	Enter interface configuration mode for the management Ethernet interface, ethernet0.
<code>ip address 192.2.0.1 255.255.255.0</code>	Specify the IP address and the subnet mask for the interface and enable IP on the source interface.

- Do not set the IP address of the Ethernet management port to 10.0.100.74. The RCP uses this address for communications within the system.

## IP Unnumbered Configuration Example

Unnumbered IP allows you to send IP traffic over a serial line interface without assigning an IP network address. This feature allows you to configure static routes to the next hop router or to a default router. The following example specifies a point-to-point interface as IP unnumbered.

```
nx# configure terminal
nx(config)# interface pos2/0
nx(config-if)# encapsulation ppp
nx(config-if)# ip unnumbered loopback0
```

The following table explains the use of the commands in the example above:

Configuration Line	Description
<code>configure terminal</code>	Enter into terminal configuration mode. From here you can enter system commands and interface settings.
<code>interface pos2/0</code>	Enter interface configuration mode for the specified interface on slot number 2/port number 0.
<code>encapsulation ppp</code>	Set PPP as the encapsulation method on the interface. Traffic is not forwarded across the interface until the encapsulation method is set.
<code>ip unnumbered loopback0</code>	Set the specified interface to run IP without having to assign an IP address and configures the specified interface to use the loopback to simulate an interface that is always up.
<code>exit</code>	Return to the previous prompt level. The (config) prompt in this case.

## Changing an IP address on a Loopback Interface

You can set only one IP address on a loopback interface. If you want to change the IP address, enter the **no ip address** command first, then the **ip address** command with the new IP address.

The following example sets 192.0.2.2 as the primary address for the loopback interface, then uses the **no ip address** command to remove the address and mask from the interface and resets 172.0.2.2 as the primary address for the interface:

```
nx# configure terminal
nx(config)# interface loopback
nx (config-if)# ip address 192.0.2.2 255.255.255.0

nx (config-if)# no ip address 192.0.2.2 255.255.255.0
nx (config-if)# ip address 172.0.2.2 255.255.255.0
```

The following table explains the use of the commands in the example above:

Configuration Line	Description
<b>configure terminal</b>	Enter into terminal configuration mode. From here you can enter system commands and interface settings.
<b>interface loopback</b>	Enter interface configuration mode for the loopback interface.
<b>ip address 192.0.2.2 255.255.255.0</b>	Specify the IP address and the subnet mask for the interface and enable IP on the source interface.
<b>no ip address 192.0.2.2 255.255.255.0</b>	Remove the IP address and subnet mask from the interface. The <b>no</b> form of the <b>ip address</b> command, with no arguments specified, removes all addresses from the interface.
<b>ip address 172.0.2.2 255.255.255.0</b>	Specify the IP address and the subnet mask for the interface and enable IP on the source interface.

## Verifying and Monitoring IP Connections

The NX-IS software provides numerous commands to verify and monitor IP connections. Run the appropriate command, such as, **show ip interface** to verify the parameters that are currently configured prior to making any configuration changes. The commands are run from the **nx#** prompt. For more information about the commands see the *NX64000 Command Reference*.

**Table 11-4. IP Commands for Verifying Configuration**

Action	Command
Display the rules defined for an IP access list. The IP access list contains rules that classify IP packets based on the values contained in the IP and layer 4 headers.	<code>show access-list</code>
Display the contents of the Address Resolution Protocol table.	<code>show ip arp</code>
Display the autonomous system path access list information. You can display information for a specified access list or all configured access lists.	<code>show ip as-path-access-list</code>
Display configuration information and status of the IP interfaces listed in the routing table. Interfaces are dynamically entered and removed from the table as their status changes.	<code>show ip interface</code>
Display the entire routing table or those routes that match the specified optional variables.	<code>show ip route</code>
Display summary information for the routing table, sorted by protocol or type. Entries indicate how many routes of various types each entity has installed into the routing table.	<code>show ip route summary</code>
Display summary information about the aggregate routes listed in the routing table (that is, those created with the summary address command). An aggregate mask can be identified because its prefix is shorter than the natural network mask.	<code>show ip route supernets-only</code>
Display the number of entries (routes) in the IP routing table. This number includes the total number of active routes, which is derived from all IP protocol local routes, and static routes.	<code>show ip rstats</code>
Display IP traffic statistics such as IP statistics, Internet Control Message Protocol (ICMP) statistics, User Datagram Protocol (UDP) statistics, IP multicast statistics, Address Resolution Protocol (ARP) statistics and IP security statistics.	<code>show ip traffic</code>
Verify network connectivity status.  The <code>ping</code> command has both a line mode and an interactive mode. Execute the <code>ping</code> command without arguments to enter interactive mode. In this mode the system prompts you to enter values for extended attributes.	<code>ping</code>
Display the actual path from the local host to a remote destination.  Execute the <code>traceroute</code> command without arguments, to enter interactive mode. In this mode the system prompts you to enter values for extended attributes.	<code>traceroute</code>



## Access List Configuration

Access lists set criteria that define which packets the system accepts, thereby controlling the traffic that enters the network. Access lists, when assigned to a system interface, enable packet filtering for that interface. Configuration for Telnet, SNMP, Protocol Independent Multicast (PIM), Multicast Source Discovery Protocol (MSDP), and route filters also use access lists as a way to identify a set of IP addresses.

When a system interface receives packets, it compares the information in the IP header against the access criteria configured. The interface either forwards the packet if the packet meets the “permit” criteria, or discards the packet if it meets the “deny” criteria.

### Key Features

The system provides two types of access lists:

- Standard access lists to filter by source addresses
  - Extended access lists to filter by:
    - Source address
    - Destination address
    - TCP or UDP source port
    - TCP or UDP destination port
    - ICMP messages and codes
- Extended access lists also provide a logging capability.

### Technology Concepts

Basic to the understanding of access lists are:

- Access lists for protocol configuration
- Packet filters for traffic management
- Filtering process
- Filtering criteria
- Filter application

## Access Lists for Protocol Configuration

SNMP, multicast, and route filtering configuration can use an access list to specify a set of IP addresses. Typically these lists are either standard access lists or IP extended access lists.

## Packet Filters for Traffic Management

Packet filtering controls the traffic that enters the system and the associated network by accepting or rejecting packets according to rules configured in access lists. Typically, you use packet filtering to limit network access and to support network security policies.

Flexibility for assigning different access lists to different interfaces lets you filter traffic based on different sets of rules for different interfaces. Prior to configuring access lists, you need a good understanding of which packets should be accepted or rejected by each system interface.

### Limiting Network Access

Access filters applied to an interface let you limit traffic to decrease network load. Packet filters can reject ICMP packets, block network protocols, and prevent traffic for specified applications from entering the system. You can configure rules to limit system access to block traffic from certain protocols by specifying the port associated with that protocol. Filters can also reject packets to and from specified destinations.

### Supporting Network Security

Because access lists let you block types of traffic that could compromise network security, packet filtering can augment network security programs. If you want to use packet filtering for security, you should be familiar with the types of traffic that can cause security breaches and configure access lists to prevent those types of traffic from accessing your network.

## Filtering Process

The system evaluates the IP header of each packet it receives when the packet enters the system. If an interface has an assigned access list, the system performs a single check to determine whether or not a packet meets criteria to be admitted to the system. If the packet does not meet any criteria, the system drops the packet.

The system processes the packet header as if it is sequentially comparing header fields against the rules in the list. When it finds a match, it takes the specified action and stops evaluating the header against successive rules. Rules appear in an access list in the order in which they were configured.

## Filtering Criteria

A rule for an access list defines “permit” or “deny” criteria based on addressing and, for extended access lists, protocol information. The ordering of the rules in a list is important to the way the system evaluates rules. You use the `access-list` command to configure access rules for standard access lists and extended access lists.



The software adds a “deny-all” rule to the end of a configured access list. You do not need to configure this rule. Because the deny-all rule appears last, the interface or application associated with the list does not admit any packets to enter the system unless specified in a list rule.

## Standard Access Lists

Standard access lists contain rules to filter on the source IP address in the IP header. A rule can specify whether to allow or prevent access to packets from any IP address, a specific address, or a set of IP addresses. To specify a range of IP addresses, you use wild cards consisting of 0s and 1s. Ones (1s) indicate a field to be disregarded. When assigning a numeric identifier to an access list, the numbers 1 through 99 identify a standard access list.

## Extended Access Lists

Extended access lists support filtering on:

- Source address(es)
- Destination address(es)
- Protocol (IP, UDP, TCP, ICMP)

You can specify the source and destination addresses in a number of ways. The list accepts designations of any address, an individual IP address, or a range of IP addresses. To specify a range of IP addresses, you use wild cards consisting of 0s and 1s. Ones (1s) indicate a field to be disregarded.

Protocol-specific rules for TCP and UDP let you specify source ports and destination ports, or a descriptor that indicates the application associated with a port. The access-list command provides a set of operands to let you specify a set, or range, of ports. For information about supported values for ports, see the [“UDP Rules” on page 12-4](#) or [“TCP Rules” on page 12-5](#).

► For information about well-known port numbers see RFC 1349, *Assigned Numbers*.

Protocol-specific rules for ICMP let you specify an ICMP message type and ICMP code. For information about supported values for ICMP message types and ICMP codes, see [“ICMP Rules” on page 12-6](#).

When assigning a numeric identifier to an access list, the numbers 100 through 199 identify an extended access list.

## IP Rules

For IP rules you specify:

- An individual source address or a set of source addresses
- An individual destination address or a set of destination addresses

## UDP Rules

For UDP rules you specify:

- An individual source address or a set of source addresses
- An individual destination address or a set of destination addresses
- Source port numbers 0 through 65535, or a port descriptor (optional)
- Destination port numbers 0 through 65535, or a port descriptor (optional)

The access-list command supports the UDP port descriptors listed in the following table:

**Table 12-1. UDP Port Descriptors**

Protocol	Syntax as command argument	UDP Port Number
Biff (mail notification, comsat)	biff	512
Bootstrap Protocol (BOOTP) client	bootpc	68
Bootstrap Protocol (BOOTP) server	bootps	67
Discard	discard	9
DNSIX security protocol auditing	dnsix	195
Domain Name Service (DNS)	domain	53
Echo	echo	7
Internet Security Association and Key Management Protocol	isakmp	500
Mobile IP registration	mobile-ip	434
IEN116 name service (obsolete)	nameserver	42
Netbios datagram service	netbios-dgm	138
Netbios name service	netbios-ns	137
Netbios session service	netbios-ss	139
Network Time Protocol	ntp	123
Routing Information Protocol (router, in.routed)	rip	520
Simple Network Management Protocol	snmp	161
SNMP Traps	snmptrap	162
Sun Remote Procedure Call	sunrpc	111
Syslog	syslog	514

**Table 12-1. UDP Port Descriptors**

Protocol	Syntax as command argument	UDP Port Number
TAC Access Control System	tacacs	49
Talk	talk	517
Trivial File Transfer Protocol	tftp	69
Time	time	37
Who service (rwho)	who	513
X Display Manager Control Protocol	xdcmp	177

## TCP Rules

For TCP rules you specify:

- An individual source address or a set of source addresses
- An individual destination address or a set of destination addresses
- Source port numbers 0 through 65535, or a port descriptor (optional)
- Destination port numbers 0 through 65535, or a port descriptor (optional)

The access-list command supports the TCP port descriptors listed in the following table:

**Table 12-2. TCP Port Descriptors**

Protocol	Syntax as command argument	TCP Port Number
Border Gateway Protocol	bgp	179
Character generator	chargen	19
Remote commands (rcmd)	cmd	514
Daytime	daytime	13
Discard	discard	9
Domain Name Service	domain	53
Echo	echo	7
Exec (rsh)	exec	512
Finger	finger	79
File Transfer Protocol	ftp	21
FTP data connections (used infrequently)	ftp-data	20

**Table 12-2. TCP Port Descriptors**

Protocol	Syntax as command argument	TCP Port Number
gopher	gopher	70
NIC hostname server	hostname	101
Ident protocol	ident	113
Internet Relay Chat	irc	194
Kerberos login	klogin	543
Kerberos shell	kshell	544
Login (rlogin)	login	513
Printer service	lpd	515
Network News Transport Protocol	nntp	119
Post Office Protocol version 2	pop2	109
Post Office Protocol version 3	pop3	110
Simple Mail Transport Protocol	smtp	25
Sun Remote Procedure Call	sunrpc	111
TAC Access Control System	tacacs	49
Talk	talk	517
Telnet	telnet	23
Time	time	37
Unix-to-Unix Copy Program	uucp	540
Nickname	whois	43
World Wide Web (HTTP)	www	80

#### ICMP Rules

For protocol-specific rules for ICMP you specify:

- An individual source address or a set of source addresses
- An individual destination address or a set of destination addresses
- Message codes 0 through 255
- Message types as listed in [Table 12-3](#)

**Table 12-3. ICMP Message Types**

ICMP Message Types	Syntax as command argument
Echo (ping)	echo
Echo reply	echo-reply
Host isolated	host-isolated
Host unreachable for precedence	host-precedence-unreachable
Host redirect	host-redirect
Host redirect for type of service	host-tos-redirect
Host unreachable for type of service	host-tos-unreachable
Host unknown	host-unknown
Host unreachable	host-unreachable
Information replies	information-reply
Information requests	information-request
Logging option	log
Mask replies	mask-reply
Mask requests	mask-request
Network redirects	net-redirect
Network redirect for type of service	net-tos-redirect
Network unreachable for type of service	net-tos-unreachable
Network unreachable	net-unreachable
Network unknown	network-unknown
Fragmentation needed but the don't fragment (DF) bit is set	packet-too-big
All parameter problems	parameter-problem
Port unreachable	port-unreachable
Precedence cutoff	precedence-unreachable
Protocol unreachable	protocol-unreachable
All redirects	redirect

**Table 12-3. ICMP Message Types**

ICMP Message Types	Syntax as command argument
Router discovery advertisements	router-advertisement
Router discovery solicitation	router-solicitation
Source quench	source-quench
All time exceeded	time-exceeded
Time stamp replies	timestamp-reply
Timestamp requests	timestamp-request
ICMP Message Type (0 - 255)	<i>type</i>
All unreachables	unreachable

## Monitoring Filtered Traffic

Extended access lists provide a traffic monitoring capability to gather information about traffic received that meets a configured filtering rule. When you configure the rule for the access list, the **log** option enables logging for that rule.

Logging for a rule lets you see how many times within a five minute interval a system interface received a packet that meets the criteria configured for a rule. You can view the log messages through the log mechanism configured for the system, or you can run the **show logging** command for a card to see the notification messages. For information about viewing log messages for a card, see [Chapter 5, “Working with the Log and Debug Utilities.”](#)

## Filter Application

Standard and extended access list application varies by protocol. You can apply configured standard access lists to:

- Telnet sessions to limit management access to the Telnet daemon running on the system (access-class)
- The **snmp-server community** command
- Route filtering commands
- PIM and MSDP commands

You can apply configured extended access lists to:

- An interface for packet filtering (**ip access-group**)
- Route filtering commands (for IP)
- PIM and MSDP commands (for IP)

## Access List Activation on Interfaces

You apply a single extended access list to an interface to implement packet filtering. There is a delay from when you assign the access list to an interface, using the `ip access-group` command, until the access list becomes active on the interface. This is because the software is programming the card to evaluate the packet headers. In some cases, this process may be time consuming.

Log messages notify you when the system is processing the access list for an interface, and when the access list is active on the interface.

The following message indicates that the system is processing the access list:

```
2001-08-16 13:54:45  NOTICE SOURCE FILTCFG (tFiltCfg04    ) SLOT IOP_4
Compiling access-list rules. Will notify when filter is loaded...
```

The following message indicates that the access list is active on the interface:

```
2001-08-16 13:54:45  NOTICE SOURCE FILTCFG (tFiltCfg04    ) SLOT IOP_4
Filter is loaded on interface 4
```

You can view the log messages through the log mechanism configured for the system, or you can run the `show logging` command for a card to see the notification messages. For information about viewing log messages for a card, see [Chapter 5, “Working with the Log and Debug Utilities.”](#)

## Packet Formats

Access lists define which fields in the IP header the system uses to filter incoming packets. This section discusses only those header fields that access lists use.

## IP Header Fields

Standard access lists filter packets based on the source IP address. Extended access lists evaluate the source address, destination address, and protocol ID fields in the IP header. The shaded areas in **Figure 12-1** show the part of the IP header evaluated for configured access lists:

Version	IHL	Type of Service	Total Length	
Identification			Control Flags	Fragment Offset
Time to Live	Protocol		Header Checksum	
Source Address				
Destination Address				
Options				Padding

**Figure 12-1. IP Header Fields Used with Access Lists**

The following table describes the fields in the IP header that access lists use to filter packets:

**Table 12-4. IP Header**

Field	Description
Protocol	Eight bit field that identifies the upper-layer protocol.
Source address	Thirty-two bit field that identifies the address from which the packet was sent.
Destination address	Thirty-two bit field that identifies the destination to which the packet is being sent.



## TCP Header Fields

Extended access lists evaluate the source port and destination port for rules identifying TCP as the protocol. The shaded areas in **Figure 12-2** show the part of the TCP header evaluated for configured access lists:

Source Port	Destination Port
Sequence Number	
Acknowledgement Number	

**Figure 12-2. TCP Header Fields Used with Access Lists**

The following table describes the fields in the TCP header that access lists use to filter packets:

**Table 12-5. TCP Header**

Field	Description
Source Port	Sixteen bit field that identifies the connection type, or port, from which the packet was sent.
Destination Port	Sixteen bit field that identifies the connection type, or port, to which the packet is being sent.

## UDP Header Fields

Extended access lists evaluate the source port and destination port for rules identifying UDP as the protocol. The shaded areas in **Figure 12-3** show the part of the UDP header evaluated for configured access lists:

Source Port	Destination Port
Length	Checksum

**Figure 12-3. UDP Header Fields Used with Access Lists**

The following table describes the fields in the UDP header that access lists use to filter packets:

**Table 12-6. UDP Header**

Field	Description
Source Port	Sixteen bit field that identifies the connection type, or port, from which the packet was sent.
Destination Port	Sixteen bit field that identifies the connection type, or port, to which the packet is being sent.

## Access List Configuration

This section provides examples of basic access list configuration tasks. [Table 12-7](#) lists the access list commands and their application. The table does not list all of the protocol commands that specify access lists as an argument.

- ▶ Before you change an existing configuration, make sure you have a good working knowledge of the current configuration. Run the associated `show` commands to view information about the configuration.

To preserve the current configuration, save the running configuration to the startup configuration file (`copy running-config startup-config`) before making changes.

- ▶ The *NX64000 Command Reference* manual describes all commands referenced in this chapter.

**Table 12-7. Access List Command Usage**

Command	Rules configuration	Rules application	Verification
<code>access-list</code>	✓		
<code>access-class</code>		✓	
<code>ip access-group</code>		✓	
<code>show access-list</code>			✓

When you configure an access list, remember that the system evaluates the header against each rule in the order in which the rule appears in the list, and stops checking rules when it finds a match. The order in which you configure rules is the same order that the rules appear in the list. If a packet does not match any of the rules configured, the system discards the packet.

The following examples show brief access lists to demonstrate configuration principles. Access lists on a system may contain numerous rules. The last example shows how to apply an access list to an interface for packet filtering.

## Configuring a Standard Access List

The following example configures a standard access list:

```
nx# configure terminal
nx(config)# access-list 2 permit 192.0.2.20 255.255.255.0
nx(config)# access-list 2 permit 192.0.24.7 255.255.255.0
```

The following table explains the use of the access list commands in the preceding example:

Configuration Line	Description
<code>configure terminal</code>	Enter terminal configuration mode so you can enter system commands.
<code>access-list 2 permit 192.0.2.20 255.255.255.0</code>	Create access list 2 and add a rule to allow packets from 192.0.2.x.
<code>access-list 2 permit 192.0.24.7 255.255.255.0</code>	Add a rule to access list 2 to allow packets from 192.0.24.x.

## Configuring Extended Access Lists

For extended access lists, you specify the IP, TCP, UDP, or ICMP protocol and protocol-specific criteria for each rule.

### Setting Rules for IP, UDP, and ICMP Access

The following example shows how to add a number of rules to an extended access list that filters IP, ICMP, and UDP packets:

```
nx# configure terminal
nx(config)# access-list 103 ip host 192.0.2.24 any
nx(config)# access-list 103 ip host 192.0.24.0 any
nx(config)# access-list 103 ip any host 192.0.2.14
nx(config)# access-list 103 ip any host 192.0.24.0
nx(config)# access-list 103 icmp host 192.0.2.24 any
nx(config)# access-list 103 icmp host 192.0.24.0 any
nx(config)# access-list 103 icmp any host 192.0.2.14
nx(config)# access-list 103 icmp any host 192.0.24.0
nx(config)# access-list 103 permit udp any 53 any 53
```

The following table explains the use of the access list commands in the preceding example:

Configuration Line	Description
<code>configure terminal</code>	Enter terminal configuration mode so you can enter system commands.
<code>access-list 103 ip host 192.0.2.24 any</code>	Create access list 103 and add a rule to accept IP packets from 192.0.2.14 to any destination.
<code>access-list 103 ip host 192.0.24.0 any</code>	Add a rule to access list 103 to accept IP packets from 192.0.24.0 to any destination.
<code>access-list 103 ip any host 192.0.2.14</code>	Add a rule to access list 103 to accept IP packets from any source to 192.0.2.14.
<code>access-list 103 ip any host 192.0.24.0</code>	Add a rule to access list 103 to accept IP packets from any source to 192.0.24.0.
<code>access-list 3 icmp host 192.0.2.24 any</code>	Add a rule to access list 103 to accept ICMP packets from 192.0.2.24 to any destination.
<code>access-list 103 icmp host 192.0.24.0 any</code>	Add a rule to access list 103 to accept ICMP packets from 192.0.24.0 to any destination.
<code>access-list 103 icmp any host 192.0.2.14</code>	Add a rule to access list 103 to accept ICMP packets from any host to 192.0.2.14.
<code>access-list 103 icmp any host 192.0.24.0</code>	Add a rule to access list 103 to accept ICMP packets from any host to 192.0.24.0.
<code>access-list 103 permit udp any 53 any 53</code>	Add a rule to access list 103 to accept any UDP packet to or from a domain name server (port 53).

### Setting Rules for IP, TCP, UDP, and ICMP Access

The following example provides another implementation of an extended access list. This list allows any IP traffic to enter the interface, but filters out specified UDP and TCP packets and all ICMP packets:

```
nx(config)# access-list 104 deny tcp any 1025 any 1025
nx(config)# access-list 104 deny udp any 1025 any 1025
nx(config)# access-list 104 deny ICMP any any
nx(config)# access-list 104 ip any any
```

The following table explains the use of the access list commands in the preceding example:

Configuration Line	Description
<code>access-list 104 deny tcp any 1025 any 1025</code>	Create access list 104 and add a rule to access list 104 to deny any TCP packets to or from port 1025 on any system (Blackjack uses port 1025).
<code>access-list 104 deny udp any 1025 any 1025</code>	Add a rule to access list 104 to deny any UDP packets to or from port 1025 on any system (Blackjack uses port 1025).
<code>access-list 104 deny ICMP any any</code>	Add a rule to access list 104 to deny any ICMP packets to or from any system.
<code>access-list 104 ip any any</code>	Add a rule to allow access to incoming IP traffic from any source to any destination.

## Configuring an Extended Access List with Logging

The logging feature of extended access lists lets you collect data about the number of packets the system receives that match a specified rule. The following example filters on TCP packets to ports between 8383 and 9000 and enables logging for the rule:

```
nx (config)# access-list 109 permit tcp any host 192.0.2.5 gt 8384 lt 8999
log
```

The preceding configuration (when assigned to an interface) produces log entries like the following example entry:

Number of packets received for this rule

```
2000-09-07 22:00:32 INFO SOURCE FILTCFG (tFiltCfg09 ) SLOT IOP_9
Five Minute Report. 162490 hits for access rule:
permit tcp from 0.0.0.0 255.255.255.255 (10,65535) to 192.0.2.5 0.0.0.0
[8384,8889]
```

Source: any IP address

Source port: any 0-65535

Destination: 192.0.2.5

Destination Port: 8384 - 8889

If at the end of a 5-minute interval the interface did not receive any packets that matched the rule, the interval timer starts counting packets for the next 5-minute interval.

## Assigning Extended Access Lists to Interfaces

You set up packet filtering on a per-interface basis. The interface must have a configured IP address before you can assign an access list to the interface by running the `access-group` command. The following example assigns access list 103, configured above, to interface pos5/0, and access list 104 to interface pos1/0:

```
nx# configure terminal
nx(config)# interface pos5/0
nx(config-if)# ip address 172.23.1.1 255.255.255.0
nx(config-if)# ip access-group 103
nx(config-if)# exit
nx(config)# ip interface pos1/0
nx(config-if)# ip address 172.20.1.1 255.255.255.0
nx(config-if)# ip access-group 104
```

## Verifying Access List Configuration

The NX-IS software provides commands to get information about access list configuration. You can run the commands listed in the following table at the `nx#` prompt:

**Table 12-8. Access Lists Commands for Verifying Configuration**

Action	Command
View the rules for access lists configured on the system.	<code>show access-list</code>
Determine whether or not an access list is active on the system.	<code>show logging slot <i>slot-number</i> notify</code>

# Quality of Service Configuration for Traffic Management

Quality of Service (QoS) helps meet the challenges of today's networks by guaranteeing forwarding services for specified types of traffic. It provides different levels of service to deliver traffic, improving on the best-effort delivery by IP. By managing system resources and the traffic requirements, QoS provides support for networks that participate in service-level agreements between service providers and their clients.

QoS predictably manages configured traffic types. The system forwards prioritized traffic based on the bandwidth assigned to associated traffic flows. This mechanism helps to guarantee the throughput of high-priority traffic and minimize delay variation.

By classifying incoming traffic, QoS prioritizes packets for forwarding. High-priority traffic is guaranteed delivery with low delay. Low-priority traffic may experience forwarding delays, and is more likely to have packets discarded during episodes of congestion. QoS ensures service to all classes of traffic, sharing available bandwidth on a prioritized basis.

QoS provides bandwidth management and congestion control for each outgoing interface. The system dynamically manages interface bandwidth to efficiently transmit traffic. Should congestion occur, QoS configuration determines which traffic may experience delayed transmission and possible packet discard.

With the current state of the technology, creating an end-to-end QoS solution across a network is complex due to a lack of standardization across QoS implementations. The NX64000 implementation complies with the differentiated services (Diff-Serv) framework under consideration by the Diff-Serv working group in the IETF. Diff-Serv is a way to define different services for specified categories of traffic.

## Key Features

The quality of service implementation on the NX64000 IP Core Router supports:

- Classification of IP traffic by the source address, destination address, Diff-Serv (differentiated service services field), and precedence (Precedence is supported for backward compatibility with IP precedence configurations.) values in the IP header
- Priority queueing with bandwidth reservation
- Congestion management to control packet loss
- A traffic scheduling mechanism to increase throughput
- Connection admission control

► The NX-IS software also supports Diff-Serv over MPLS label-switched paths. For information about MPLS support for Diff-Serv, see [Chapter 14, “Multi-protocol Label Switching \(MPLS\) Configuration.”](#)

## Technology Concepts

Basic to the understanding of QoS are:

- Traffic classification
- Bandwidth management
- Congestion management
- Traffic scheduling
- Logging and debugging facilities

### Traffic Classification

Traffic classification is the process of assigning a priority to a traffic flow that has specified characteristics. The system can then forward prioritized traffic in different ways. For example, traffic sent from a specified address can be assigned a high priority to minimize transmission delay, and traffic sent to another address can be assigned a low priority to ensure best-effort delivery.

### Traffic Requirements

Planning a QoS implementation requires identifying the types of traffic that flow through the system to determine the transmission requirements for:

- Latency
- Predictability of delay
- Bandwidth, both constant rate and burst rate
- Tolerance for packet drop during congestion



The following table lists examples of delivery characteristics for groups of traffic that have special requirements:

Traffic Type	Delivery Characteristics
Interactive video	Predictable delay and latency
Voice	Predictable delay and latency (More tolerant of packet loss than video)
Traffic to a web site	Traffic bursts when popular content is published

## Priority Lists

A priority list defines rules used to classify traffic into different priority levels. The rules apply to traffic transmitted over an incoming interface. On the NX64000 IP Core Router, you can define priority lists based on the following attributes (from the IP header):

- **Destination address**—The destination IP address (and subnet mask) of the packet
- **Source address**—The IP address (and subnet mask) from which the packet originated
- **Diff-Serv**—A definition of different levels of services for transmitting traffic on a network

Diff-Serv associates marked traffic (traffic with a value set in the differentiated services code point field of the IP header) with a defined set of per-hop (or forwarding) behaviors. The per-hop behavior may also identify packets as preferred or non-preferred. During congestion, the system continues to forward traffic, but may drop Diff-Serv traffic marked as non-preferred. Typically, per-hop behaviors designate a service level.

► The guidelines for per-hop behaviors are still being standardized by the IETF.

- **IP precedence**—The relative importance of a packet on the network  
IP precedence was defined approximately two decades ago to map types to traffic, such as control traffic, traffic for immediate delivery, routine traffic, and so forth to specified delivery requirements.  
This mechanism is in wide use today, but is being replaced by Diff-Serv. Diff-Serv supports IP precedence for backward compatibility. Forwarding behaviors for IP precedence values are mapped to per-hop-behaviors for Diff-Serv.
- **Other**—A default priority list groups traffic that does not match other attributes, then assign the group a priority

## Priority Queues

A priority queue is a partition of bandwidth on an interface. The NX64000 IP Core Router supports eight priority queues on each outgoing interface. Queue 0 has the highest priority and queue 7 the lowest priority. The system transmits traffic by servicing priority queues in a fixed order, with the queues serviced successively from queue 0 to queue 7. The queue with the highest priority has the lowest delay and the lowest probability of dropping packets during congestion.

Assigning relative importance to different categories of traffic, and associating these groups with a comparable priority queue establishes how the system forwards traffic flows. The combination of the traffic type and priority defines the throughput characteristics of traffic grouped within a specified priority list.

The NX64000 system also lets you assign MPLS traffic or Frame Relay traffic for a specified DLCI to a QoS priority queue. For information about configuring Frame Relay DLCIs to send traffic to a specified QoS queue, see [Chapter 8, “Frame Relay Configuration.”](#)

The system sends all control traffic and routing protocol traffic to queue 0. This guarantees the delivery of control traffic at a low delay. Lucent Technologies recommends reserving queue 0 for these traffic types, and *not* configuring the system to send other types of traffic to queue 0.

The following table lists the default priority queues for specified types of traffic:

**Table 13-1. Default Priority Queues**

Traffic Type	Default Queue
Control traffic	0
Frame Relay	6
MPLS	6
Protocol-independent Multicast	5
Unclassified traffic that does not meet the criteria for preceding items	7

## Bandwidth Management

Bandwidth is the data transfer capacity for an interface. Line rate is the scheduling rate for outgoing packets. The NX64000 system uses a default line rate of 95% to schedule outgoing traffic. You can change this value if needed.

On interfaces configured to use QoS, you can provision bandwidth for each queue. The bandwidth assigned to a priority queue is referred to as reserved bandwidth because this bandwidth is always available to the assigned queue.

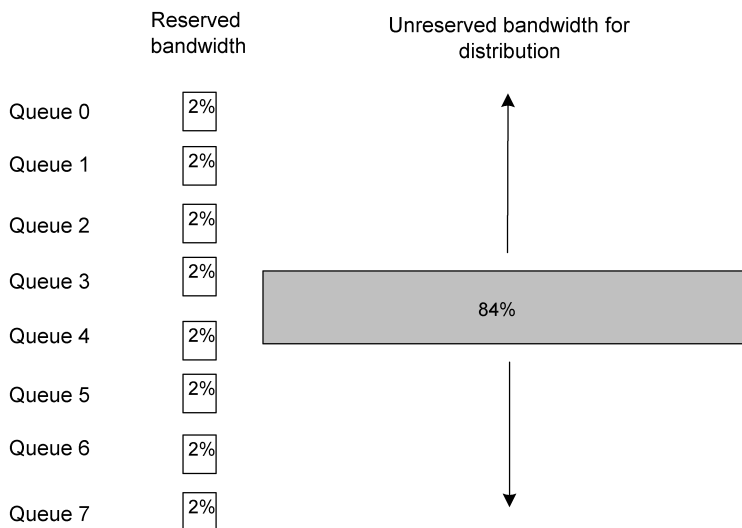
The value of each of the bandwidth parameters has a dependency on the values set for the others.

- Bandwidth allocated to Frame Relay DLCIs
- Bandwidth allocated to the priority queues
- Interface line rate

Configuration of each of these values must take into consideration the values set for the others.

## Queue Bandwidth

You should set the queue bandwidth to at least 1% for each queue to ensure that the QoS Manager can service traffic for any queue. By default, the QoS Manager assigns 2% bandwidth to each priority queue. This way, each queue can forward traffic. The remainder of the bandwidth is available for allocation to other queues on a strict priority as needed. The following illustration shows the default bandwidth configuration for an interface.



**Figure 13-1. Default Bandwidth Configuration**

Bandwidth allocation is important to how the system services the different priority queues. The NX64000 system services queues on a priority basis using the configured reserved bandwidth for each queue, then the unreserved bandwidth and bandwidth available from underutilized queues to provide additional bandwidth to queues with overflow traffic. By dynamically managing bandwidth, the system helps ensure throughput of prioritized traffic.

If an interface supports Frame Relay, MPLS, or multicast traffic, ensure that the default queue for the specified traffic type has sufficient bandwidth to service traffic. For a description of the default queues on the system, see [Table 13-1 on page 13-4](#).

## Protocol Overhead

You can tune how to efficiently use interface bandwidth by setting a protocol overhead for encapsulation protocols, that is layer-2 protocols such as Frame Relay or ATM. To set the protocol overhead you need to know the byte overhead for the layer-2 configuration on the system.

The system uses a default value of 16. In most cases this is the maximum overhead a protocol would add to an ATM cell or Frame Relay frame. The number of bytes assigned to the protocol overhead is important because traffic is scheduled for dispatch by number of bytes, not number of packets.

## Congestion Management

Congestion management is a mechanism to avoid congestion and ensure traffic throughput. Should episodes of congestion occur, the system can reject connections or drop packets to ensure the throughput of high-priority traffic. The following mechanisms help to avoid congestion on an interface, and to manage congestion should it occur.

### Connection Admission Control (CAC)

During connection setup, the system determines whether or not it can provide the requested quality of service for the connection. The system admits or rejects connections based on the available bandwidth on the outgoing interface and the requirements to meet QoS guarantees.

With CAC enabled, the system prevents allocation of queue bandwidth that exceeds the bandwidth available on an interface. The system factors all resource allocations, including those for Frame Relay DLCIs, into the total reserved bandwidth.

If CAC is disabled, the total bandwidth allocated to a queue (configured bandwidth plus bandwidth allocated to Frame Relay DLCIs) can be the same as the scheduling capacity of the interface (which has a dependency on the line rate setting).

- Interfaces have CAC enabled by default.

### Congestion Watermark

The system selectively drops packets destined for a specified priority queue on an interface when that queue reaches a configured size. With a congestion watermark set, the system drops the packets at the input interface, thereby controlling congestion for the output interface.

Setting a congestion watermark prevents congestion on the output queue. During congestion, the system is more likely to drop nonpreferred, or out-of-profile traffic.

### Weighted Random Early Detection (WRED)

The system randomly drops packets when a priority queue reaches a configured minimum threshold, then drops all packets when the queue reaches a configured maximum threshold.

- On an interface, you either set a watermark *or* use WRED to manage congestion.  
For congestion watermark and WRED, lower drop thresholds for lower priority queues help ensure throughput for the high-priority queues.

Before configuring a congestion control mechanism for an interface, you should be familiar with the maximum queue size for interfaces on the type of card being configured. [Table 13-2](#) lists the maximum queue sizes:

**Table 13-2. Maximum Queue Size for Card Types**

Card Type	Maximum Queue Size (in packets)
OC-3 ATM, OC-3 POS, DS 3	512
OC-12 POS, QOC-48 POS	1024
OC-12 ATM	2048
OC-48 POS	4096
OC-192 POS	4096
GigabitEthernet	2048

## Packet Discard

Packet headers (marked for classification) identify which packets the system is more likely to drop during periods of congestion. For Diff-Serv, the priority tag value (as configured on an NX64000 system) classifies a packet as preferred (in profile) or nonpreferred (out of profile). Nonpreferred packets have a greater probability of being dropped during congestion.

## Traffic Scheduling

When an input interface receives packets, the QoS Manager reads packet headers to classify packets based on the information in the header (DS-byte, source address, destination address, IP precedence). The classification assigns a priority level that identifies the priority queue on an output interface. For more information about priority queues, see [“Traffic Classification” on page 13-2](#).

The QoS Manager then evaluates the bandwidth for the destination queue on a specified interface to determine whether to forward the packet. During episodes of congestion, it might drop packets, otherwise it forwards packets to the destination queue.

When traffic reaches a queue on an output interface, the scheduling module schedules packets for transmission from the system.

The system’s scheduling module schedules queued traffic depending on:

- The queue priority
- The reserved bandwidth configured for each queue
- The number of packets assigned to each queue for forwarding

The scheduling module uses a prioritized fair queueing algorithm, a variant of weighted fair queueing, to service outgoing traffic. It dispatches queued traffic in the following way:

1. Checks each queue to determine traffic bandwidth requirements and the reserved bandwidth configured.
2. Distributes any available bandwidth to queues requiring additional bandwidth—starting with queue 0 and progressing through the queues in priority order to queue 7.

Unreserved bandwidth plus any free bandwidth resulting from underutilized queues comprises the available bandwidth.

3. As traffic departs from the queues, the system redistributes available bandwidth to queues using the distribution scheme described above.

Traffic scheduling becomes important when traffic congestion occurs on the interface either due to traffic bursts or sustained volume. Congestion causes both transmission delay and packets loss (resulting from the system dropping packets).

### Examples of Traffic Scheduling

The following examples show how the scheduling module shares bandwidth among priority queues to meet traffic demands, illustrating **step 1** and **step 2** above.

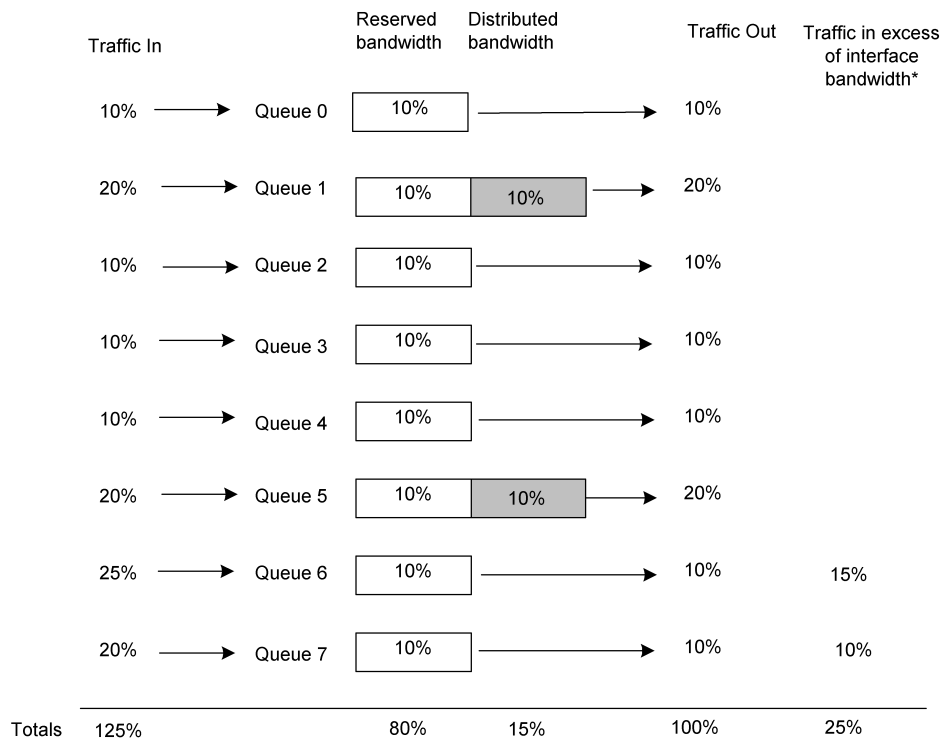
- To demonstrate bandwidth usage, the examples show the traffic in and out as a percentage of the scheduling capacity (100%) on the outgoing interface.

#### Example 1: Traffic Scheduling on an Oversubscribed Interface

This example illustrates how the system schedules traffic when the bandwidth required by incoming traffic is greater than the total bandwidth (shown as 100%).

- Each priority queue is assigned 10% reserved bandwidth.  
This leaves 20% of the bandwidth available for distribution. During congestion, this 20% is available to queues with overflow, or excess, traffic.
- The system distributes the unreserved bandwidth by priority, with 10% going to queue 1 to meet traffic requirements, then the remaining 10% to queue 5 to meet that queue's requirements.
- Queue 6 and queue 7 still have overflow traffic. With no additional bandwidth available, the 25% traffic overflow may cause packets to be delayed or dropped.

The following illustration shows the queue bandwidth configuration, and how the system distributes unreserved bandwidth:



\* May experience packet delay or drop

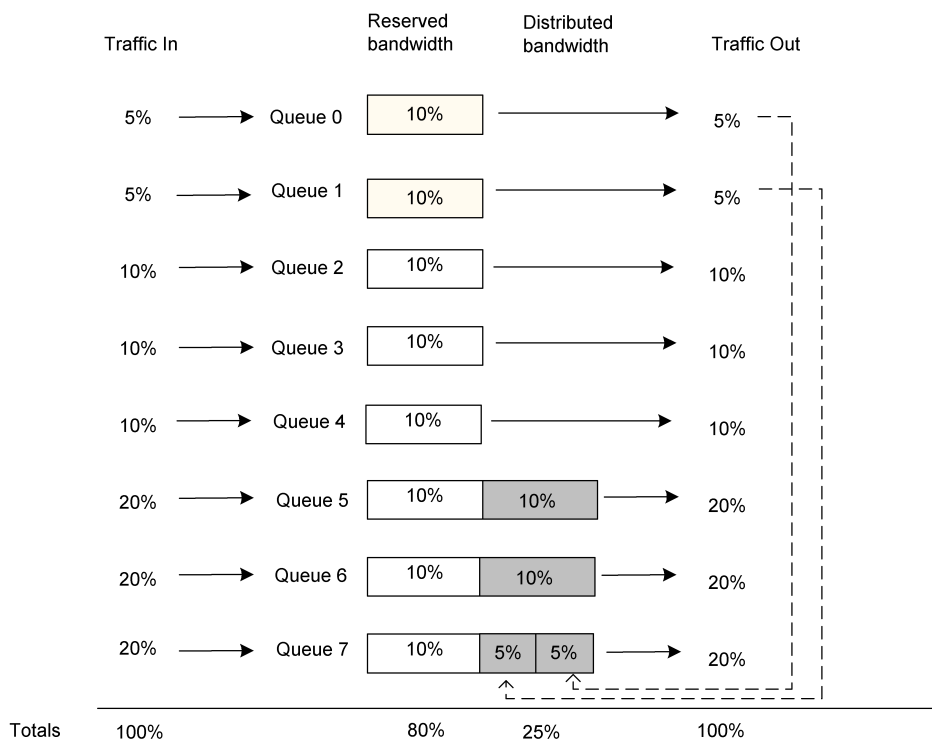
**Figure 13-2. Example: Traffic Scheduling for Oversubscribed Queues**

#### Example 2: Traffic Scheduling by Sharing Bandwidth from Undersubscribed Queues

This example illustrates how the system allocates unused queue bandwidth to other queues that have inadequate bandwidth.

- Each priority queue is assigned 10% reserved bandwidth.  
This leaves 20% of the bandwidth available for distribution. During congestion, this 20% is available to queues with overflow, or excess, traffic.
- Queues 0 and 1 are using only half of the reserved bandwidth for the queue. This makes the other half of the bandwidth (in this case 5% for each queue) available for distribution to other queues requiring additional bandwidth.
- A total of 30% of the bandwidth is available for distribution: the 20% bandwidth not allocated, plus the 5% from queue 0 and the 5% from queue 1.

The following illustration shows bandwidth distribution during a time when three queues are congested, but two queues have unused bandwidth:



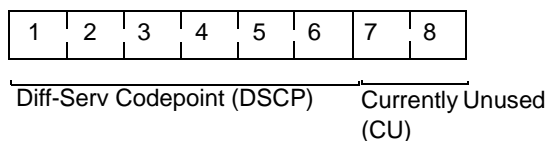
**Figure 13-3. Example: Bandwidth Sharing from Undersubscribed Queues**

## Debugging and Logging Facilities

The NX-IS software provides logging and debug commands for QoS. In addition, output from `show queue-statistics`, other QoS show commands, and the `show interfaces detail` command provides information to help troubleshoot QoS problems. For information about troubleshooting packet transmission problems, see the *NX64000 Troubleshooting Guide*.

## Differentiated Services Field

QoS uses the differentiated services field, also called the Diff-Serv byte or DS byte, of the IP header to specify the per-hop behavior for the packet. The DS byte consists of a 6-bit Diff-Serv Codepoint (DSCP) and 2 bits currently unused (CU):



**Figure 13-4. Diff-Serv Services Field**



The DS-byte is intended to replace the Type-of-Service (TOS) byte previously used by IP precedence. The first three bits of the Diff-Serv field mark IP precedence values for backward compatibility. For detailed information about this change and about how you can use codepoints definitions on your network, see RFC 2474, *Definition of the Differential Services Field (DS Field) in the IPv4 and IPv6 Headers*.

## QoS Configuration

The following sections illustrate a basic QoS configuration. Typically, QoS configuration is the same across contiguous NX64000 routers in a path. The interfaces may vary, but the set of QoS commands would remain the same. The sample configuration that follows could be applied to specified interfaces on a series of systems.

Keep in mind that although priority queues exist on outgoing interfaces, you configure priority lists for incoming interfaces. This way the system classifies incoming traffic before it forwards traffic to an outgoing interface. A priority list defines the rules applying to traffic transmitted over the incoming interface.

- If you are changing an existing configuration, make sure you have a good working knowledge of the current configuration before making changes. Run the associated `show` command to view information about the configuration. For information about the commands you can run to get information about QoS configuration, see [Table 13-4 on page 13-17](#).

For more information about the QoS commands, see the *NX64000 Command Reference*.

The following table lists the QoS commands and their application:

**Table 13-3. QoS Command Usage**

Command	Congestion control	Resource management	Priority assignment	Verification
<code>cac</code>	✓			
<code>congestion-watermark</code>	✓			
<code>line-rate</code>		✓		
<code>no priority-list all</code>			✓	
<code>priority group</code>			✓	
<code>priority-list default</code>			✓	
<code>priority-list diff-serv</code>			✓	

**Table 13-3. QoS Command Usage**

Command	Congestion control	Resource management	Priority assignment	Verification
priority-list precedence			✓	
priority-list source-addr			✓	
protocol-overhead		✓		
queue-bandwidth		✓		
random-detect-enable	✓			
show priority-group				✓
show priority-list				✓
show queue-bandwidth				✓
show queue-size				✓
show queue-statistics				✓
show random-detect				✓

## Basic QoS Configuration Tasks

When configuring QoS on the system, you can create a set of priority lists to define traffic forwarding characteristics, then apply those priority lists to a number of interfaces. The following lists describes the basic tasks to configure QoS:

- Create priority lists.



You can assign only one priority list to an interface, but each priority list can contain a number of rules. All rules must be of one type, such as destination address or source address.

- Assign a priority list to an incoming interface.
- Configure a congestion management mechanism (congestion watermark or WRED) for the priority queues on an outgoing interface.
- Reserve bandwidth for the priority queues on an outgoing interface, if appropriate.

The following illustration shows the interface traffic classification and bandwidth allocation used in the following configuration examples. The example assumes the default line rate of 95%.

Traffic type clasified on the incoming interface		Percentage of reserved bandwidth on the outgoing interface	
Control messages	Queue 0	2%	
Expedited delivery	Queue 1	10%	
Frame Relay priority 2	Queue 2	10%	
Destination address 192.0.24.....	Queue 3	5%	43% unreserved bandwidth
Destination address 192.0.2...	Queue 4	5%	
Source address 192.0.2....	Queue 5	5%	
Frame Relay default	Queue 6	10%	
Other - does not match any classification criteria	Queue 7	10%	

**Figure 13-5. QoS Configuration Example**

## Creating a Priority List

A priority lists lets you define traffic classes and assign them to priority queues.

In this example, priority list 1 specifies a traffic type for destination addresses to assign traffic to queues 3 and 4, and sets the default queue to 7:

```
nx# configure terminal
nx(config)# priority-list 1 dest-addr 192.0.24.10 255.255.255.0 3
nx(config)# priority-list 1 dest-addr 192.0.2.4 255.255.255.0 4
nx(config)# priority-list 1 default 7
```

The following table explains the use of the QoS commands in the example above:

Configuration Line	Description
<code>priority-list 1 dest-addr 192.0.24.10 255.255.255.0 3</code>	Assign traffic with a destination of 192.0.24.x to queue 3.
<code>priority-list 1 dest-addr 192.0.2.4 255.255.255.0 4</code>	Assign traffic with a destination of 192.0.2.x to queue 4.
<code>priority-list 1 default 7</code>	Assign traffic not matching other classification criteria to queue 7.

## Deleting Priority Lists

You can delete an entire priority list, or a specific entry configured for a priority list.

The following example deletes all of the items in priority list 1:

```
nx# configure terminal
nx(config)# no priority list 1
```

You can also delete individual entries on a list by entering the **no** form of the command used to add the item to the list. For example, to delete the item from priority list 1 that sets a priority of three for traffic sent to the address 192.0.24.x (created in [“Creating a Priority List”](#)), you enter:

```
nx# configure terminal
nx(config)# no priority-list 1 dest-addr 192.0.24.10 255.255.255.0 3
```

## Assigning a Priority Group on an Interface

This example shows how to use the **priority-group** command to assign the priority list configured in [“Creating a Priority List” on page 13-13](#) to interface pos2/0.

```
nx# configure terminal
nx(config)# interface pos5/3
nx(config-if)# encapsulation frame-relay
nx(config-if)# ip address 172.21.2.11
nx(config-if)# priority-group 1
```

## Configuring WRED on an Outgoing Interface

This example enables WRED and sets parameters for the priority queue configured to use WRED. The interface has CAC enabled by default.

```
nx# configure terminal
nx(config)# interface pos1/0
nx(config-if)# encapsulation frame-relay
nx(config-if)# ip address 172.21.1.1
nx(config-if)# priority-group 1
nx(config-if)# random-detect enable 7
nx(config-if)# random-detect priority-queue 7 512 1024
```

The following table explains each configuration entry specific to QoS:

Configuration Line	Description
<b>random-detect enable 7</b>	Enable WRED on queue 7.
<b>random-detect priority-queue 7 512 1024</b>	Set the two threshold parameters for WRED for queue 7. WRED becomes active at 512 packets, and drops all packets when the queue size reaches 1024 packets.

## Configuring Queue Bandwidth

Each outgoing interface supports eight priority queues. By configuring the traffic types for each queue, and reserve bandwidth to ensure the system adequately services the queues.

On interfaces supporting Frame Relay DLCIs, run the `show queue-bandwidth` command before configuring queue bandwidth to determine:

- The percentage of the bandwidth allocated to Frame Relay DLCIs
- The programmed line rate

To establish the bandwidth available to the priority queues, subtract the bandwidth allocated to the Frame Relay DLCIs from the programmed line rate.

The following example shows no bandwidth allocated to Frame Relay DLCIs, making the 95% line rate (default) available for queue bandwidth configuration. The output shows the default queue bandwidth allocation of 2% per queue.

```
nx# show queue-bandwidth pos5/0
Bandwidth Allocation on Interface (slot 5/port 0).
Connection Admission Control (CAC)           : ENABLED
Programmed Line Rate for (pos5/0)            : 2280 Mbps (95%)
Priority      Queue BW Alloc.      FR DLCI BW Alloc.      Total BW Resv.
Queue        % (Mbps / % )        Mbps      ( % )        Mbps      ( % )
-----
0            2 ( 45 / 1.97)        0      ( 0.00)        45      ( 1.97)
1            2 ( 45 / 1.97)        0      ( 0.00)        45      ( 1.97)
2            2 ( 45 / 1.97)        0      ( 0.00)        45      ( 1.97)
3            2 ( 45 / 1.97)        0      ( 0.00)        45      ( 1.97)
4            2 ( 45 / 1.97)        0      ( 0.00)        45      ( 1.97)
5            2 ( 45 / 1.97)        0      ( 0.00)        45      ( 1.97)
6            2 ( 45 / 1.97)        0      ( 0.00)        45      ( 1.97)
7            2 ( 45 / 1.97)        0      ( 0.00)        45      ( 1.97)
-----
Summary      16 ( 360 / 15.79)      0      ( 0.00)        360      ( 15.79)
```

This example reserves bandwidth for queues 1 - 7 as listed in the following table. Queue 0 has the default bandwidth of 2%. The 57% total bandwidth reservation is less than the 95% available:

Queue	Bandwidth reserved
1	10%
2	10%
3	5%
4	5%
5	5%
6	10%
7	10%

```
nx# configure terminal
nx(config)# interface pos5/0
nx(config-if)# queue-bandwidth 1 10
nx(config-if)# queue-bandwidth 2 10
nx(config-if)# queue-bandwidth 3 5
nx(config-if)# queue-bandwidth 4 5
nx(config-if)# queue-bandwidth 5 5
nx(config-if)# queue-bandwidth 6 10
nx(config-if)# queue-bandwidth 7 10
```

In each of the preceding configuration lines, the command identifies the queue first, then the bandwidth percentage assigned to the queue.

- If you try to allocate bandwidth greater than line rate capacity to a queue, the system displays the following message:

```
Bandwidth Allocation Failed: queue # is trying to over-subscribe
bandwidth on Interface (slot #, port #).
```

If you try to allocate bandwidth beyond the available bandwidth, the system displays the following message:

```
CAC Failure: Requested bandwidth ( percentage = consequent Mbps)
is not available on Interface (slot #, port #).
```

## Changing the Line Rate

By default, the line rate for an interface is 95%. In most instances, you do not need to change the line rate setting.

### To change the line rate:

1. Run the `show queue-bandwidth` command.
2. Calculate the total allocated bandwidth (total configured queue bandwidth plus the total bandwidth allocated for Frame Relay DLCIs).
3. Use the `line-rate` command to set the line rate.

This value must be the same or greater than the total allocated bandwidth in step 2.

- If you attempt to change a line rate to a value less the bandwidth already committed for priority queues and Frame Relay DLCIs, the system returns an error message:

```
Failure: new line rate is too small to support existing bw
reservation.
```

## Verifying QoS Configuration and Monitoring Traffic Transmission

The NX-IS software provides a number of show commands that let you verify configuration and assess statistics for QoS. The following table lists the type of information you can view with each command:

**Table 13-4. QoS Commands to Verify Configuration and Monitor Traffic**

Action	Command
Display information about the priority rules configured for an interface.	<code>show priority-group</code>
Display priority list rules.	<code>show priority-list</code>
Display the bandwidth of the priority queues on an interface.	<code>show queue-bandwidth</code>
Display information about the watermark threshold and queue size for priority queues on a specified interface.	<code>show queue-size</code>
Display the current queue statistics for an interface.	<code>show queue-statistics</code>
Displays data about WRED for an interface, sorted by queue number.	<code>show random-detect</code>





# Multi-protocol Label Switching (MPLS) Configuration

MPLS is a label-switching mechanism that forwards network traffic over paths configured on top of existing network topology. It lets you set up fixed paths between specified routers. MPLS works in conjunction with layer-2 link protocols and layer-3 routing protocols. MPLS, as a technology, continues to be defined by the IETF Multi-protocol Label Switching working group.

This implementation of MPLS integrates with the following interior gateway protocols (IGPs):

- Intermediate System-to-Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)

Integrating MPLS paths into a routing scheme lets you control traffic flows through parts of a network, and simplify routing configuration by using labels to configure these paths. The IGP considers an MPLS path as a single hop. Manipulating the metric or cost for a path supports traffic engineering operations on a network.

MPLS also supports Diff-Serv to provide service classes for forwarding traffic. Configuring Diff-Serv for an MPLS path establishes a Diff-Serv forwarding treatment for configured MPLS paths.

## Key Features

The MPLS implementation in NX-IS version 1.7 supports:

- Label-switched paths (LSPs) over point-to-point links
- Static LSPs with each hop in the path configured (including label values) from the LSP entry point to the LSP exit point
- RSVP-signaled LSPs configured from an ingress router
- Integration with IS-IS and OSPF
- LSP traffic sent by default to quality of service (QoS) priority queue 6 on an interface
- Penultimate-hop-popping to optimize processing at an egress label-switched router (LSR)
- LSP support for Diff-Serv service classes

## Technology Concepts

Basic to understanding of MPLS are:

- MPLS labels
- Label-switched routers
- Label-switched paths
- MPLS packet forwarding
- Label switching
- Integration with routing protocols
- Service classes for LSPs
- Debugging and logging for MPLS

## RFCs and Standards

This implementation of MPLS is based on the following standards and Internet drafts:

Document	Title
Internet Draft: draft-ietf-mpls-arch-07.txt	Multiprotocol Label Switching Architecture
Internet Draft: draft-ietf-mpls-label-encaps-08.txt	MPLS Label Stack Encoding
Internet Draft: draft-ietf-mpls-rsvp-lsp-tunnel-07.txt	RSVP-TE: Extensions to RSVP for LSP Tunnels
RFC 2836	Per Hop Behavior Identification Codes, May 2000
RFC 2475	An Architecture for Differentiated Services, Dec 1998
RFC 2474	Definition of the Differentiated Services Field (DS field) in the IPv4 and IPv6 Headers
Internet Draft: draft-ietf-mpls-diff-ext-07.txt	MPLS Support of Differentiated Services, August 2000

## MPLS Labels

For PPP packets, an MPLS label is a short identifier (20 bits) that appears after the link-layer header. The label has significance locally between two routers on an MPLS network where label-switching is the underlying mechanism for forwarding packets. For more detailed information about MPLS labels, see **“Packet Format” on page 14-11**; for more detailed information about the packet-switching process see **“Label Switching” on page 14-6**.

## Label-switched Routers

Label-switched routers are systems configured to run MPLS. Segments of an LSP connect a series of LSRs in an MPLS domain.

LSRs are classified as ingress, transit, and egress. The MPLS functions an LSR performs determines its type:

- Ingress—The LSR at the entrance to the MPLS domain. This router originates an LSP.
- Transit—The LSRs that form cross connects within an MPLS domain. Cross connects connect incoming and outgoing LSP segments.
- Egress—The LSR at the exit from the MPLS domain. This router terminates an LSP.

An LSR can function as any or all of these types, depending on the configuration. Typically, a single LSR acts as an ingress and egress LSR. The classification underscores the unidirectional nature of LSPs.

The terms upstream and downstream refer to the position of an LSR relative to another LSR in an LSP. A downstream LSR is one closer to an LSP destination. An upstream LSR is one closer to the origination of an LSP.

## Label-switched Paths

A label-switched path, sometimes referred to as a tunnel, is a path that passes through MPLS-enabled routers from the entry point to the exit point of an MPLS domain. Each end point becomes a single hop in an IP route, with the hops within an LSP visible only to MPLS. The sections of an LSP between two LSRs are referred to as LSP segments. Label-switched paths form the basis of MPLS.

LSPs transmit traffic in *only one direction* from point to point. Transmitting traffic from an entry point to an exit point and back, for example from A to B and from B to A, requires two LSPs—one in each direction. When an inbound and outbound LSP with the same end points are configured on the same tunnel interface it creates a bidirectional tunnel. With the end points the same, the LSPs can run over the same or a different topology.

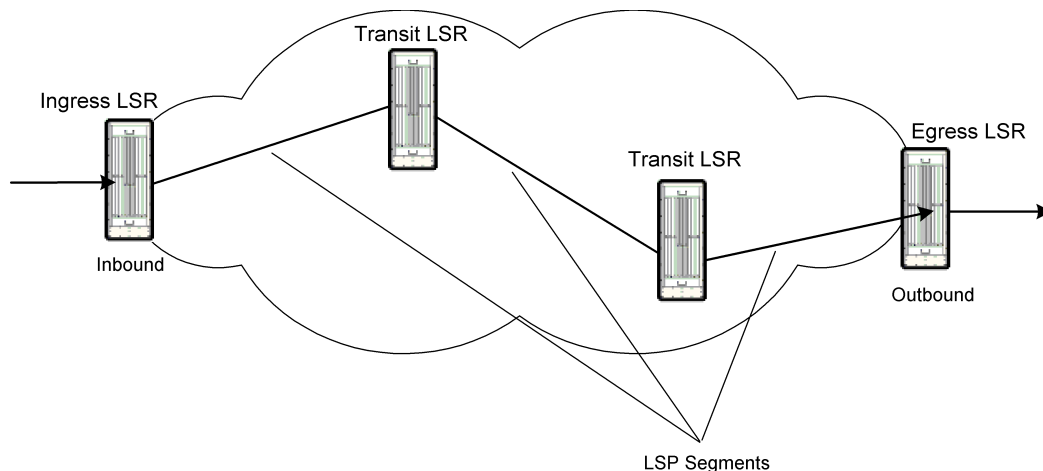
You can configure up to 1,500 paths on the system due to a wide label range (0 to 1,048,575) for PPP interfaces, and the ability to reuse labels that provides up to 400 ingress labels and 400 egress labels per interface.

### Path components

Each path consists of:

- An entry point to an LSP (the LSP origination) on an edge router
- Segments of an LSP between two routers within an MPLS domain
- Cross connects within a router between segments
- An exit point from an LSP (the LSP termination) on an edge router

The following illustration shows a sample MPLS domain:



**Figure 14-1. MPLS Domain**

### Path types

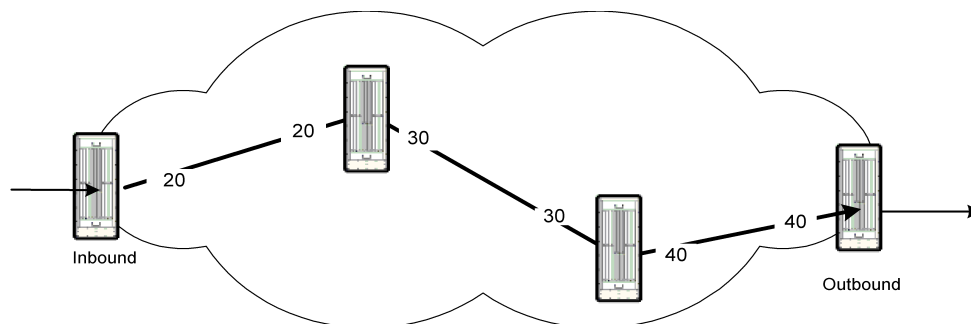
MPLS forwards traffic over both static and RSVP-signaled paths.

### Static Paths

Static paths have each hop configured from the systems's CLI, including values for associated labels. When you configure label values, the value set for the outgoing label on one system matches the one set for the incoming label on an adjacent system.

Typically, you configure the LSRs in a path in order from the egress router to the ingress router. By configuring the ingress LSR last, you ensure that the ingress system does not attempt to send traffic over the path before the path configuration is complete. If you do configure the ingress router first, you should ensure that the system does not send traffic over the tunnel interface before the entire path is configured. Traffic entering an LSP that is not completely configured is dropped.

The following illustration shows configured label values for a sample LSP:



**Figure 14-2. Label Values on an LSP**

## RSVP-signaled Paths

RSVP-signaled paths (also referred to as simply signaled paths in this chapter) rely on the Resource Reservation Protocol for Traffic Engineering (RSVP-TE, also referred to as simply RSVP in this chapter) to set up an MPLS path, manage labels for the path, and keep track of path status. If a hop is not accessible, RSVP sends an error message.

Configuration on the ingress LSR defines the explicit path for a signaled LSP. The remainder of the systems in the path must have MPLS and RSVP enabled on them. Full path configuration on one system simplifies maintenance because you make changes on only one LSR to align path changes with network updates.

The RSVP protocol must be activated from the ingress LSR to make the LSP available for packet transmission. The explicit path should be configured, and all systems in the path enabled with MPLS and RSVP, before you turn on RSVP signaling for the path. Otherwise, RSVP generates error messages.

## MPLS Packet Forwarding

MPLS forwards traffic from end-to-end over an MPLS network by manipulating the MPLS labels appended to IP packets. MPLS simplifies packet forwarding because it deals only with the short MPLS label, rather than a lengthy header.

MPLS switches traffic along a path by manipulating labels in the following way:

- **At entry**—When packets enter an MPLS domain, MPLS applies a label. For PPP connections, this label is part of a shim header.
- **At a cross connect**—When packets with an MPLS label arrive at a transit router, MPLS examines the label on the incoming packet, swaps labels to apply an outgoing label, and forwards the packet out of a specified cross-connected interface.
- **At exit**—MPLS forwards packets as directed by the IP header.

LSPs carry traffic routed by IS-IS or OSPF (or a static route) as configured. The LSPs essentially provide a short-cut for a route, with the entry point acting as one hop and the exit as another hop. Setting the cost, or metric, for an LSP helps balance traffic loads by determining whether traffic uses the LSP, based on the cost setting.

## Interfaces

MPLS uses both logical and physical interfaces to manage labels and forward traffic through the system. If an interface in the LSP is inoperative, traffic cannot travel over the associated LSP.

## Tunnel Interfaces on Ingress and Egress

Central to the MPLS implementation on the NX64000 system are the logical interfaces, called tunnel interfaces, that transmit traffic on ingress and egress LSRs. On ingress LSRs, the tunnel interface applies an MPLS label to a packet and transmits the packet to an associated LSP. On egress LSRs, the tunnel interface receives packets from an associated LSP, and forwards the packet as directed by the IP packet header. Each tunnel interface can support two LSPs, one incoming and one outgoing.

Configuring logical interfaces of the type tunnel includes setting the mode to MPLS then binding the tunnel interface to a physical interface. The physical interface manages packets as they arrive and leave the system. The physical interface must have MPLS enabled, and for signaled LSPs, must have RSVP enabled.

### Interfaces for Cross Connects

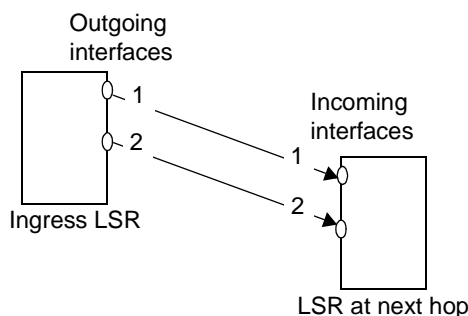
Cross connects rely on physical interfaces on the system. These physical interfaces have MPLS as well as RSVP enabled for RSVP-signaled paths.

## Label Switching

Manipulation of the label on MPLS packets controls packet forwarding over an LSP. Either manual configuration or RSVP-TE (consistent with the LSP configuration) assigns labels values for specified interfaces. These labels have significance only to contiguous LSRs along an LSP.

**On an ingress LSR**—An ingress LSR inserts a label on arriving packets. The value of this label is the same as the value for the inbound label at the next hop in the LSP.

The following illustration shows adjacent LSRs with label values the same at opposite ends of a segment:

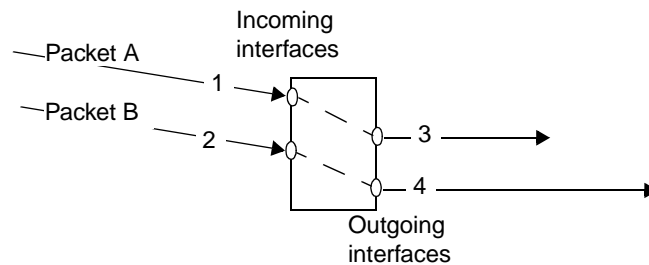


**Figure 14-3. Label Values on Adjacent LSRs**

**On a transit LSR**—Transit LSRs switch the labels to forward packets along a specific path within an MPLS domain. Each router switches the label on a packet as it passes the packet along. The label on an arriving packet has the same value as the label for the inbound interface. The value of the label changes at the outbound interface to the value set for the inbound label at the next hop. The label at the incoming interface determines the outgoing label and outbound interface.

The following illustration shows label-switching at a cross connect:

- Packet A arrives with label 1, and exits with label 3.
- Packet B arrives with label 2 and exits with label 4.



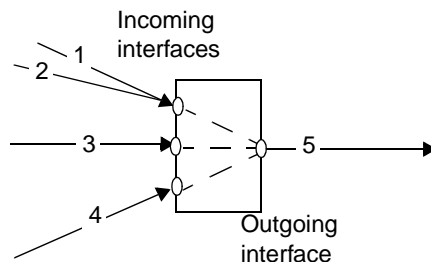
**Figure 14-4. Label Switching at a Cross Connect**

**On an egress LSR** — This implementation of MPLS supports penultimate-hop-popping to optimize processing at the terminating LSR. With penultimate-hop-popping, the second-to-last router strips the label as the packet passes to the last system. The egress LSR then only passes the packet along according to its IP destination address.

## Path Merge

MPLS supports forwarding traffic that arrives with different labels on one or more interfaces out of a single interface with the same label. In this way, the system can forward packets received from different sources to the same LSR. Merging paths is typically called a label merge because packets coming into the system with different labels exit with the same label.

The following illustration shows packets arriving with four different labels on three different interfaces, but all exiting with the same label:



**Figure 14-5. Label Merge**

For static LSPs, you create a label-merge on a system by configuring cross connects so that a number of paths with different incoming labels exit from the system with the same outgoing label.

## Integration with Routing Protocols

This implementation of MPLS supports integration with IS-IS and OSPF. When an IGP routes traffic through an LSP, the IGP views the entire LSP as one hop. This simplifies routing configuration. A routing area can combine hop-by-hop routes with LSPs that represent a single hop.

When routing packets through an IGP, the IGP recognizes an active tunnel interface for an LSP. It then forwards traffic through the LSP as indicated by the relative cost, or metric, of the path. An LSP can also be part of a static route.

## Traffic Engineering

Traffic engineering is a management strategy to help balance traffic load among systems in a network. You can use MPLS in conjunction with an IGP to support traffic engineering on a network by:

- Assigning a low metric, or cost, to an LSP to ensure that the IGP sends traffic over a specified path, rather than a hop-by-hop route.
- Assigning a high cost to ensure that traffic typically does not travel over a path, but make the path available as a backup should other systems be inaccessible.
- Setting up a path as a manual adjacency to limit which traffic flows use an LSP.

## LSP Mode

When you configure an IGP to use an LSP, you can set the mode for the LSP to manage protocol messages in one of the following ways:

- **Private**—Uses the LSP as a manual adjacency. Prevents the IGP from sending protocol messages over the LSP. The IGP does not advertise the LSP. An inbound LSP should be configured as private on a tunnel interface that does not also support an outbound LSP.
- **Normal (OSPF only)**—Lets the IGP operate as it would on other system interfaces. Traffic flows from end-to-end over one LSP and back over a second LSP, with the same end points specified. This configuration supports sending protocol messages as traffic flows in both directions.
- **Public (IS-IS only)**—Uses the LSP as a manual adjacency, and includes the LSP in link state advertisements. Prevents IS-IS from sending protocol messages over the LSP.

## Exchanging Routing Control Packets

OSPF support for normal mode LSPs allows the exchange of routing control packets. This exchange requires a bidirectional LSP, that is, a pair of LSPs configured to use the same end points. One LSP routes packets in one direction, and the other LSP routes packets in the opposite direction. OSPF must be enabled on the tunnel interface on each router.

## Service Classes for MPLS LSPs

In the core of an MPLS domain, MPLS LSPs can be configured on NX64000 systems to support Diff-Serv. Diff-Serv associates marked traffic with a defined set of per-hop (or forwarding) behaviors. These forwarding treatments define classes of service for packet transmission. A consistent configuration across the systems in the path ensures designated services, and creates an MPLS-Diff-Serv domain.

## Per-hop Behavior Groups

A per-hop behavior group is set of per-hop behaviors that share the same attributes. A per-hop behavior defines the forwarding treatment for specified traffic. Different values for the attributes assigned to the per-hop behaviors in a group distinguishes one per-hop behavior from another.



The NX-IS software supports a strict priority per-hop behavior group (SP-PHB) and an assured throughput per-hop behavior group (AT-PHB). Both of these groups are specific to the NX-IS software. These groups rely on the following attributes to define the members of the group:

- Priority
- Bandwidth reservation
- Congestion control—by enabling weighted random early detection or congestion watermarks

#### Strict Priority Per-hop Behavior Group

The strict priority per-hop behavior group contains per-hop behaviors that specify traffic forwarding based on the traffic priority. The system forwards packets with the highest priority before packets with lower priorities. It also supports congestion control, but *not* bandwidth reservation.

The SP-PHB group defines eight per-hop behaviors. The members of the group are: spphb1 (highest priority) through spphb 8 (lowest priority). Each of the eight per-hop behaviors maps to a priority queue and to a value in the EXP (experimental) field of the MPLS header. This EXP value defines a class of service:

Per-hop Behavior	Priority Queue	EXP Value
spphb1	0	0
spphb 2	1	1
spphb 3	2	2
spphb 4	3	3
spphb 5	4	4
spphb 6	5	5
spphb 7	6	6
spphb 8	7	7

#### Assured Throughput Per-hop Behavior Group

The assured throughput per-hop behavior group contains per-hop behaviors that:

- Fairly allocate interface bandwidth for forwarding all traffic in the group
- Ensure throughput
- Provide distribution of available bandwidth by strict priority from an underutilized queue for a PHB

The AT-PHB group defines eight per-hop behaviors. The members of the group are: atphb 1 (highest priority) through atphb 8 (lowest priority). Because the AT-PHB group supports only a single class of service, there is no mapping to the EXP field. The MPLS label carries information about the per-hop behavior. Each of the eight per-hop behaviors maps to prioritized traffic queues:

Per-hop Behavior	Priority Queue
atphb 1	0
atphb 2	1
atphb 3	2
atphb 4	3
atphb 5	4
atphb 6	5
atphb 7	6
atphb 8	7

## Diff-Serv Map Classes

A map class assigns one or more per-hop behaviors, or a per-hop behavior group, to an LSP. If a map class contains more than one per-hop behavior, a single group (either the SP-PHB group or the AT-PHB group) must include all of the behaviors. Map class assignments also identify whether the associated LSP supports a single class of service or more than one class of service. Map classes that support a single class of service have one entry. Map classes that support multiple classes have a number of entries.

## Single-class Service

LSPs assigned a single class of service, that is a single per-hop behavior, transmit traffic using one defined class. This type of service is often used when aggregating LSPs to ensure that all of the traffic receives the same forwarding treatment. Single-class service supports either AT-PHB or SP-PHB.

MPLS labels provide the forwarding information for single-class service. This type of service is also referred to as L-LSP (label-only-inferred-psc LSP, where psc is a per-hop behavior scheduling class).

## Multiclass Service

LSPs assigned a multiclass of service transmit traffic assigned to a number of classes, that is, multiple per-hop behaviors. All of the per-hop behaviors must be members of one group. This type of service lets one LSP support up to eight per-hop behaviors. Multiclass service supports only SP-PHB.

The multiclass service is also referred to as E-LSP (EXP-inferred-psc LSP where EXP represents the EXP field in the MPLS header and psc is a per-hop behavior scheduling class).

## Quality of Service for MPLS LSPs

You should configure either class of service or QoS for an interface.

When you enable MPLS on an interface, you can specify a priority queue and set the drop preference for preferred or non-preferred traffic. Any QoS setting takes precedence over service class settings for an LSP that traverses that interface. The default queue for MPLS traffic is queue 6.

## Debugging and Logging for MPLS and RSVP

The NX-IS software provides logging and debug commands for MPLS and RSVP. In addition, show commands plus the ping command can be used to troubleshoot LSP connections. For more information about troubleshooting MPLS problems, see the *NX64000 Troubleshooting Guide*.

## Packet Format

MPLS inserts a shim header for PPP links. This header contains:

- A 20-bit label
- A 3-bit field reserved for experimental use

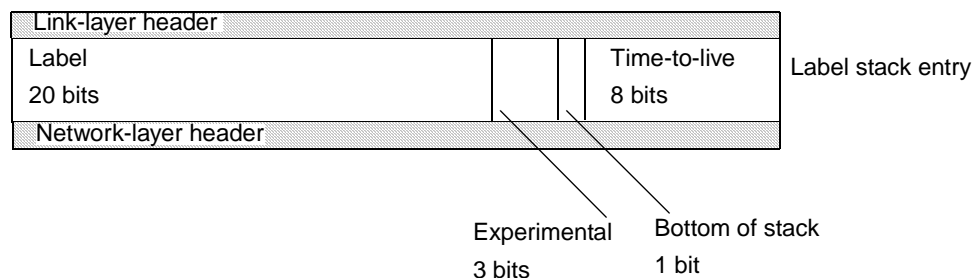
The strict priority per-hop behavior group uses the 3-bit experimental field to set class of service.

- A 1-bit bottom of the stack space

This value is set to one for the last entry in a label stack, and zero for all other label stack entries.

- An 8-bit time-to-live (TTL) field

The MPLS label appears directly after a link-layer header. The bottom of the label stack appears directly before the network layer packet. MPLS was created to support multiple levels of label stacking to allow more than one level of encapsulation. The implementation in NX-IS currently supports one level of labels, and therefore one level of LSP termination.



**Figure 14-6. MPLS Header Format for PPP Packets**

## MPLS Configuration

The following sections illustrate two basic configurations, one for setting up static paths and another for setting up RSVP-signaled paths. The sample configurations also show how to integrate MPLS LSPs into IS-IS or OSPF networks, and how to set up service classes for an LSP.

The examples illustrate the different configurations required to originate an LSP, terminate an LSP, and set up LSP cross connects. The inbound and outbound LSPs in the examples use the same end points, a fairly typical configuration that simplifies maintenance.

- ▶ If you change an existing MPLS configuration, make sure you have a good working knowledge of the current configuration before making changes to it. Run the associated **show** commands to view information about the configuration. For information about the commands you can run to get information about MPLS configuration, see [Table 14-2 on page 14-27](#).
- ▶ For more information about the MPLS commands, see the *NX64000 Command Reference*.

The following table lists the MPLS commands and their application:

**Table 14-1. MPLS Command Usage**

Command	Static LSP	Signaled LSP	Integration with IGPs	Diff-Serv	Verification
<b>diffserv enable group</b>				✓	
<b>diffserv enable phb</b>				✓	
<b>group</b>				✓	
<b>interface</b>	✓	✓			
<b>ip explicit-path</b>		✓			
<b>ip ospf mpls tunnel</b>			✓		
<b>ip router isis</b>			✓		
<b>list</b>		✓			
<b>map-class diffserv</b>				✓	
<b>mpls cross-connect</b>	✓			✓	

**Table 14-1. MPLS Command Usage**

Command	Static LSP	Signaled LSP	Integration with IGP	Diff-Serv	Verification
mpls enable	✓	✓		✓	
mpls lsp signal		✓		✓	
mpls lsp static	✓			✓	
next-address		✓			
phb				✓	
replace-entry		✓			
rsvp diffserv class		✓		✓	
rsvp diffserv error		✓		✓	
rsvp enable		✓			
show diffserv group					✓
show diffserv mapclass					✓
show diffserv phb					✓
show ip explicit-path					✓
show mapclass					✓
show mpls cross-connects					✓
show mpls in-segments					✓
show mpls interfaces					✓
show mpls lsps					✓
show mpls out-segments					✓
show rsvp interface					✓
show rsvp neighbor					✓
show rsvp session					✓

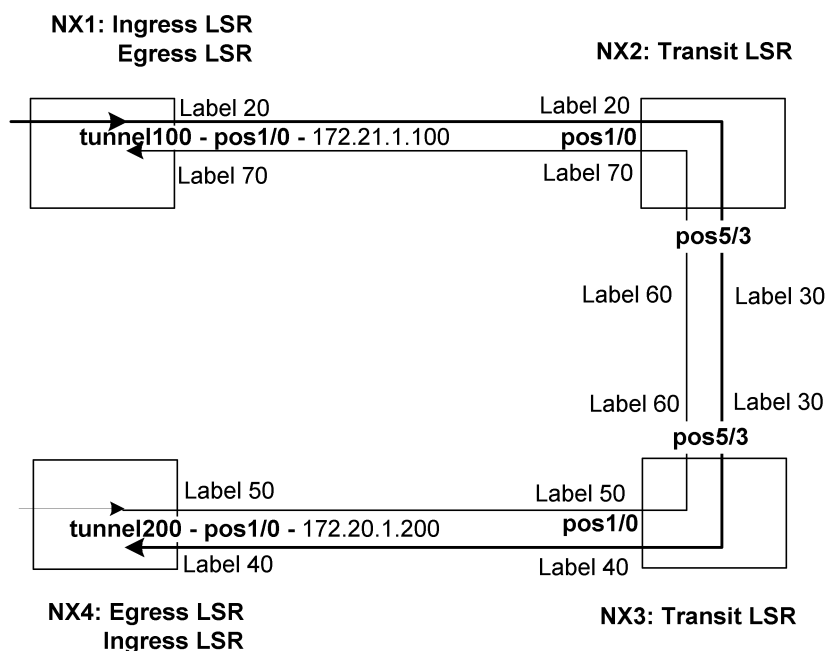
**Table 14-1. MPLS Command Usage**

Command	Static LSP	Signaled LSP	Integration with IGP	Diff-Serv	Verification
tunnel bind	✓	✓			
tunnel destination			✓		
tunnel mode mpls	✓	✓			

## Static Label-switched Path Configuration

The following example shows the configuration of two static LSPs, one from NX1 to NX4 and the other from NX4 to NX1. The NX2 and NX3 systems act as transit LSRs for MPLS cross connects.

Static paths require path configuration at each hop—ingress, egress, and cross connects.



**Figure 14-7. Example of a Static LSP Configuration**

**To configure a static LSP:**

1. Set up the origination of an LSP on the ingress LSR.
2. Set up the termination of an LSP on the egress LSR.
3. Set up LSP integration with an IGP, if appropriate.

4. Configure cross connects on transit LSRs in the MPLS core.
5. Set up support for Diff-Serv service classes, if appropriate.

## Setting up the End Points for an Incoming and Outgoing LSP

In most cases, a system that originates an outgoing LSP also terminates an associated incoming LSP. In this case, the configuration on the ingress LSR to originate an LSP and the configuration on an egress LSR to terminate an LSP follow the same procedure.

### To originate or terminate an LSP:

1. Configure a tunnel interface.
2. Set the mode of the tunnel interface to MPLS.
3. Bind the tunnel interface to a physical interface.
4. Associate a static LSP with the tunnel interface and specify a label.

In this example NX1 and NX4 function as both ingress and egress LSRs, with a single tunnel interface on each system configured to support both an inbound LSP and an outbound LSP.

NX1:

```
nx1# configure terminal
nx1(config)# interface pos1/0
nx1(config-if)# encapsulation ppp
nx1(config-if)# mpls enable
nx1(config-if)# exit
nx1(config)# interface tunnel100
nx1(config-if)# ip address 172.21.1.100 255.255.255.0
nx1(config-if)# tunnel mode mpls
nx1(config-if)# tunnel bind pos1/0
nx1(config-if)# mpls lsp static ingress 20
nx1(config-if)# mpls lsp static egress 70
```

NX4:

```
nx4# configure terminal
nx4(config)# interface pos1/0
nx4(config-if)# encapsulation ppp
nx4(config-if)# mpls enable
nx4(config-if)# exit
nx4(config)# interface tunnel200
nx4(config-if)# ip address 172.20.1.200 255.255.255.0
nx4(config-if)# tunnel mode mpls
nx4(config-if)# tunnel bind pos1/0
nx4(config-if)# mpls lsp static ingress 50
nx4(config-if)# mpls lsp static egress 40
```

The following table explains each configuration entry specific to MPLS on ingress and egress LSRs:

Configuration Line	Description
<code>mpls enable</code>	Enables MPLS on the interface.
On NX1: <code>interface tunnel100</code> On NX4: <code>interface tunnel200</code>	Creates a logical interface of type tunnel.
<code>tunnel mode mpls</code>	Sets the encapsulation mode of the tunnel interface to MPLS.
On NX1: <code>tunnel bind pos1/0</code> On NX4: <code>tunnel bind pos1/0</code>	Binds tunnel interface 100 to physical interface pos1/0 on NX1, and tunnel interface 200 to physical interface pos1/0 on NX4.
On NX1: <code>mpls lsp static ingress 20</code> On NX4: <code>mpls lsp static ingress 50</code>	Configures an inbound LSP, that is the origination of an LSP, within tunnel interface 100 and sets the outgoing label to 20.  Configures an inbound LSP, that is the origination of an LSP, within tunnel interface 200 and sets the outgoing label to 50.
On NX1: <code>mpls lsp static egress 70</code> On NX4: <code>mpls lsp static egress 40</code>	Configures an outbound LSP, that is the termination point for the LSP. On NX1 sets the label value to 70. On NX4 sets the label value to 40.

## Setting Up LSP Integration with an IGP

You can integrate LSPs with either OSPF or IS-IS.

### To integrate an IGP:

1. Set a destination for the LSP.
2. Configure OSPF or IS-IS to use a specified LSP.
3. Set the cost, or metric, of the LSP for the IGP.

## Setting Up OSPF to Use an LSP

The following lines would appear in the configuration section for the tunnel100 interface on NX1 for an *ingress LSP* and tunnel200 interface on NX4 for an *ingress LSP* if the associated LSP is used in OSPF routing:



NX1:

For tunnel100 configuration:

```
nx1(config-if)# ip ospf mpls-tunnel private
nx1(config-if)# tunnel destination 172.20.1.200
nx1(config-if)# ip ospf cost 5
```

NX4:

For tunnel200 configuration:

```
nx4(config-if)# ip ospf mpls-tunnel private
nx4(config-if)# tunnel destination 172.21.1.100
nx4(config-if)# ip ospf cost 5
```

The following table explains each configuration entry for a tunnel interface associated with an LSP used in OSPF routing:

Configuration Line	Description
<code>ip ospf mpls-tunnel private</code>	Configures OSPF to use a specified MPLS tunnel interface, and sets the mode as private for the associated LSP.
On NX1: <code>tunnel destination 172.20.1.200</code> On NX4: <code>tunnel destination 172.21.1.100</code>	Sets the destination for the LSP. The LSP destination for the LSP originating on NX1 is 172.20.1.200. The LSP destination for the LSP originating on NX4 is 172.21.1.100.
<code>ip ospf cost 5</code>	Sets the cost for the LSP to 5.

- Typically, OSPF is configured to use private mode to prevent an interface from receiving protocol messages. If you want to send and receive protocol messages, set the mode to normal for a tunnel interface that has both an inbound and outbound LSP with the same end points.

## Setting Up IS-IS to Use an LSP

The following lines would appear in the configuration section for the tunnel100 interface on NX1 for an *ingress LSR* and tunnel200 interface on NX4 for an *ingress LSR* if the associated LSP is used in IS-IS routing:

NX1:

For tunnel100 configuration:

```
nx1(config-if)# ip router isis private
nx1(config-if)# isis metric 5
nx1(config-if)# isis circuit-type level1-2
nx1(config-if)# tunnel destination 172.20.1.200
```

NX4:

For tunnel200 configuration:

```
nx4(config-if)# ip router isis private
nx4(config-if)# isis metric 5
nx4(config-if)# isis circuit-type level1-2
nx4(config-if)# tunnel destination 172.21.1.100
```

The following table explains each configuration entry for a tunnel interface associated with an LSP used in IS-IS routing:

Configuration Line	Description
<code>ip router isis private</code>	Configures IS-IS to use a specified MPLS tunnel interface, and sets the mode to private for the associated LSP.
<code>isis metric 5</code>	Sets the cost for the LSP to 5.
<code>isis circuit-type level1-2</code>	Sets the IS-IS circuit type to level 1-2. This is part of a typical IS-IS configuration on any interface that does IS-IS routing.
On NX1: <code>tunnel destination 172.20.1.200</code> On NX4: <code>tunnel destination 172.21.1.100</code>	Sets the destination for the LSP. The LSP destination for the LSP originating on NX1 is 172.20.1.200. The LSP destination for the LSP originating on NX4 is 172.21.1.100

- Typically, IS-IS is configured to use private mode to prevent an interface from receiving protocol messages.

## Configuring Cross Connects

Each transit router requires a cross connect be configured between the incoming interface and outgoing interface.

**To configure cross connects on transit LSRs in the MPLS core:**

1. Enable MPLS on physical interfaces that forward traffic for MPLS.
2. Configure a cross connect and specify labels.

In this example, NX2 and NX3 function as transit routers where cross connections link an incoming LSP segment with an outgoing LSP segment. Each segment is associated with an MPLS-enabled physical interface that has assigned label values.

NX2:

```
nx2# configure terminal
nx2(config)# interface pos1/0
nx2(config-if)# encapsulation ppp
nx2(config-if)# mpls enable
nx2(config-if)# exit
nx2(config)# interface pos5/3
nx2(config-if)# encapsulation ppp
nx2(config-if)# mpls enable
nx2(config-if)# exit
nx2(config)# mpls cross-connect pos1/0 20 pos5/3 30
```

NX3:

```
nx3# configure terminal
nx3(config)# interface pos5/3
nx3(config-if)# encapsulation ppp
nx3(config-if)# mpls enable
nx3(config-if)# exit
nx3(config)# interface pos1/0
nx3(config-if)# encapsulation ppp
nx3(config-if)# mpls enable
nx3(config-if)# exit
nx3(config)# mpls cross-connect pos5/3 30 pos1/0 40
```

The following table explains configuration entry specific to MPLS on transit LSRs:

Configuration Line	Description
<b>mpls enable</b>	Enables MPLS on the interface.
On NX3: <b>mpls cross-connect pos1/0 20 pos5/3 30</b> On NX4: <b>mpls cross-connect pos5/3 30 pos1/0 40</b>	Sets up an MPLS cross connect.  On NX3, the incoming interface is pos1/0 and the outgoing interface is pos5/3. The label value swaps from 20 to 30.  On NX4, the incoming interface is pos5/3 and the outgoing interface is pos1/0. The label value swaps from 30 to 40.

## Setting up Service Class Support for Static LSPs

The NX64000 IP Core Router provides Diff-Serv support on transit LSRs. Configuring Diff-Serv support requires enabling per-hop behaviors on the system, and associating a Diff-Serv per-hop behavior with an LSP. The ingress LSR and the egress LSR should support the same per-hop behaviors to form a Diff-Serv MPLS domain.

**To set up a service class for an LSP:**

1. Enable a per-hop group or one or more per-hop behaviors on the system
2. Create a map class and assign one or more per-hop behaviors or a per-hop group to the class.
3. Assign the map class to a cross connect.

The following examples show basic Diff-Serv for MPLS configuration on the system, and illustrate how to set up an LSP to use Diff-Serv.

### Creating Diff-Serv Map Classes

The following configuration enables a per-hop behavior group on the system, and sets up a map class to use this per-hop behavior group. Each NX64000 transit router through which the LSP passes requires the same configuration. The example shows the configuration for NX2, but the same configuration would also appear on NX3.

```
nx2# configure terminal
nx2(config)# diffserv enable group spphb
nx2(config)# map-class diffserv multi-class class1
nx2(config-diffserv-mapclass)# phb spphb 1 exp 0
nx2(config-diffserv-mapclass)# phb spphb 2 exp 1
```

The following table explains configuration entry specific to setting up a service class for MPLS:

Configuration Line	Description
<code>diffserv enable group spphb</code>	Enables the strict priority per-hop behavior group on the system.
<code>map-class diffserv multi-class class1</code>	Creates a Diff-Serv map class, named 1, that supports multiple classes of service.
<code>phb spphb 1 exp 0</code> <code>phb spphb 2 exp 1</code>	Adds the per-hop behaviors 1 and 2 from the strict priority per-hop behavior group to map class 1. These per-hop behaviors are applied to labeled packets.

### Configuring Diff-Serv for Cross Connects

Setting up a Diff-Serv MPLS domain requires you assign a configured map class on each NX64000 transit router. The following examples show the `mapclass` extension added to the `mpls cross-connects` command. In the example, this extension assigns class1 to the cross connect.

NX2:

```
nx2# configure terminal
nx2(config)# mpls cross-connect pos1/0 20 pos5/3 30 mapclass class1
```

NX3:

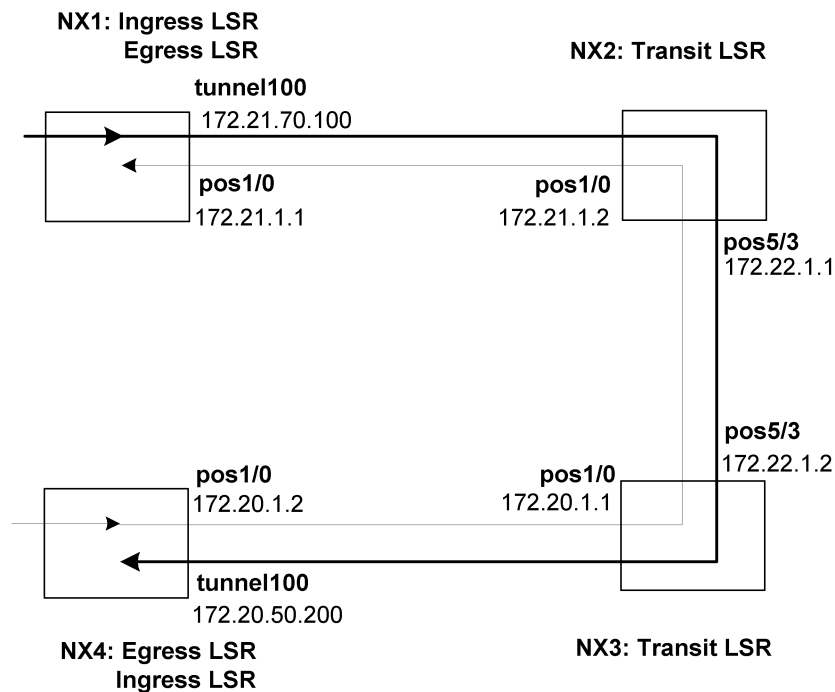
```
nx3# configure terminal
nx3(config)# mpls cross-connect pos5/3 30 pos1/0 40 mapclass class1
```

For more information about configuring cross connects, see [“Configuring Cross Connects” on page 14-18](#).

## RSVP-signaled Label-switched Path Configuration

The following example shows the configuration of an RSVP-signaled LSP, from NX1 to NX4, and another from NX4 to NX1. The NX2 and NX3 systems act as transit LSRs for MPLS cross connects. RSVP-TE sets up these cross connects and their labels.

PPP establishes the connection between systems *on the same subnet*.



**Figure 14-8. Example RSVP-signaled LSP Configuration**

Most of the configuration for signaled LSPs is done on the ingress LSR. Only RSVP and MPLS must be enabled on the transit and egress LSRs. Typically, an edge LSR both originates and terminates an LSP with the same end points.

### To set up a signaled LSP:

1. Enable MPLS and RSVP on transit LSRs.
2. Enable MPLS and RSVP on the egress LSR.
3. Configure the signaled LSP from an ingress LSR.

► Physical interfaces that are part of a signaled LSP must be assigned IP addresses. LSPs cannot be set up through IP unnumbered interfaces.

Tunnel interfaces must also be assigned IP addresses. This way the system can route traffic through a tunnel interface to an LSP.

## Setting up Transit LSRs

Interfaces on transit LSRs must have MPLS and RSVP enabled. The path configuration on the ingress LSR specifies the IP address of each hop. RSVP-TE sets up the cross connect and the labels for the cross connect.

### To configure a signaled transit LSR:

1. Specify an IP address (if not already set) on the incoming interface for the LSP.
2. Enable MPLS on the incoming and outgoing interfaces.
3. Enable RSVP on the incoming and outgoing interfaces.
4. Set up Diff-Serv service class support for the LSP, if appropriate. (For an example, see [“Setting up Service Class Support for Signaled LSPs” on page 14-26.](#))

In this example, NX2 and NX3 function as transit routers where cross connections link an incoming LSP segment with an outgoing LSP segment.

NX2:

```
nx2# configure terminal
nx2(config)# interface pos1/0
nx2(config-if)# encapsulation ppp
nx2(config-if)# ip address 172.21.1.2 255.255.255.0
nx2(config-if)# mpls enable
nx2(config-if)# rsvp enable
nx2(config-if)# exit
nx2(config)# interface pos5/3
nx2(config-if)# encapsulation ppp
nx2(config-if)# ip address 172.22.1.1 255.255.255.0
nx2(config-if)# mpls enable
nx2(config-if)# rsvp enable
```

NX3:

```
nx3# configure terminal
nx3(config)# interface pos5/3
nx3(config-if)# encapsulation ppp
nx3(config-if)# ip address 172.22.1.2 255.255.255.0
nx3(config-if)# mpls enable
nx3(config-if)# rsvp enable
nx3(config-if)# exit
nx3(config)# interface pos1/0
nx3(config-if)# encapsulation ppp
nx3(config-if)# ip address 172.20.1.1 255.255.255.0
nx3(config-if)# mpls enable
nx3(config-if)# rsvp enable
```

The following table lists the configuration entries to enable MPLS and RSVP on transit LSRs:

Configuration Line	Description
<b>mpls enable</b>	Enables MPLS on the interface.
<b>rsvp enable</b>	Enables RSVP-TE on the interface.

## Setting up Signaled Paths at the Ingress LSR

The configuration on the ingress LSR defines each hop in the explicit path from the ingress LSR to the egress LSR.

### To establish a signaled LSP:

1. Configure an explicit path from an ingress LSR to an egress LSR.
2. Enable MPLS and RSVP on physical interfaces that have a connection to the next hop.
3. Configure a tunnel interface.
4. Set the mode of the tunnel interface to MPLS.
5. Bind the tunnel interface to a physical interface.
6. Specify the explicit path the LSP should use, and set a destination for the LSP.

In this example, NX1 provides the configuration for the LSP to NX4, and NX4 provides the configuration for the LSP to NX1. The explicit path configuration includes both the physical interface address and the address of the tunnel interface to which it binds on the egress LSR.

- The identifier for the tunnel interface, 100, is the same on systems at both ends of the LSPs to make the configuration easier to maintain. With large numbers of tunnel interfaces, it is easier to locate the tunnel interfaces for LSP end points if these interfaces have the same number.
- On an egress LSR, the IP address for a tunnel interface should be on a different subnet if the subnet is masked (for example, mask 255.255.255.0).

NX1:

Ingress LSR for signaled path to 172.20.50.200:

```
nx1# configure terminal
nx1(config)# ip explicit-path 50
nx1(config-ip-path)# next-address 172.21.1.2
hop_index=1
nx1(config-ip-path)# next-address 172.22.1.2
hop_index=2
nx1(config-ip-path)# next-address 172.20.1.2
hop_index=3
nx1(config-ip-path)# next-address 172.20.50.200
hop_index=4
nx1(config-ip-path)# exit
nx1(config)# interface pos1/0
nx1(config-if)# encapsulation ppp
nx1(config-if)# ip address 172.21.1.1 255.255.255.0
nx1(config-if)# mpls enable
nx1(config-if)# rsvp enable
nx1(config-if)# exit
nx1(config)# interface tunnel100
nx1(config-if)# ip address 172.21.70.100
nx1(config-if)# tunnel mode mpls
```

```
nx1(config-if)# tunnel bind pos1/0
nx1(config-if)# mpls lsp signal lspid 10 50 172.20.50.200
```

NX4:

Ingress LSR for signaled path to 172.21.70.100:

```
nx4# configure terminal
nx4(config)# ip explicit-path 70
nx4(config-ip-path)# next-address 172.20.1.1
hop_index=1
nx4(config-ip-path)# next-address 172.22.1.1
hop_index=2
nx4(config-ip-path)# next-address 172.21.1.1
hop_index=3
nx4(config-ip-path)# next-address 172.21.70.100
hop_index=4
nx4(config-ip-path)# exit
nx4(config)# interface pos1/0
nx4(config-if)# encapsulation ppp
nx4(config-if)# ip address 172.20.1.2 200 255.255.255.0
nx4(config-if)# mpls enable
nx4(config-if)# rsvp enable
nx4(config-if)# exit
nx4(config)# interface tunnel100
nx4(config-if)# ip address 172.20.50.200
nx4(config-if)# tunnel mode mpls
nx4(config-if)# tunnel bind pos1/0
nx4(config-if)# mpls lsp signal lspid 90 70 172.21.70.100
```

The following table explains configuration entries specific to MPLS on an ingress LSR:

Configuration Line	Description
On NX1: <b>ip explicit-path 50</b> On NX4: <b>ip explicit-path 70</b>	Creates explicit path 50 (the defined path, that RSVP-TE uses when it sets up a signaled path on NX1, and explicit path 70 on NX4.  To support a bidirectional LSP, the explicit path specifies the address of the tunnel interface on the egress LSR.
On NX1 and NX4: <b>next-address ip-address</b>	Sets the next address in the sequence of LSRs that define an explicit path.
On NX1 and NX4: <b>mpls enable</b>	Enables MPLS on the interface.
On NX1 and NX4: <b>rsvp enable</b>	Enables RSVP-TE on the interface.



Configuration Line	Description
On NX1: <code>interface tunnel100</code> On NX4: <code>interface tunnel100</code>	Creates a logical interface of type tunnel.
On NX1 and NX4: <code>tunnel mode mpls</code>	Sets the tunnel mode of a physical interface to MPLS.
On NX1 and NX4: <code>tunnel bind pos1/0</code>	Binds tunnel interface 100 to physical interface pos1/0. <b>Note:</b> In this example, both systems use the pos1/0 interface. Any interface could be used.
On NX1: <code>mpls lsp signal lspid 10 50</code> <code>172.20.50.200</code> On NX4: <code>mpls lsp signal lspid 90 70</code> <code>172.21.1.100</code>	On NX1, specifies for LSP 10, explicit path 50, and destination IP address 172.20.50.200. On NX4, specifies for LSP 90, explicit path 70, and destination IP address 172.21.70.100.  Because both LSPs have the same end points, the destination IP address is that of the tunnel interface.

- If you create an LSP that transmits traffic in one direction, but does not return traffic in the other direction the LSP can terminate in either a physical interface or a tunnel interface.

## Verifying Path Configuration

You can view the addresses in a defined explicit path by running the `list` command on an ingress LSR. The following example shows the explicit path address list configured on NX1:

```
nx1# configure terminal
nx1(config)# ip explicit-path 50
nx1(config-ip-path)# list
1: next-address 172.21.1.2
2: next-address 172.22.1.2
3: next-address 172.20.1.2
4: next-address 172.20.1.200
```

If you want to compare explicit paths configured on the system, specify a path identifier as an argument for the command to view the addresses in any configured explicit path. For example to view the addresses in explicit path 100, enter:

```
nx(config-ip-path)# list 100
```

## Changing the Configuration of an Explicit Path

Use the **replace-entry** command to change an address previously configured in an explicit path. You enter this command at the prompt for a specified explicit path. The following example shows how to change the second address in explicit path 50:

```
nx1# configure terminal
nx1(config)# ip explicit-path 50
nx1(config-ip-path)# replace-entry 2 172.22.1.50
nx1(config-ip-path)# list
1: next-address 172.21.1.2
2: next-address 172.22.1.50
3: next-address 172.20.1.2
4: next-address 172.20.1.200
```

## Setting up an IGP to Use the LSP

You set up IS-IS or OSPF to use a signaled LSP in the same way you configure these protocols to forward traffic over a static LSP. For information about setting up OSPF to use the LSP, see [“Setting Up OSPF to Use an LSP” on page 14-16](#). For information about setting up IS-IS to use the LSP, see [“Setting Up IS-IS to Use an LSP” on page 14-17](#).

## Setting up Service Class Support for Signaled LSPs

The NX64000 switch/router provides Diff-Serv support for signaled LSPs on transit LSRs.

### To configure Diff-Serv support on a transit LSR:

1. Enable a per-hop behavior group or a per-hop behavior on the system.
2. Set the Diff-Serv class for interoperability with other systems traversed by the LSP.
3. Set the Diff-Serv error-code for interoperability with other systems traversed by the LSP.

All of the values you configure must be the same for each system in the path.

The following example sets up Diff-Serv support on NX2. The configuration on NX3 would be same because the sample configuration uses the same interfaces on the two systems.

```
nx2# configure terminal
nx2(config)# diffserv enable group spphb
nx2(config)# interface pos5/3
nx2(config-if)# rsvp diffserv-class 70
nx2(config-if)# rsvp diffserv-error 71
nx2(config-if)# exit
nx2(config)# interface pos1/0
nx2(config-if)# rsvp diffserv-class 70
nx2(config-if)# rsvp diffserv-error 71
```

The following table lists the configuration entries to enable MPLS and RSVP on transit LSRs:

Configuration Line	Description
<b>diffserv enable group spphb</b>	Enables the strict priority per-hop behavior group.

Configuration Line	Description
<code>rsvp diffserv-class 70</code>	Sets the Diff-Serv class to 70.
<code>rsvp diffserv-error 71</code>	Sets the Diff-Serv error code to 71.

## MPLS Commands to Verify Configuration

The NX-IS software provides a number of show commands that let you verify configuration and assess traffic statistics for MPLS-enabled and RSVP-enabled interfaces. The following tables lists the type of information you can view from each show command:

**Table 14-2. MPLS Commands to Verify Configuration**

Action	On this LSR type	For this LSP type	Command
Display a list of the routers in an RSVP explicit path.	Ingress	Signaled	<code>show ip explicit-path</code>
Display information about the interfaces configured to form a cross connect for an LSP.	Transit	Signaled, static	<code>show mpls cross-connects</code>
List the status or statistics of interfaces that provide incoming segments.	Egress, transit	Signaled, static	<code>show mpls in-segments</code>
Display information for all interfaces on the system that have MPLS enabled.	Ingress, egress, transit	Signaled, static	<code>show mpls interfaces</code>
Display information about each LSP that originates or terminates on this router.	Ingress, egress	Signaled, static	<code>show mpls lsps</code>
List the status or statistics of interfaces that provide outgoing segments.	Ingress, transit	Signaled, static	<code>show mpls out-segments</code>
Display the list of interfaces that have RSVP enabled.	Ingress, egress, transit	Signaled	<code>show rsvp interfaces</code>
Display information about the adjacent routers in a signaled LSP.	Ingress, egress, transit	Signaled	<code>show rsvp neighbors</code>
Display the status of signaled LSPs being set up by RSVP.	Ingress, egress, transit	Signaled	<code>show rsvp session</code>
Display information about all the members included in one or all per-hop behavior groups.	Ingress, egress, transit	Signaled, static	<code>show diffserv group</code>

**Table 14-2. MPLS Commands to Verify Configuration**

Action	On this LSR type	For this LSP type	Command
Display configuration information about a specified Diff-Serv map class, or all Diff-Serv map classes.	Ingress, egress, transit	Signaled, static	<code>show diffserv mapclass</code>
Display information about a specified member of a per-hop behavior group.	Ingress, egress, transit	Signaled, static	<code>show diffserv phb</code>

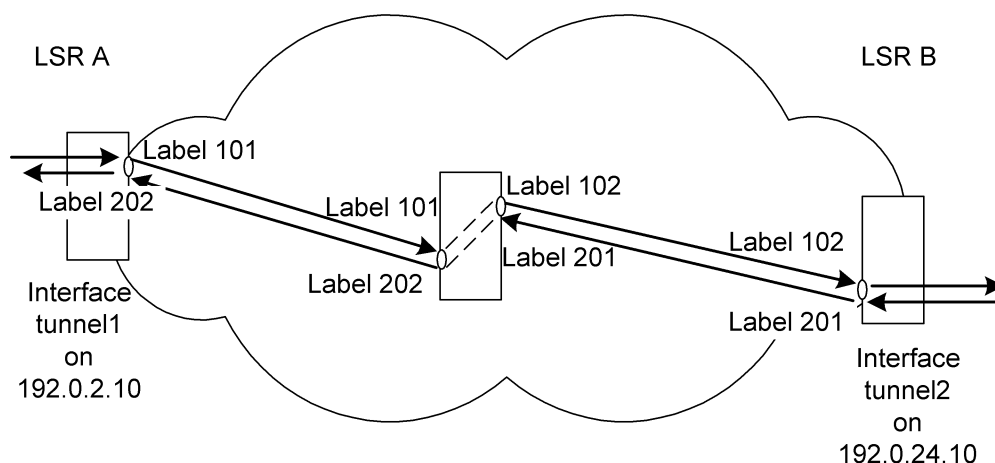
## Ping and Traceroute over MPLS Paths

The one directional nature of MPLS LSPs presents special issues when using the `ping` or `traceroute` utilities to verify network connectivity.

### Ping

Ping sends out ICMP echo requests in one direction, and receives ICMP echo responses in the opposite (return) direction. Ping requests can succeed only with LSPs configured in both directions, for example from A to B, and the return from B to A.

The following illustration shows a configuration where LSR A successfully pings 192.0.24.0, the IP address on LSR B:



**Figure 14-9. Ping over 2 LSPs**

## Traceroute

Traceroute has the same configuration requirements as ping for the command to succeed over LSPs. The configuration shown in [Figure 14-9](#) also supports successfully running traceroute over LSPs. When interpreting command output, note that the Time-To-Live decrements at each hop, including the hops within an LSP.

## Interoperability Issues

If you use a Juniper system within an LSP, you should be aware of the implementation differences between the two types of systems. The following table lists these differences and makes configuration suggestions to let the systems forward traffic along an LSP:

Feature	Implementation Difference	Do this on the Juniper system	Do this on the NX64000 system
RSVP signaling	Juniper systems implement a constrained SPF protocol for signaled LSPs. The NX systems provide an implicit implementation for RSVP-TE.	For the Juniper system to initiate an RSVP path, add the command to specify “no constrained shortest path first.” <b>Note:</b> Do not make this change for static routes.	n/a
LSP termination	Juniper systems terminate LSPs only when the label is zero. The NX systems supports termination of LSPs over arbitrary labels.	n/a	In a configuration that has a Juniper system at the edge, and the NX system is the penultimate router (next to last) for a static LSP, set the outgoing label to three (3).



# Internet Protocol (IP) Multicast Configuration

This chapter describes the basic behavior of and required configuration for IP multicast protocols running on the NX64000. IP Multicast uses the following protocols:

- Protocol Independent Multicast (PIM), Sparse Mode, Version 2
- Internet Group Management Protocol (IGMP), Version 1 and 2
- Multicast Source Discovery Protocol (MSDP)

IP multicasting is the transmission of an IP datagram to a host group (a set of hosts identified by a single, class D, IP destination address). The membership of a host or multicast group is dynamic. A host, anywhere in the IP multicast routing domain, can join or leave a group at any time.

Multicasting is particularly suited for applications such as webcasting or video conferencing, which can take advantage of packet replication by multicast routers. Transmission of one data stream to a multicast group uses less bandwidth than transmission of multiple unicast streams to multiple hosts.

## Key Features

The implementation in NX-IS version 1.7 supports:

- PIM Version 2 only. There is no interoperability with routers implementing PIM Version 1.
- PIM—Sparse Mode (PIM-SM) only
- Neighbor discovery on LANs via Hello messages
- Interface based enabling
- Information tracking on hosts leaving/joining multicast groups
- Host queries when the router is elected as the Designated Router (DR) for a LAN
- Tracking changes to the unicast routing table to update information on incoming and outgoing interfaces

## Technology Concepts

Basic to understanding of IP multicast are:

- PIM and PIM-SM
- IGMP
- MSDP
- Designated routers (DRs)
- Bootstrap routers (BSRs)
- Rendezvous points (RPs)
- Access lists

## RFCs and Standards

The NX-IS implementation of PIM is based in part on the following Request For Comments (RFCs) and standards:

RFC	Title
RFC 2117	Protocol Independent Multicast-Sparse Mode (PIM-SM)
RFC 2362	Protocol Independent Multicast Sparse Mode (PIM-SM) and recent IETF PIM WG drafts
RFC 2236	Internet Group Management Protocol, Version 2
RFC 2933	Internet Group Management Protocol MIB

## Protocol Independent Multicast (PIM)

PIM is a multicast routing protocol with control mechanisms for building multicast data distribution trees and for forwarding multicast data down these trees. PIM is not dependent on the type of unicast routing protocol in use in the network, however it uses information from the unicast routing table, so a correctly configured unicast routing is required for PIM to function. PIM routers discover neighboring PIM routers via the exchange of Hello messages.

PIM relies on an underlying topology-gathering protocol to populate a routing table with routes. This routing table is called the multicast routing information base (MRIB). The routes in this table are taken directly from the unicast routing table. The routes in the MRIB represent a multicast-capable path to each subnet. The MRIB is used to determine the path that PIM control messages, such as Join messages, take to get to the source subnet, and data flows along the reverse path of the Join messages.

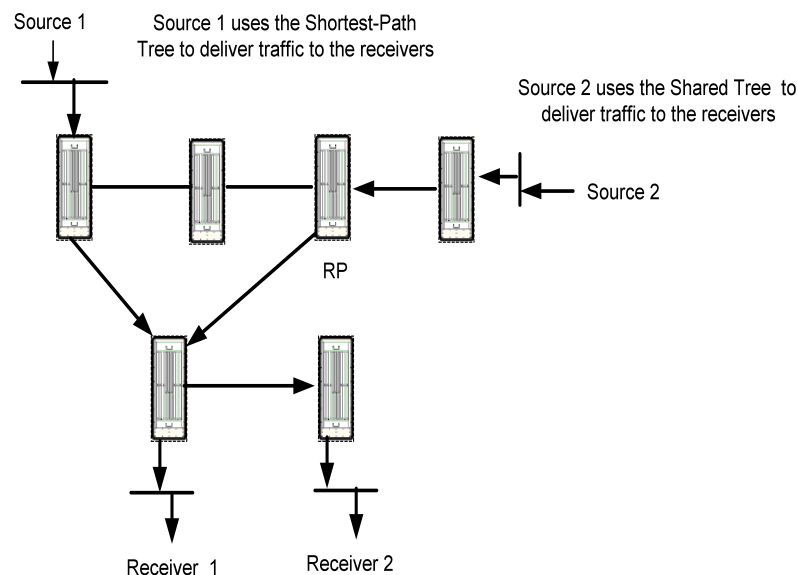


## PIM Sparse Mode (PIM-SM)

The system uses PIM-SM for building multicast data distribution trees in an environment where group members are widely distributed, and can be found only on some subnetworks. When operating in sparse mode, interfaces and routes are added to the IP multicast routing table only when join messages are received from routers or when a member is directly connected to the interface.

► Unless the data is received on the rendezvous point (RP) interface it is discarded.

The tree creation is initiated by a receiver joining a group. When a group is joined at the last-hop designated router (DR), it sends a join towards the RP and a branch is created from the DR to RP. The data can now be forwarded down the shared tree from the RP to the last-hop DR.



**Figure 15-1. PIM Sparse Mode with Shared Tree and Shortest-Path Tree**

A separate branch is built from the RP to the first-hop DR. Initially, datagrams from a multicast source are encapsulated by the DR into PIM register messages. The PIM register messages are unicast to the group RP. Upon reception of the first PIM register message, the RP sends a join towards the source of the multicast datagrams, so further datagrams can be forwarded without encapsulation towards the RP.

An additional shortest-path tree branch can be built from the last-hop DR when it receives the first packet of multicast data. If configured, the DR sends a join towards the sources and builds a *shortcut* source-specific branch so data can be forwarded directly from the *first-hop* DR to the *last-hop* DR.

## Internet Group Management Protocol (IGMP)

IP hosts use IGMP to report multicast group memberships to neighboring multicast routers. IGMP enables the router to dynamically establish group membership. There is no limit to the number of members that can join a group. IP hosts send IGMP group membership Report and Leave Group messages to neighboring multicast routers to indicate if they want to receive multicast data for a particular class D IP group address.

Multicast routers listen for IGMP messages from hosts to keep track of which groups have members on each network attached to the router. The IGMP protocol informs the multicast routing protocol (PIM) when members of a group are present. In turn, the routing protocol sets up appropriate routing mechanisms to deliver multicast data for a particular group.

The IGMP protocol is automatically enabled on an interface when the interface is configured for PIM routing.

## Multicast Source Discovery Protocol (MSDP)

MSDP enables connections and distributes knowledge of multicast sources between multiple PIM-SM domains. Each PIM-SM domain uses its own independent rendezvous points (RPs) and does not have to depend on RPs in other domains.

MSDP fits in between PIM and Border Gateway Protocol (BGP) and is closely related to BGP. MSDP accesses BGP routing table structures to make decisions about accepting received Source-Active (SA) messages. (Information about active sources in a domain is shared between MSDP peers in SA messages.) This information is stored in a cache if the receiving peer is configured as a caching peer. Non-caching peers can send out SA-Request messages to caching peers to request information about active sources when a receiver joins a new multicast group in their domain.

MSDP enabled routers peer with other MSDP enabled routers in other domains. This is done using a TCP connection between the two nodes (in a similar way to BGP). For more information on BGP, see [Chapter 20, “Border Gateway Protocol \(BGP\) Commands”](#).

You must configure an MSDP peer to use it as the default peer.

- If you specify a standard access list, the peer only accepts Source Active (SA) messages from RPs that are permitted by the specified list.
- If the system is configured with only one default peer, then all SA messages are accepted from this peer.
- If multiple default peers are configured on the system, then the peer's priority is used to decide which peer is the active default peer (from which, all SA messages are accepted.)
- If two default peers have the same priority then the one with the lowest numerical IP address becomes the active default peer.
- If a default peer has an access list configured then the system always accepts SA messages that pass this list regardless of the priority of the peer.

## Designated Routers (DRs)

You designate a BSR to provide a mechanism for the router to discover and distribute RPs and to dynamically learn the group-to-RP mappings. On a LAN, a designated router (DR) is elected as part of the Hello process. In sparse mode domains, it is the job of the DR, nearest the source, to encapsulate multicast traffic in unicast register packets and forward them towards the relevant RP. It is also the job of the DR to send join messages towards the RP when a new receiver appears on its LAN, as discovered via IGMP.

## Bootstrap Routers (BSRs)

A bootstrap router (BSR) originates bootstrap messages that carry out dynamic BSR election when needed and distribute RP information. The bootstrap message indicates the state of the RPs. If an RP is included in the bootstrap message, then the RP is tagged as being up and available. RPs not included in the message are removed from the list of RPs.

To obtain the RP information, all routers within a PIM domain receive, collect, and store bootstrap messages originated by the BSR. Each router continues to use the contents of the most recently received bootstrap message until it receives a new bootstrap message.

## Rendezvous Points (RPs)

A rendezvous point (RP) is a router configured to act as the root of the non-source-specific distribution tree for a multicast group. Join messages from receivers for a group are sent to the RP, and data from senders is sent to the RP so that receivers can discover who the senders are, and start to receive traffic for the group.

In PIM-SM mode, an RP is used as the root of forwarding from multiple sources via a shared-tree to receivers. Within the administered PIM-SM domain you can configure multiple routers as RP candidates serving a set of multicast groups. For each SM group there needs to be at least one candidate RP available.

## Access Lists

You can use an access list to specify which group addresses and prefixes use the system as a RP candidate. The system advertises itself as an RP candidate for all permissions the Class-D entries in the list specified by the `group-list` option in the command. By default, (if no `group-list` option is specified) the system advertises itself as an RP candidate for the group prefix. This means it becomes an RP candidate for all groups. For more information on access list usage, see the `access-list` command in the *NX64000 Command Reference* or [Chapter 12, “Access List Configuration”](#).

## Packet Format

Each data packet is preceded by a fixed size header. The header contains:

**Table 15-1. PIM Header Field Descriptions**

Field	Description
PIM Ver	The version of PIM.
Type	Types for specific PIM messages. These are reserved, set to zero and ignored upon receipt.
Reserved	Transmits as zero and is ignored upon receipt.
Checksum	<p>The checksum is standard IP checksum, i.e. a 16-bit complement of the sum of the entire PIM message, excluding the data portion in the Register message. For computing the checksum, the checksum field is zeroed.</p> <p><b>Note:</b> The checksum for Registers is done only on first 8 bytes of a packet, including the PIM header and the next 4 bytes, excluding the data packet portion. For interoperability reasons, a message carrying checksum done over the entire PIM register message is accepted.</p>

## Message Types

As defined in RFC2117 *Protocol Independent Multicast-Sparse Mode (PIM-SM)*, there are nine defined message types. The following list describes the message types:

- Hello Messages—are sent periodically by routers on all interfaces.
- Register Messages—are sent by the designated router (DR) to the rendezvous point (RP) when a multicast packet is ready to transmit.
- Register-Stop Messages—are sent from the RP to the sender of the Register message.
- Join/Prune Messages—are sent by routers towards sources and RPs. The Join messages are used to establish information to build trees (RP (shared) trees or source trees). Prune messages are sent to discover members that have left groups or sources that no longer use a tree and removes (prunes) them.
- Bootstrap Messages—are sent (via multicast) to all PIM routers within a group, and sent out all interfaces that have PIM neighbors (excluding the one over which the message was received). Bootstrap messages originate at the bootstrap router (BSR).
- Assert Messages—are sent when a multicast data packet is received on an outgoing interface that is associated with a group belonging to a source.
- Candidate-RP-Advertisements—are sent from the candidate rendezvous points (RPs) to the bootstrap router (BSR).

# Internet Protocol Multicast Configuration

This section provides examples of basic Protocol Independent Multicast (PIM) configuration tasks. **Table 15-2** lists the commands that comprise the current implementation and their application in configuring and/or maintaining a PIM network.

- All commands referenced in this chapter are described in the *NX64000 Command Reference* manual.

**Table 15-2. Internet Protocol Multicast Commands Usage**

Commands	Routing	PIM Configuration	IGMP Configuration	MSDP Configuration	Verification
clear ip mroute	✓				
clear ip msdp local-sources	✓				
clear ip msdp peer	✓				
clear ip msdp mcache	✓				
clear ip msdp statistics	✓				
ip igmp last-memb-query-intvl		✓	✓		
ip igmp query-interval		✓	✓		
ip igmp query-max-response-time		✓	✓		
ip igmp robustness		✓	✓		
ip igmp version		✓	✓		
ip msdp cache-sa-state		✓		✓	
ip msdp default-peer	✓	✓		✓	
ip msdp description		✓		✓	
ip msdp filter-sa-request	✓	✓		✓	
ip msdp mesh-group		✓		✓	
ip msdp peer	✓	✓		✓	

**Table 15-2. Internet Protocol Multicast Commands Usage (Continued)**

Commands	Routing	PIM Configuration	IGMP Configuration	MSDP Configuration	Verification
ip msdp redistribute	✓	✓		✓	
ip msdp sa-filter in	✓	✓		✓	
ip msdp sa-filter out	✓	✓		✓	
ip msdp shutdown	✓	✓		✓	
ip msdp ttl-threshold		✓		✓	
ip multicast-routing	✓	✓	✓	✓	
ip pim	✓	✓			
ip pim bsr-border	✓	✓			
ip pim bsr-candidate	✓	✓			
ip pim dr-priority		✓			
ip pim message-interval		✓			
ip pim query-interval		✓			
ip pim reg-checksum-cisco	✓	✓			
ip pim rp-candidate	✓	✓			
ip pim spt-threshold	✓	✓			
ip pim state-ref-int	✓	✓			
show ip igmp interface					✓
show ip igmp groups					✓
show ip mcache					✓
show ip mroute					✓
show ip msdp count					✓

**Table 15-2. Internet Protocol Multicast Commands Usage (Continued)**

Commands	Routing	PIM Configuration	IGMP Configuration	MSDP Configuration	Verification
show ip msdp local-sources					✓
show ip msdp peer					✓
show ip msdp sa-cache					✓
show ip msdp summary					✓
show ip pim bsr-router					✓
show ip pim interface					✓
show ip pim neighbor					✓
show ip pim rp					✓
show ip pim rp-hash					✓
show ip pim rp-set					✓
show ip pim spt-threshold					✓
show ip pim summary					✓

## Basic PIM Configuration Tasks

The following sections illustrate the settings used to configure PIM. Refer to the first part of this chapter for information on the features.

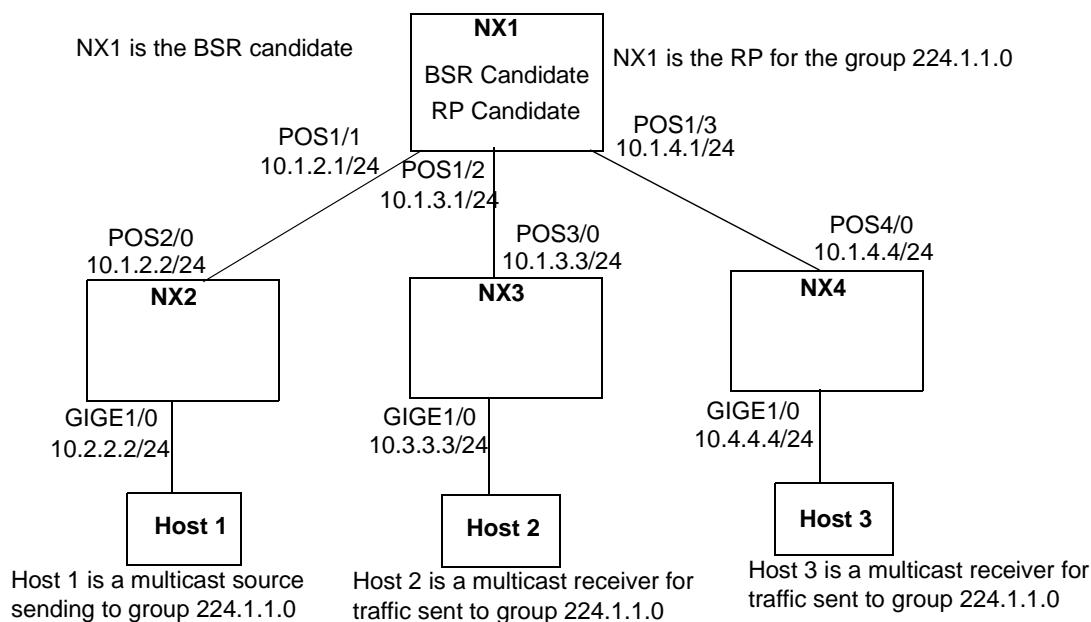
**To configure PIM, you must do the following on all routers that are to run PIM:**

1. Load the PIM module.
2. Enable IP Multicast Routing.
3. Configure global PIM parameters.
4. Enable PIM-SM on the required interfaces.

You can then configure optional PIM interface parameters if desired.

- After you configure PIM, if you unload the PIM module after configuring PIM, all configuration information is lost from the running configuration file. If you saved the information to the startup configuration file, the information is still contained there.

The following figure illustrates the PIM network as configured in this guide:



**Figure 15-2. A Simple PIM Network**

## Enabling PIM

After configuring interfaces (see [Chapter 7, “Cards and Interfaces”](#)) load and enable PIM on all involved routers. In the following configuration, you load PIM on each router, enable sparse mode on all interfaces, and enable multicast routing on all routers. NX1 is also designated as both the BSR candidate and RP candidate.

- You must enable IP multicast routing on all routers in the network and sparse mode on all interfaces within the PIM enabled network.

NX1:

```
nx1# configure terminal
nx1(config)# load pim
nx1(config)# ip multicast-routing
nx1(config)# interface pos1/1 point-to-point
nx1(config-if)# encapsulation ppp
nx1(config-if)# ip address 10.1.2.1 255.255.255.0
nx1(config-if)# ip pim sparse-mode
nx1(config-if)# exit
nx1(config)# interface pos1/2 point-to-point
```



```
nx1(config-if)# encapsulation ppp
nx1(config-if)# ip address 10.1.3.1 255.255.255.0
nx1(config-if)# ip pim sparse-mode
nx1(config-if)# exit
nx1(config)# interface pos1/3 point-to-point
nx1(config-if)# encapsulation ppp
nx1(config-if)# ip address 10.1.4.1 255.255.255.0
nx1(config-if)# ip pim sparse-mode
nx1(config-if)# exit
nx1(config)# ip pim bsr-candidate pos1/1
nx1(config)# ip pim rp-candidate pos1/1
```

NX2:

```
nx2# configure terminal
nx2(config)# load pim
nx2(config)# ip multicast-routing
nx2(config)# interface pos2/0 point-to-point
nx1(config-if)# encapsulation ppp
nx2(config-if)# ip address 10.1.2.2 255.255.255.0
nx2(config-if)# ip pim sparse-mode
nx2(config-if)# exit
nx2(config)# interface gigabitethernet1/0
nx2(config-if)# ip address 10.2.2.2 255.255.255.0
nx2(config-if)# ip pim sparse-mode
```

NX3:

```
nx3# configure terminal
nx3(config)# load pim
nx3(config)# ip multicast-routing
nx3(config)# interface pos3/0 point-to-point
nx1(config-if)# encapsulation ppp
nx3(config-if)# ip address 10.1.3.3 255.255.255.0
nx3(config-if)# ip pim sparse-mode
nx3(config-if)# exit
nx3(config)# interface gigabitethernet1/0
nx2(config-if)# ip address 10.3.3.3 255.255.255.0
nx3(config-if)# ip pim sparse-mode
```

NX4:

```
nx4# configure terminal
nx4(config)# load pim
nx4(config)# ip multicast-routing
nx4(config)# interface pos4/0
nx1(config-if)# encapsulation ppp
nx4(config-if)# ip address 10.1.4.4 255.255.255.0
nx4(config-if)# ip pim sparse-mode
nx4(config-if)# exit
nx4(config)# interface gigabitethernet1/0
nx2(config-if)# ip address 10.4.4.4 255.255.255.0
nx4(config-if)# ip pim sparse-mode
```

The following table explains the use of commands in the preceding example:

Configuration Line	Description
<code>load pim</code>	Loads the PIM software.
<code>ip multicast-routing</code>	Enables multicast routing on this system.
<code>interface</code>	Designates the interface.
<code>encapsulation ppp</code>	Designates PPP as the encapsulation protocol.
<code>ip address</code>	Specifies the IP address of the interface.
<code>ip pim sparse-mode</code>	Enables PIM sparse mode on this interface.
On NX1 only:	
<code>ip pim bsr-candidate</code>	Designates a BSR candidate.
<code>ip pim rp-candidate</code>	Designates an RP candidate.

## Configuring a Bootstrap Router (BSR) Candidate

You must designate at least one router in the network as a BSR candidate. It is recommended that you designate at least two routers as BSRs for redundancy. The following example designates NX1 as the BSR.

```
nx1# configure terminal
nx1(config)# ip multicast-routing
nx1(config)# ip pim bsr-candidate pos1/3 30 80
```

## Configuring a Bootstrap Router (BSR) Border

Designating a BSR border creates a PIM domain. This border blocks bootstrap messages, but not other PIM messages. Each domain uses a different BSR.

The following example designates interface pos5/0 as a BSR border:

```
nx# configure terminal
nx(config)# interface pos4/0
nx(config-if)# ip pim bsr-border
```

## Configuring a Rendezvous Point (RP) Candidate

You must designate at least one router in the network as an RP candidate. It is recommended that you designate at least two routers as RPs for redundancy. The following example designates NX1 as the RP.

```
nx1# configure terminal
nx1(config)# ip multicast-routing
nx1(config)# ip pim rp-candidate atm0/0.19 group-list 10 60 50
nx1(config)# access-list 10 permit 224.1.1.1
```

The following table explains use of commands in the preceding example:

Configuration Line	Description
<code>ip multicast-routing</code>	Enables multicast routing.
<code>ip pim rp-candidate atm0/0.19</code> <code>group-list 10 60 50</code>	Designates the RP candidate.
<code>access-list 10 permit 224.1.1.1</code>	Assigns access-list 10 (this must match the group-list number in the <code>ip pim rp-candidate</code> command), and is used to specify which group or group prefix's (range of groups) the system is an RP candidate for.

## Basic MSDP Configuration Tasks

Before configuring MSDP, BGP must know about the networks containing the addresses being used as MSDP peers. If the networks are not announced, you must configure MSDP default peering using the `ip msdp default-peer` command. To configure MSDP, perform the following tasks.

- Configure MSDP peers, which includes:
  - default peers
  - mesh groups
- Configure Source-Active (SA) messages

### Configuring MSDP Peers

When you configure an MSDP peer, MSDP compares the router's local IP address with the address of the remote peer. The address used as the local address in this comparison is chosen as follows:

1. If you specify the `connect-source` option with the `ip msdp peer` command, then the system uses the address from the specified interface in the comparison.
2. If you do not specify the `connect-source` option, then MSDP searches for an IP interface with a subnet address that encompasses the destination peer's address. MSDP uses the local address of the specified interface in the comparison.
3. If no IP interface is found then the system uses the router identification as the local address in the comparison.

If the local IP address is lower than the remote peer's address, MSDP puts this peer in the connecting state and tries to make a TCP connection to the remote peer. If the local address is higher, then MSDP puts this peer in the listen state, waiting for the remote end to connect.

If the system does not share a common subnet with the remote peer, then using the router ID as the local address in the comparison could result in both ends of the peer connection going into the same state. In this situation, use the `connect-source` option to ensure one end goes into the connecting state and the other the listening state.

You enable MSDP by configuring an MSDP peer for the local router.

```
nx# configure terminal
nx(config)# ip msdp peer 1.1.19.1 connect-source 1.14.2.1 remote-as 4
nx(config)# ip msdp description 1.1.19.1 labrouter
```

The following table explains the use of the commands in the preceding example:

Configuration Line	Description
<code>ip msdp peer 1.1.19.1 connect-source 1.14.2.1 remote-as 4</code>	Enables MSDP and configures an MSDP peer.  connect-source - uses the specified interface as the source for the connection.  remote-as - designates the autonomous system (AS) number for the peer.
<code>ip msdp description 1.1.19.1 labrouter</code>	Enters a text description for the specified peer.

## Configuring Default MSDP Peers

To configure a default MSDP peer:

```
nx# configure terminal
nx(config)# ip msdp default-peer 1.1.19.1 list 10
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip msdp default-peer 1.1.19.1 list 10</code>	Defines the peer at IP address 1.1.19.1 as the default peer. Assigns access list 10 as the filter. For more information on creating access lists see <a href="#">Chapter 12, "Access List Configuration"</a> .

## Configuring MSDP Mesh Groups

An MSDP mesh group is a group of MSDP peers that have full connectivity with one another. The advantage to mesh groups is that SA messages received by any member of the group are not forwarded to other members.

To create a mesh group, issue the `ip msdp mesh-group` command for each peer you want to include in the group:

```
nx# configure terminal
nx(config)# ip msdp mesh-group 60 1.14.1.2
```

The following table explains use of commands in the preceding example:

Configuration Line	Description
<code>ip msdp mesh-group 60 1.14.1.2</code>	Creates an MSDP mesh group and assigns the specified peer as a member of the group.

## Shutting Down MSDP Peers

You may not want an MSDP peer go active until you have completed the peer's configuration. To do this, you can shut down the peer, configure it, and bring it up at a later time. When you shut down an MSDP session, you do not lose the configuration information for that peer. When in shutdown mode, the TCP connection is terminated and not restarted.

```
nx# configure terminal
nx(config)# ip msdp shutdown 1.14.1.2
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip msdp shutdown 1.14.1.2</code>	Administratively shuts down the peer 1.14.1.2.

## Configuring Source-Active (SA) Messages

There are several optional configuration steps you can perform to customize how SA messages are handled. Controlling multicast source information is done one of two ways:

- Specifying which sources are advertised
- Specifying who will receive this information

## Configuring SA Caching

By default, MSDP SA messages received by the system are forwarded and not cached (stored in memory). A member joining a group after an RP has received a message must wait for the next SA message to be learned. This is known as join latency.

```
nx# configure terminal
nx(config)# ip msdp cache-sa-state list 10
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip msdp cache-sa-state list 10</code>	Enables caching of the SA messages and designates whether or not to use an access list to filter SA entries which are added to the cache. If no access list is configured, all received SA messages are added to the cache. For more information on creating access lists see <a href="#">Chapter 12, "Access List Configuration"</a> .

## Configuring SA Requests

SA messages are sent periodically, updating the MSDP peers about new members. Using the `ip msdp sa-request` command, you can configure the system to send SA Request messages, to specified MSDP peers, when a new member joins a group.

```
nx# configure terminal
nx(config)# ip msdp sa-request 1.1.19.1
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip msdp sa-request 1.1.19.1</code>	Configures the system to send SA Request messages to MSDP peer 1.1.19.1 each time a member joins a group. You must issue this command for each MSDP peer to which you want to send SA Request messages.

## Configuring SA Message Distribution

RPs with known sources distribute SA messages. By default, all sources that are known to an RP are advertised and distributed unless you filter the sources by issuing the `ip msdp redistribute` command. By specifying either an access list or AS path access list you can designate which source and group pairs are distributed.

```
nx# configure terminal
nx(config)# ip msdp redistribute list 110
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip msdp redistribute list 110</code>	Designates using an extended access list to determine which sources to advertise. See <a href="#">Chapter 12, "Access List Configuration"</a> .

## Filtering SA Request Messages

Systems that are configured to cache SA information respond to all SA Request messages unless otherwise configured. You can configure the system to ignore SA Request messages from a specified peer or to use an access list to filter messages from specific groups.

```
nx# configure terminal
nx(config)# ip msdp filter-sa-request 1.14.1.2 list 10
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip msdp filter-sa-request 1.14.1.2 list 10</code>	Assigns an access list to filter which messages to accept from the specified peer.

## Configuring Outgoing SA Message Filtering

By default, the router forwards all the SA messages it receives to all of its MSDP peers. However, you can prevent outgoing messages from being forwarded to a peer by using a filter (or by setting a time-to-live (TTL) threshold value).

Creating filters enables you to filter sources and apply access lists to enable receiving messages only from certain sources. The filter is used for outgoing SA messages sent to specified MSDP-enabled peers.

```
nx# configure terminal
nx(config)# ip msdp sa-filter out 1.1.19.1
nx(config)# ip msdp sa-filter out 1.1.19.1 list 100
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<code>ip msdp sa-filter out 1.1.19.1</code>	Designates filtering all SA messages assigned to the specified peer.
<code>ip msdp sa-filter out 1.1.19.1 list 100</code>	Passes only those SA messages that pass the extended access list to the specified peer.

## Configuring Incoming SA Message Filtering

By default, the system accepts all SA messages sent by MSDP peers. To control the source information the system accepts from an MSDP peer, you can designate filters for incoming SA messages. You can assign filters to:

- Filter all incoming SA messages from a specified peer
- Filter messages based on a specified extended access list

```
nx# configure terminal
nx(config)# ip msdp sa-filter in 1.14.1.2 list 110
```

The following table explains use of commands in the preceding example:

Configuration Line	Description
<code>ip msdp sa-filter in 1.14.1.2 list 110</code>	Enables only incoming SA messages from peer 1.14.1.2 that pass the extended access list.

## Configuring Time-to-Live (TTL) Thresholds

Designating a TTL threshold tells the system to encapsulate data in the first SA message sent out by a source. If the TTL value in the source IP packet is greater than or equal to the threshold, then the data in the first SA message is encapsulated. If the TTL value is less than the threshold, the SA message is sent without encapsulated data.

```
nx# configure terminal
nx(config)# ip msdp ttl-threshold 1.1.19.1 60
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip msdp ttl-threshold 1.1.19.1 60</code>	Sets the TTL threshold to 60 seconds in the first SA message sent from peer 1.1.19.1.

## Optional IGMP Configuration Tasks

The system automatically enables or disables IGMP when PIM is enabled or disabled on an interface. No additional configuration is required, but you can modify the parameters in the following sections in order to tune IGMP behavior.

### Configuring the IGMP Last Member Query Interval

The system enables you to configure the maximum time interval for last-member queries. Last member queries are group-specific queries sent in response to leave-group messages. Leave-group messages are sent when the last member of a specific group leaves the group or is no longer reachable. The shorter the last-member query interval, the less time taken to detect a lost group member.

```
nx# configure terminal
nx(config)# ip igmp last-memb-query-intvl 30
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip igmp last-memb-query-intvl 30</code>	Sets the last member query interval to 30 seconds.

### Configuring IGMP Host Query Message Intervals

The system sends IGMP host-query messages to determine which multicast groups have members on attached networks.

```
nx# configure terminal
nx(config)# ip igmp query-interval 30
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip igmp query-interval 30</code>	Sets the host-query message interval to 30 seconds.



## Configuring the IGMP Version

By default, the system uses IGMP Version 2. This version enables you to configure options like query time-out and maximum query response time. Both IGMP version 1 and 2 are supported but IGMP version 2 is compatible with version 1 hosts.

```
nx# configure terminal
nx(config)# ip igmp version 1
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip igmp version 1</code>	Sets the IGMP version to 1.

## Configuring the Maximum Query Response Time

The system enables you to configure the maximum query response time advertised in IGMPv2 queries on an interface. This interval enables the system to detect when group members are no longer directly connected.

```
nx# configure terminal
nx(config)# ip igmp query-max-response-time 30
```

The following table explains the command in the preceding example:

Configuration Line	Description
<code>ip igmp query-max-response-time 30</code>	Sets the maximum response time for queries to 30 seconds.

## Verifying PIM Configuration

The NX-IS software provides a number of show commands that let you verify configuration. The following tables lists the type of information you can view from each show command:

**Table 15-3. PIM Commands for Verifying Configuration**

Action	Command
Display IGMP enabled interfaces.	<code>show ip igmp interface</code>
Display the multicast groups that are directly connected to the router and were learned via IGMP.	<code>show ip igmp groups</code>
Display the contents of the IP fast switching cache.	<code>show ip mcache</code>
Display the contents of the IP multicast routing table. If you do not enter any arguments, all entries in the IP multicast routing table are shown. The default for this command is to show all groups and sources.	<code>show ip mroute</code>

**Table 15-3. PIM Commands for Verifying Configuration (Continued)**

Action	Command
Display the number of MSDP sources and groups originated in SA messages from each access list. To use this command SA caching must be enabled.	<code>show ip msdp count</code>
Display the current state of local MSDP resources that are sending traffic.	<code>show ip msdp local-sources</code>
Display detailed information about the MSDP peer. If you do not specify a peer the information for all peers is shown.	<code>show ip msdp peer</code>
Display information stored in the cache about MSDP peers.	<code>show ip msdp sa-cache</code>
Display a summary of the current status of all MSDP peers.	<code>show ip msdp summary</code>
Display BSR information, including elected BSR, and locally configured candidate RP advertisement information.	<code>show ip pim bsr-router</code>
Display information about interfaces that are configured for PIM.	<code>show ip pim interface</code>
Display a list of the PIM neighbors discovered by the specified interface.	<code>show ip pim neighbor</code>
Display active cached RPs and any associated multicast routing entries.	<code>show ip pim rp</code>
Display which RP is associated with a specified group.	<code>show ip pim rp-hash</code>
Display the details about the group prefixes of the specified RP candidate.	<code>show ip pim rp-set</code>
Display information regarding the configured shortest path source-tree (SPT) threshold.	<code>show ip pim spt-threshold</code>
Display a summary of the current status of all configured PIM settings.	<code>show ip pim summary</code>

# OSPF Configuration

The Open Shortest Path First (OSPF) protocol is a link-state routing protocol that runs within an autonomous system (AS). Link-state means that when there is a topology change, for example, a link goes down or a router is added to the network, a link-state advertisement (LSA) is sent to all directly connected routers. The routers update their tables and flood LSAs to their neighbors to ensure that all routers have the same topology information in their LSA database(s).

Each router in the AS maintains a database of the shortest paths to all destinations within the AS and calculates the best path (lowest cost) to each destination. The routes are recalculated after any topological change. OSPF relies on the Hello protocol and a flooding mechanism to learn and to disseminate link status information. The hello protocol enables OSPF to learn about neighbors while the flooding mechanism allows for all routers in an AS or area to have synchronized, that is, consistency of information in the link state database.

## Key Features

The OSPF protocol provides many features that efficiently route information across the network. These features include:

- Efficient propagation of information about topology changes, fast convergence, and an hierarchical network structure (autonomous systems and areas) to reduce network traffic overhead.
- Configurable administrative metrics that allow routing based on lowest-cost routes.
- Scalability. OSPF was designed for large networks.
- OSPF is based on an open industry standard.

## Technology Concepts

The following sections give an overview of basic OSPF concepts such as:

- Network Topology
- Hello Protocol
- Link State Protocol
  - Link State Advertisements
  - Link State Database
- Network Types
- Interface Cost
- Authentication
- Packet Formats

## RFCs and Standards

The NX-IS implementation of OSPF is based on the following RFCs:

RFCs and Standards	Title
RFC 2328	OSPF version 2
RFC 1850	OSPF Version 2 Management Information Base

## Network Topology

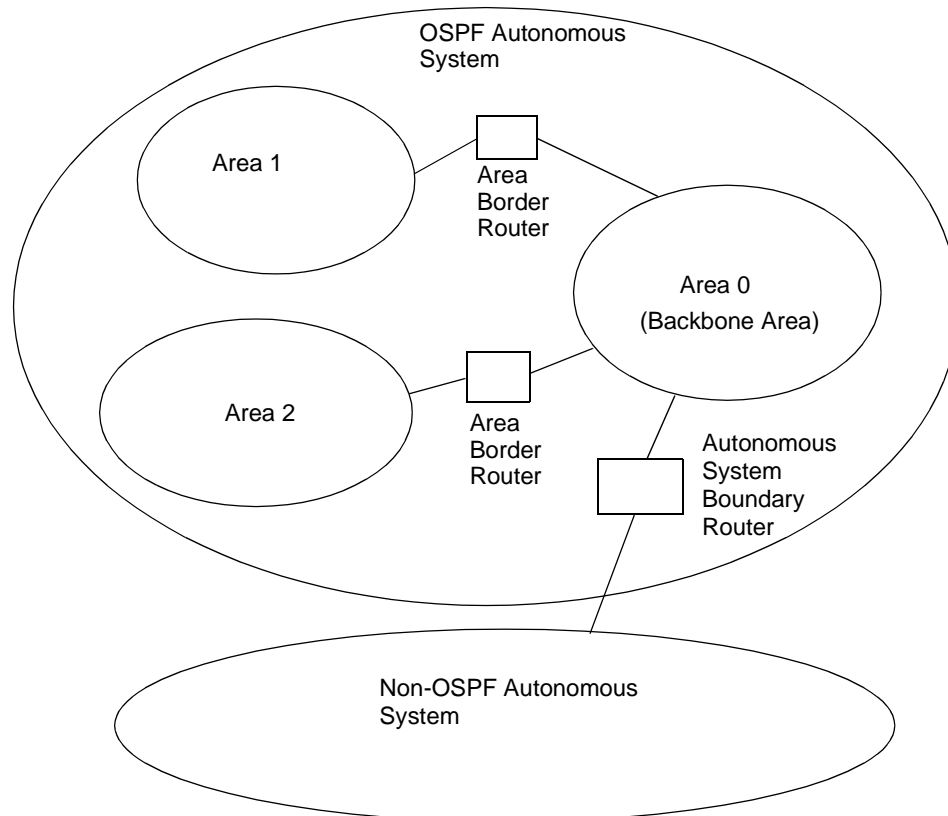
The hierarchical structure of the OSPF network topology is based on the following concepts:

- Autonomous Systems or Routing Domains
- Areas
  - Backbone Area
  - Stub Area
  - Not-So-Stubby-Area (NSSA)
- Area Border Routers (ABR)
- Neighbors and Adjacency
- Designated Router (DR) and Backup Designated Router (BDR)
- Autonomous System Boundary Routers (ASBR)

## Autonomous Systems

OSPF's routing hierarchy groups networks into autonomous systems (AS) or Routing Domains. An AS consists of one or more interconnected areas (or subdomains) that contain a set of routers under a single administration, that is, a single routing protocol (for example, OSPF or IS-IS) and that share the same set of routing metrics and routing information.

Figure 16-1 illustrates an OSPF autonomous system.



**Figure 16-1. OSPF Autonomous System or Routing Domain**

## Areas

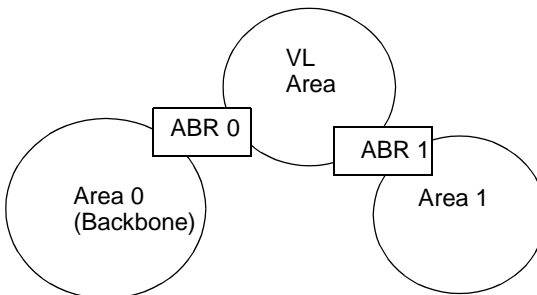
The OSPF protocol limits the number of link state updates and, therefore, network traffic by implementing Areas which are subsets of the OSPF autonomous system. The network overhead is reduced because the internal routers within an area know only the topology of their area and which Area Border Router (ABR) handles packets whose destination is outside the internal router's own area. This information is stored in the internal router's Link State Database (refer to "[Link State Advertisements and Link State Database](#)" for more information). Internal routers in an Area maintain common Link State databases. An Area Border Router (ABR) interfaces with two or more areas and to the backbone and it maintains Link State databases for each of those areas. (Reducing network traffic and limiting the amount of information an internal router has to maintain in its own Link State Database are two features that make OSPF scalable to large networks.)

## Backbone Area

If there is more than one area in the AS, you must assign one of them areaID 0.0.0.0. This area is then referred to as the backbone or transit area. All areas within the AS must be physically or virtually connected to the backbone area by an Area Border Router (ABR) which has routing tables that describe the topology of the backbone area and other areas. OSPF requires that all areas within the AS provide topology status to the backbone area which it then floods to all routers in the AS.

## Virtual Links

If it is not possible to physically connect an area to the backbone, you must connect the area to the backbone by way of a virtual link (configured with the `area virtual-link` command) which provides a logical connection for the area to the backbone. As shown in **Figure 16-2** to create the virtual link (VL) the ABR from the backbone (ABR 0) and the ABR from the area that must be connected to the backbone (ABR 1) must share an area (VL area).



**Figure 16-2. OSPF Virtual Link**

Areas are also defined as stub areas and not-so-stubby areas (NSSA).

## Stub Areas

Stub areas, configured using the `area stub` command, cannot carry AS-external routing information, therefore they are limited to router and network LSAs or information about the topology of their own area. A stub areas use default routes for entry and exit. This prevents AS external advertisements from flooding the area and requiring additional router resources.

Because all routers within a stub area must agree on the fact that the area is a stub area, all routers within the area must have the area configured as a stub. Stub areas are also limited in that:

- They cannot be part of a virtual link configuration.
- AS boundary routers (ASBR) cannot be placed within a stub area.
- If more than one ABR interfaces with the stub area, the ABR with the lowest path cost is used.

## Not-So-Stubby-Areas (NSSA)

Not-So-Stubby-Areas (NSSA), which are configured using the `area nssa` command, are similar to the OSPF stub areas except that they, in a limited way, can import AS external routes. Also unlike stub areas, you can place ASBRs within a NSSA.

## Neighbors and Adjacencies

Neighbors are any two or more routers that have interfaces to a common network. Neighbors are discovered and the neighbor relationship is maintained by the Hello protocol.

An adjacency is established with a neighbor when at least one of the following conditions is true:

- The network type, as defined by the `ip ospf network` command, is point-to-point, point-to-multipoint, broadcast, or non-broadcast.
- ▶ For non-broadcast networks (NBMA), the `neighbor` command is used to designate neighboring routers that are eligible to become adjacencies.
- The neighbor router is either the designated router or the backup designated router (except in point-to-point, point-to-multipoint, and point-to-multipoint non-broadcast).

Therefore, an adjacency is formed when any two neighbor routers exchange routing information on an on-going basis. In OSPF exchanging routing information means that the routers must maintain a common Link State database. However, if every router had to exchange routing information with every other router on the network an excessive amount of traffic would be created.

To limit network traffic within an area, the concept of designated router (DR) and backup designated router (BDR) was implemented. A new router on the network learns of the routers with which it must maintain connectivity and then forms a full adjacency with the DR and BDR. The DR and BDR have adjacency with each other and with all other routers in the area as well so link state updates are disseminated and learned through the DR and BDR.

## Designated Router (DR) and Backup Designated Router (BDR)

Any OSPF broadcast and non-broadcast (NBMA) network that has at least two connected routers must have a DR. Since the DR is adjacent to all other routers on the network, its primary purpose is that by learning and distributing/disseminating all link status, it helps maintain the Link State Database (refer to [“Link State Advertisements and Link State Database”](#) for more information) on each router synchronized, that is, containing the same information. By being the only router on a given network to disseminate LSA information, the DR reduce the number of adjacencies required in a network and, in turn reduces, the amount of routing protocol traffic.

As its name indicates, the BDR is there to take over the duties of the DR in case that router fails. Since the DR and BDR are always fully synchronized, the disruption in case of DR failure is minimal the only thing that needs to happen is that new LSAs have to be disseminated to announce the new DR. Once the BDR becomes DR another router is elected BDR by the process described below.

## Designated Router and Backup Designated Router Selection

The DR and BDR are established as follows. When a router comes on the network, if it doesn't receive hello packets from any other router, it assumes that it is the only router on the network, declares itself the DR and sets the DR field in its hello packet to indicate that fact. When a second router comes on, it will see that there is already a DR (when it receives that DR's hello packet) but if it doesn't receive hello packets from other routers indicating the presence of a BDR, it elects itself the BDR and sets the BDR field in its hello packet accordingly.

If, in a network where no DR has been established yet, hello packets are transmitted at the same time from multiple routers so no router can elect itself as DR based on timing of the hello packet being transmitted, the assignment goes to the router with the highest router priority. (Priority 0 routers do not become DRs or BDRs since they don't have enough resources.) If there is a conflict with the router priority as well, the DR is elected based on the routerID so the router ID becomes the final tie-breaker. The same happens with the election of the BDR if no decision can be made because of simultaneous transmission of hello packets.

## Autonomous Systems Boundary Router (ASBR)

In the OSPF AS, OSPF is used for routing within an area (intra-area routing) and routing between areas (inter-area routing). An external gateway protocol is used for routing beyond the OSPF network.

As shown in [Figure 16-1](#), an ASBR links an OSPF AS with a non-OSPF network. In an NX-IS network, an ASBR runs both the OSPF protocol for routing within the AS and a protocol, such as BGP or IS-IS, to allow routing between the OSPF network and a non-OSPF network. (See or [Chapter 17, "IS-IS Configuration"](#) or [Chapter 18, "BGP Configuration"](#) for more information on route redistribution into a non-OSPF network.)

## Hello Protocol

OSPF uses the hello protocol packets for neighbor discovery and to confirm neighbor status and the status of the connection between neighbors. (Refer to ["OSPF Packets Format"](#) for information on Hello packet format.) Therefore, neighbors send hello packets to each other to announce their presence and to verify connectivity. When a router receives hello packets from a neighbor, it knows that the neighbor is alive and there is two-way or bidirectional connectivity between them. Hello packets are also used to establish the DR and the BDR in an AS. (Refer to ["Designated Router \(DR\) and Backup Designated Router \(BDR\)"](#) for information on establishing a DR and a BDR in the AS.)

You can set the interval, in seconds, between hellos (`ip ospf hello-interval`) and can also set how long (in seconds) a router must wait before declaring a neighbor down due to lack of hello packets from the neighbor (`ip ospf dead-interval`). The value of the "hello interval" (the default is 10 seconds) and the value of the "dead interval" (the default is 40 seconds) must match on all routers on a common network. The hello intervals come with the usual trade offs — a short interval allows quick detection of a downed neighbor but creates more traffic because of more frequent transmission of hello packets.

OSPF Hello packets are multicast to those router interfaces that have OSPF enabled on them but, for testing purposes, interfaces can also be configured not to send hello packets (`passive-interface`) so that those interfaces do not form adjacencies.



- The multicast addresses are 224.0.0.5 and 224.0.0.6.

The frequency of topological change calculations is based on intervals specified using the `timers spf` command.

## Link-State Protocol

Link-state protocols, like OSPF and IS-IS, exchange routing information between routers within an autonomous system (AS). An AS is a group of routers that share routing information and use the same protocol. Using the shortest path first (SPF) routing algorithm to determine the network topology and status, these link-state protocols minimize routing overhead and speed convergence, a state or time when all routers in the network have synchronized databases, that is, they all have the same network topology information.

Link-state protocols depend on each router in the network learning the network's complete topology. This is accomplished by each router in the network learning of its set of neighbors, which comprise the router's local topology. The router then floods its local topology to all neighbors including the area's ABR. The ABR forwards this information to the backbone area which, in turn disseminates it to other areas in the AS. Convergence happens quite fast but depending on the size and complexity of the network, from the time a change in status is detected, it may take several seconds for the information to be disseminated and for routers to update their Link State database(s) and recalculate their routing tables.

### Link State Advertisements and Link State Database

OSPF uses link-state advertisements (LSAs) to exchange link state information, that is, to transmit routing updates and synchronize routing tables. LSAs are generated and broadcast either when a link state changes. Each LSA has a unique sequence number (LSP ID) which is incremented by 1 after each LSA is sent, regardless of whether the information in the LSA has changed. The smaller the sequence number, the older the LSA.

A LSD is composed of Link State Advertisements (LSAs) and describes the topology of the AS that the router belongs to. Routers in the same area have identical LSD which are kept synchronized by the flooding process, that is, the process of disseminating link state information/updates to all the routers in the area. If a router belongs to more than one area then it has a LSD for each area.

Each router in an OSPF network builds a shortest path tree with itself as the root of the tree from its LSD. If the router belongs to more than one area, it builds a tree from the database for each area. Routing tables that include network destination information and metrics, such as route costs, associated with the routes are then calculated based on the shortest-path tree information.

This implementation of OSPF includes 7 types of Link State Advertisements which are described in [Table 16-1](#).

**Table 16-1. Link State Advertisements**

LSA Type	LSA Name	Description
1	Router LSA	It is generated by every router for every area in which it has an OSPF interface. This LSA provides information on the state of that router's interface for the area. The Router LSA is an intra-area advertisement, therefore, it is flooded only to the applicable area.
2	Network LSA	It is generated by an area's DR and lists the routers connected to network. It is flooded to designated router's own area only.
3	Summary LSA for IP network	It is generated by the ABRs and specifies the routes to destinations (networks) inside the area. This LSA is flooded to the area associated with the LSA, that is, the area originating the LSA.
4	Summary LSA for ASBR	It is generated by the ABRs and specifies the routes to AS's boundary routers. This LSA is flooded to the area associated with the LSA, that is, the area originating the LSA.
5	AS-External LSA	It is generated by ASBR and specifies the routes to a destination in another AS. This LSA is flooded throughout the AS from which it originates.
7	AS-External LSA	Type 7 AS-External LSA applies to NSSAs only. Type 7 LSA is generated by an NSSA and flooded within the NSSA only.
10	Area, scope opaque	For Traffic Engineering enabled interfaces.

## OSPF Network Types

OSPF supports the following networks types which are configured using the `ip ospf network` command:

- Point-to-point - joins a pair of OSPF routers.
- Broadcast - supports multiple routers and makes it possible for a single message to be propagated to all connected routers.
- Non-broadcast which can be:
  - Non-broadcast multi-access (NBMA) - provides connectivity but no broadcasting function.
  - Point-to-multipoint - all connections between OSPF routers behave as point-to-point links

## Interface Cost

Metrics influence route preference by associating a cost with a route or adjacency. This cost is used by the SPF algorithm to calculate the best path to a destination. You assign a cost to an interface using the `ip ospf cost` command. However, if the redistributed route has a cost specifically assigned to it, you cannot override that cost.

## Authentication

OSPF supports authentication for all protocol exchanges. The AuType field in packet header, configured on a per-interface basis using the `ip ospf authentication-key` command, specifies the type of authentication to be used. The current implementation supports two types of authentication:

- Type 0 - null or no authentication of routing exchanges.
- Type 1 - simple password authentication. With this type of authentication a password (text-string) is configured per area. Each router must first be configured with its networks' passwords before it can participate in routing. Thus, password authentication is a way of ensuring that routers are restricted to or recognized only by their own domain.

The AuType must be the same for all routers within an area.

## OSPF Packets Format

Following is a description of basic OSPF packets formats including:

- Common Header
- Hello packet
- LSA Header

For description of LSA and other OSPF packets, refer to the applicable RFC.

## Common OSPF Header Format

OSPF uses a common 24-byte header for all packets. This header is shown in **Figure 16-3**.

Version	Type	Packet Length
Router ID		
Area ID		
Checksum		AuType
Authentication Type		
Authentication Data		

**Figure 16-3. Common OSPF Header**

The header fields are defined in **Table 16-2**.

**Table 16-2. Common OSPF Header Fields**

Field	Description
Version	OSPF's version number.
Type	Identifies the packet as: <ul style="list-style-type: none"><li>• Type 1 - Hello packet used to discover and maintain neighbors.</li><li>• Type 2 - Database Description packet to summarize database contents.</li><li>• Type 3 - Link State Request packet to request a Database download.</li><li>• Type 4 - Link State Update packet to request a Database update.</li><li>• Type 5 - Link State Acknowledgment packet to acknowledge flooding.</li></ul>
Packet Length	The length of the packet including the header.
Router ID	The ID of the router transmitting the packet.
Area ID	Identifies the OSPF area that the packet is associated with.
Checksum	Checksum for the content of the packet including the header but excluding the authentication field. Packets with incorrect checksum are discarded.

Field	Description
AuType	Identifies the Authentication type. The current NX-IS implementation supports: <ul style="list-style-type: none"> <li>Type 0 - null or no authentication</li> <li>Type 1 - password authentication</li> </ul>
Authentication	The content of this field (a password) is used with the AuType for authentication.

## OSPF Hello Packet

Figure 16-4 shows the format of an hello packet.

Common OSPF Packet Header		
Network Mask		
Hello Interval	Options	Rtr Pri
Router Dead Interval		
Designated Router		
Backup Designated Router		
Neighbor		
Neighbor		

**Figure 16-4. OSPF Hello Packet**

The fields of the hello packet are defined in Table 16-3.

**Table 16-3. OSPF Hello Packet Fields**

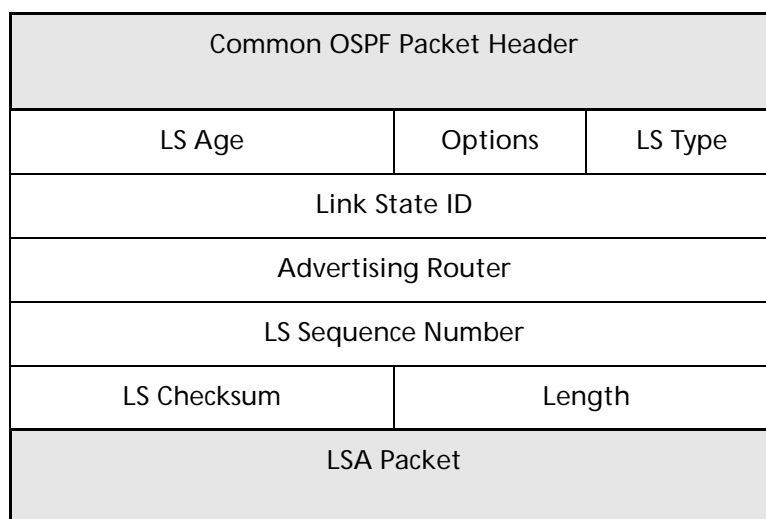
Field	Description
Network Mask	Network mask for the interface
Hello Interval	The number of seconds that must elapse between transmission of hello packets from this router
Options	Certain router's capabilities must match for routers to become neighbors. Refer to RFC 2328 for more information.

**Table 16-3. OSPF Hello Packet Fields**

Field	Description
Rtr Pri	The router priority that, if necessary, is used to determine the designated router or the backup designated router.
Router Dead Interval	The number of seconds that must elapse before this router can define a neighbor as down due to lack of hellos from the neighbor.
Designated Router	Identifies the designated router.
Backup Designated Router	Identifies the backup designated router
Neighbor	Router IDs from neighbors whose hello packets have been received within the allowed dead time interval and, therefore, considered “live” neighbors.

## OSPF LSA Header

The OSPF LSA packets have a common 20 byte header (shown in [Figure 16-5](#)) that, in the packet, follows the common OSPF packet header.



**Figure 16-5. Common LSA Header**

The fields of the LSAs' header are described in [Table 16-4](#).

**Table 16-4. Common LSA Header**

Field	Description
LS Age	The age, in seconds, of the link state advertisement

**Table 16-4. Common LSA Header**

Field	Description
Options	Indicates the optional capabilities are associated with the advertisement. Refer to RFC 2328 for more information.
LS Type	Indicates the type or function of the link state advertisement as: <ul style="list-style-type: none"> <li>• Type 1 - Router LSA</li> <li>• Type 2 - Network LSA</li> <li>• Type 3 - IP Network Summary LSA</li> <li>• Type 4 - ASBR Summary LSA</li> <li>• Type 5 - AS-External LSA</li> <li>• Type 7- AS-External LSA for NSSAs</li> <li>• Type 10 - Area, scope opaque</li> </ul>
Link State ID	This field identifies the element of the routing domain being described by the advertisement.
Advertising Router	Specifies the ID of the router originating the LSA.
LS Sequence Number	This field is used to detect old and duplicate link state advertisements. Smaller sequence numbers indicate older LSAs.
LS Checksum	The checksum of the contents of the advertisement excluding the LS Age field. Packets with incorrect checksum are discarded.
Length	The length in bytes of the link state advertisement. It includes the length of the 20 byte header.

## OSPF Configuration

This section provides examples of basic OSPF configuration tasks. Refer to the first part of this chapter for an overview of the protocol operation. **Table 16-5** lists the OSPF commands that comprise the current implementation and their application in configuring and/or maintaining an OSPF network.



All commands referenced in this chapter are described in the *NX64000 Command Reference* manual.

**Table 16-5. OSPF Command Usage**

Command	OSPF Enable	Basic configuration	Area Configuration	Authentication	Route Cost	Redistribution	Verification and Debug
area default-cost					✓		
area nssa			✓				
area range			✓				
area stub			✓				
area virtual-link			✓				
auto-cost					✓		
default-information originate						✓	
default-metric					✓		
distance ospf					✓		
ip ospf authentication-key				✓			
ip ospf cost					✓		
ip ospf dead-interval		✓					
ip ospf hello-interval		✓					
ip ospf network		✓					
ip ospf priority		✓					
ip ospf retransmit-interval		✓					
ip ospf transmit-delay		✓					
neighbor		✓					
network		✓					
no area			✓				



**Table 16-5. OSPF Command Usage**

Command	OSPF Enable	Basic configuration	Area Configuration	Authentication	Route Cost	Redistribution	Verification and Debug
passive-interface							✓
redistribute						✓	
router ospf	✓						
show ip ospf							✓
show ip ospf border-routers							✓
show ip ospf database							✓
show ip ospf interface							✓
show ip ospf neighbor							✓
show ip ospf virtual-links							✓
summary-address						✓	
timers spf		✓					

## Configuration Overview

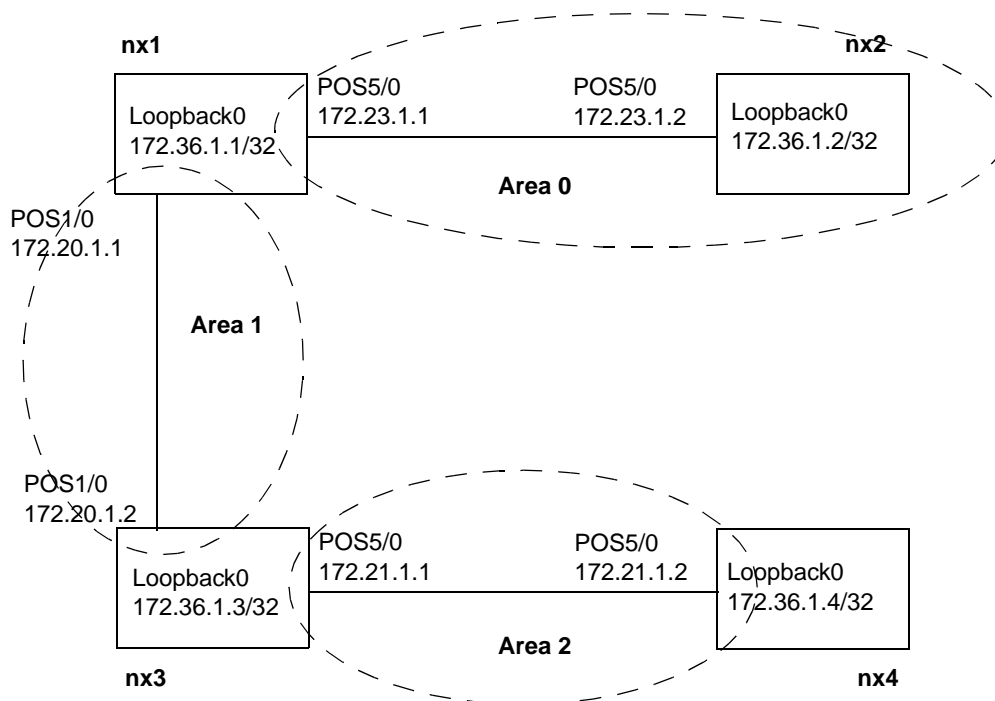
In general, to configure OSPF on an NX router, you must minimally configure the following:

- load the protocol
- enable the protocol on the router
- assign interfaces to an area

Other examples include:

- assigning virtual links
- setting passwords
- creating a stub area
- redistributing static routes

**Figure 16-6** illustrates the OSPF network as configured in this guide:



**Figure 16-6. Sample OSPF Network**

### Loading and Enabling OSPF

After having configured the interfaces (refer to Cards and Interfaces chapter of this guide), load and enable OSPF on the router. To complete these steps, you must load the protocol, enter OSPF router configuration mode, and enable the protocol on an interface by assigning the interface to an area.

NX1:

```
nx1# configure terminal
nx1(config)# load spf
nx1(config)# router ospf
nx1(config-router)# network 172.23.0.0 0.0.255.255 area 0
nx1(config-router)# network 172.20.0.0 0.0.255.255 area 1
```

NX2:

```
nx2# configure terminal
nx2(config)# load spf
nx2(config)# router ospf
nx2(config-router)# network 172.23.0.0 0.0.255.255 area 0
```

NX3:

```
nx3# configure terminal
nx3(config)# load spf
nx3(config)# router ospf
nx3(config-router)# network 172.20.0.0 0.0.255.255 area 1
nx3(config-router)# network 172.21.0.0 0.0.255.255 area 2
```

NX4:

```
nx4# configure terminal
nx4(config)# load spf
nx4(config)# router ospf
nx4(config-router)# network 172.21.0.0 0.0.255.255 area 2
```

The following table explains each line in the preceding example:

Configuration Line	Description
<b>load spf</b>	Loads OSPF from software release 1.7.0.
<b>router ospf</b>	Enables OSPF on the router and enters OSPF router configuration mode.
<b>network ip-address mask area area-id</b>	Assigns interfaces to an area based on the specified address/mask pair. Interfaces that fall within the range are enabled for OSPF and assigned to the specified area. For example, interfaces that fall within the 172.21.x.x range on NX4 are assigned to area 2. An interface can only belong to one area.

## Configuring Authentication

This implementation of OSPF supports interface-based simple password authentication. That is, each interface must be configured for authentication. Only those interfaces with a matching passwords establish connections. The following example sets passwords on NX1. Interface POS5/0, a member of area 0, is configured with a password of *backbonerouter*. Interface POS1/0, a member of area 1, is configured with a password of *what's up doc*. (You must use quotes to allow spaces between words.) Members of area 2 use the password *stub2*.

NX1:

```
nx1(config)# interface pos5/0
nx1(config-if)# ip ospf 1 authentication-key backbonerouter
nx1(config-if)# exit
nx1(config)# interface pos1/0
nx1(config-if)# ip ospf 1 authentication-key "what's up doc"
```

NX2:

```
nx2(config)# interface pos5/0
nx2(config-if)# ip ospf 1 authentication-key backbonerouter
```

NX3:

```
nx3(config)# interface pos1/0
nx3(config-if)# ip ospf 1 authentication-key "what's up doc"
nx3(config-if)# exit
nx3(config)# interface pos5/0
nx3(config-if)# ip ospf 1 authentication-key stub2
```

NX4:

```
nx4(config)# interface pos5/0
nx4(config-if)# ip ospf 1 authentication-key stub2
```

## Configuring Virtual Links

When your network has area border routers that are not directly connected to the backbone, you must create virtual links to ensure the integrity of your network. In this guides configuration, NX3 is an ABR (member of areas 1 and 2), but does not have a direct connection to the backbone. Setting up a virtual link through NX1 establishes NX3's backbone connection.

When configuring virtual links, you must configure the link on both endpoints of the connection. In the following example, the configuration is implemented for NX1 and NX3, while NX2 and NX4 remain unchanged. Each router's loopback interface is designated as the endpoint of the link. In addition, because the rest of the network has authentication enabled, the virtual link must use a password to control admittance to the backbone. Finally, each loopback interface is enabled for OSPF.

- There are several optional parameters available when configuring virtual links. See the *NX64000 Command Reference* for a complete command description.

NX1:

```
nx1(config)# router ospf
nx1(config-router)# area 1 virtual-link 172.36.1.3 authentication-key
backbone router
nx1(config-router)# network 172.36.1.1 0.0.0.0 area 1
```

NX3:

```
nx3(config)# router ospf
nx3(config-router)# area 2 virtual-link 172.36.1.1 authentication-key
backbone router
nx3(config-router)# network 172.36.1.3 0.0.0.0 area 2
```

The following table explains each line in the preceding example:

Configuration Line	Description
<b>router ospf</b>	Enters OSPF router configuration mode. (OSPF was enabled in a previous step.)

Configuration Line	Description
<pre>area area-id virtual-link router-id authentication-key backbonerouter area 1 virtual-link 172.36.1.3 area 1 virtual-link 172.36.1.1</pre>	<p>Configures the router as an endpoint of a virtual link. The specified area ID is the transit area, and must be the same on both routers (area 1 in this example). The router ID (in this example each router's loopback address) must identify the router at the other end of the link. Finally a password controls access to the backbone, since authentication has been configured for this network.</p>
<pre>network 172.36.1.1 0.0.0.0 area 1 network 172.36.1.3 0.0.0.0 area 2</pre>	<p>Assigns interface 172.36.1.1 (NX1's loopback interface) to area 1.</p> <p>Assigns interface 172.36.1.3 (NX3's loopback interface) to area 2.</p>

## Creating a Stub Area

Because stub areas do not accept external routes and use a default route as a gateway, the routers within have reduced routing database sizes and memory requirements. When configuring a stub area, any router within the area must be configured as a stub router. In the following, area 2 is configured as a stub area. The no-summary option prevents flooding of OSPF summary routes.

NX3:

```
nx3(config)# router ospf
nx3(config-router)# area 2 stub no-summary
```

NX4:

```
nx4(config)# router ospf
nx4(config-router)# area 2 stub no-summary
```

## Using Summary Addresses

OSPF supports two methods for summarizing addresses.

- One method, which is specific to advertising into an NSSA, is by using the summary-address command that aggregates AS external type 5 LSAs to type 7 LSAs.

```
nx1(config)# router ospf
nx1(config-router)# summary-address 172.31.1.1 255.255.255.0
```

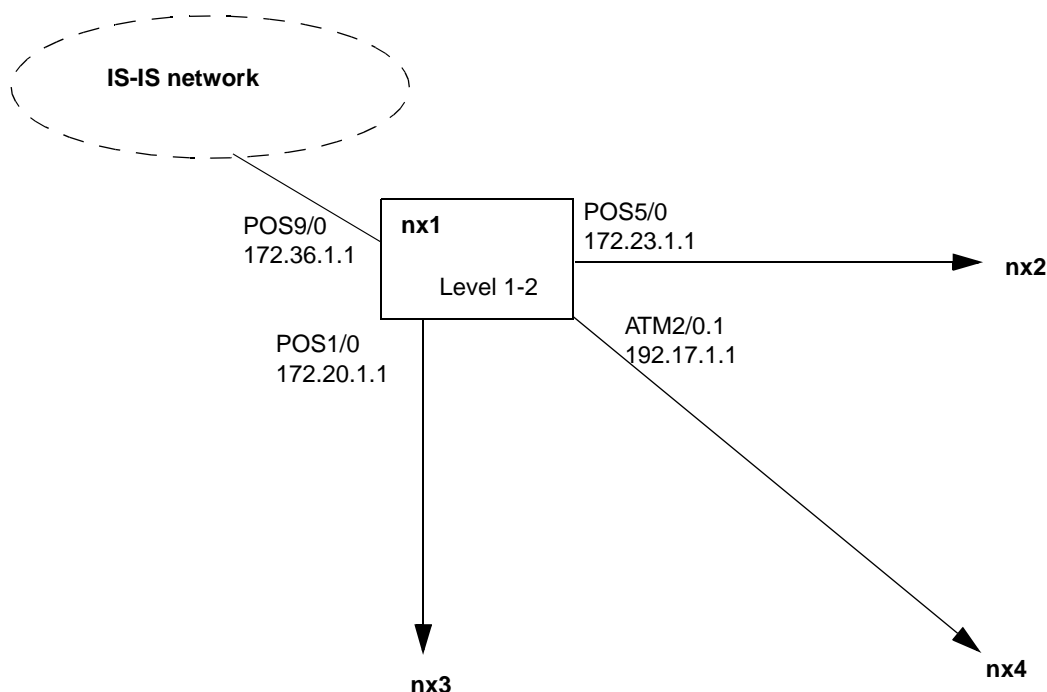
- The second method using the area range command is to configure an ABR to summarize routes according to a range specified.

```
nx1(config)# router ospf
nx1(config-router)# area 1 range 172.0.0.0 255.0.0.0
```

## Route Redistribution

Redistribution alters the routing scheme to allow routes from another protocol, such as IS-IS, into the OSPF routing table. For example, a company merged with another and now they want to merge their two networks. One is an OSPF network and one an IS-IS. To pass routing information between the two parts of the network, routers on the border must be running both OSPF and IS-IS, and then must redistribute routes between the two protocols. The diagram below modifies the original by adding the IS-IS network off of NX1. (Only the NX1 portion is shown.) Assume NX1 is configured to run IS-IS as well as OSPF, and forms the border between the two domains.

- ▶ While route redistribution is possible, it is not recommended because for one thing it increases the size of routing tables.



**Figure 16-7. OSPF Configuration Example with IS-IS Addition**

The following configures NX1 to redistribute between the two protocols (i.e., IS-IS routes into the OSPF and OSPF routes into the IS-IS network).

NX1:

```
nx1(config)# router ospf
nx1(config-router)# redistribute isis
nx1(config-router)# exit
nx1(config)# router isis
nx1(config-router)# redistribute ospf
```

## Verifying OSPF Configuration

The NX-IS software provides a number of **show** commands that let you verify your OSPF configuration. The following table lists the type of information you can view from each show command:

**Table 16-6. OSPF Commands for Verifying Configuration**

Action	Command
Display various configured parameters for an OSPF area, including: <ul style="list-style-type: none"> <li>• process and route identifier</li> <li>• types of services supported</li> <li>• SPF algorithm information</li> <li>• configured areas</li> <li>• area range summaries</li> </ul>	<b>show ip ospf</b>
Display routing table entries for ABRs and ASBRs, including: <ul style="list-style-type: none"> <li>• destination and next hop</li> <li>• configured interfaces and associated cost</li> <li>• destination router type</li> <li>• route type</li> <li>• area that route was learned from</li> </ul>	<b>show ip ospf border-routers</b>
Display a router's LSA database and a summary of the neighboring LSA database. Use this to verify that peerings have formed and that adjacencies exist. Information includes: <ul style="list-style-type: none"> <li>• link ID</li> <li>• advertising router</li> <li>• LSA age and sequence number</li> <li>• checksum value</li> </ul>	<b>show ip ospf database</b>
Verify that OSPF is running on the intended interface and display detailed interface information, including: <ul style="list-style-type: none"> <li>• name, type, and state</li> <li>• IP address, mask, and router ID</li> <li>• interface cost</li> <li>• priority setting</li> <li>• timer information</li> <li>• neighbor information</li> </ul>	<b>show ip ospf interface</b>

**Table 16-6. OSPF Commands for Verifying Configuration**

Action	Command
Display information about a specific neighbor or lists all neighbors, including: <ul style="list-style-type: none"><li>• IP address</li><li>• priority setting</li><li>• remaining timer</li><li>• neighbor IP address</li></ul>	<code>show ip ospf neighbors</code>
Display virtual link information, including: <ul style="list-style-type: none"><li>• ID and state of the link</li><li>• transit area and delay</li><li>• timer settings</li><li>• adjacency state</li></ul>	<code>show ip ospf virtual-links</code>
Display routes used by this router.	<code>show ip route</code>



## IS-IS Configuration

The Intermediate System-to-Intermediate System (IS-IS) protocol is an interior gateway protocol (IGP) designed to allow a single system to route both IP and Open Systems Interconnection (OSI) traffic. (OSI traffic is in the form of Connectionless Network Protocol (CLNP) packets, the ISO's answer to IP.) IS-IS is a single, integrated protocol that can route, by area, either IP, OSI, or “dual” traffic, and uses a shared backbone for the routing domain. Packets are forwarded without any type of encapsulation. The current implementation supports only IP routing.

IS-IS is a link-state protocol. It is similar to OSPF in that it uses link state advertisements (LSAs) sent to all routers within a given area and hello packets to detect whether a router is still functioning.

### Key Features

IS-IS was designed for large networks, providing the following features:

- Simplified LSPs that consume less bandwidth
- Route summarization resulting from the routing hierarchy provides fast convergence
- Use of variable-length subnet masks that conserves address space
- Configurable administrative metrics that allow more granular control

## Technology Concepts

IS-IS is a link state protocol, meaning that each router maintains a table that tracks the best route to any known destination. All systems have a copy of the table, and use link-state packets (LSPs) to transmit routing updates and synchronize the table as needed. Because of IS-IS's simplified (compared to OSPF) method of providing link-state protocol benefits, it works well for large networks. Basic to understanding IS-IS are:

- Routing levels
- Network topology
- Routing domain types
- Link-state protocol
- IS-IS adjacencies
- Routing selection process
- Authentication
- Designated routers

## RFCs and Standards

This implementation of IS-IS is based on the following Request For Comments (RFCs), enabling standardized IS-IS on the systems:

**Table 17-1. RFCs Implementing Standardized IS-IS**

Standard	Title
RFC 1195	Use of OSI IS-IS for Routing in TCP/IP and Dual Environments
RFC 1142	OSI IS-IS Intra-domain Routing Protocol
ISO 8473	The ISO Connectionless Network Layer Protocol
ISO 9542	The ISO End System to Intermediate System Protocol
ISO DP 10589	OSI IS-IS Intra-domain Routing Protocol

## Defining an Intermediate System

An intermediate system is OSI terminology for an Internet router. That is, it is not an end system or destination, but a system between end systems that transfers packets and provides relaying functions. An intermediate system does not function above layer 3 (the network layer) except for some network management functionality.

## Routing Levels

Beyond the physical division of networks, IS-IS uses a routing hierarchy to divide a network into areas (see [“Network Topology” on page 17-4](#) for more information). Routers route to a destination based on the addresses of these areas. By assigning levels to a router and a circuit, you configure the network topology. Routers can be configured as either level 1 or level 1 and 2; circuits can be configured as level 1 only, level 2 only, or level 1 and 2. The table below gives a brief summary of each level function and the following sections describe them in more detail:

**Table 17-2. Summary of Routing Levels**

Router Level	Summary Description
Level 1	Router forms adjacencies within an area (intra-area).
Level 2	Router forms adjacencies between areas (interarea).
Level 1 and 2	Router can function as either intra- or interarea.

When you assign a router to a specific level, you set the function of the router (who the router can communicate with). When you configure an interface's circuit type, you control formation of adjacencies with neighbors over that interface. For example, if you want to create a level 2-only network, set the router to level 1-2 and all circuits to level 2.

### Level 1 Routing

Level 1 routing is intra-area, that is, the router maintains information on systems within its area. This allows routers to form intra-area adjacencies with routers that share a common area. Level 1 routers know the topology in their area, but do not have a view of routers outside of the area. Any traffic with a destination outside the area is forwarded to the local level 2 router. A level 1 router recognizes its own (and other router's areas) by the area (or ID) portion of its address, which is manually configured.

### Level 2 Routing

Level 2 routing is interarea (across the backbone), and is the only routing type that can exchange information with routers outside of the routing domain. Level 2 routing adds hierarchy, allowing you to break up large networks. Level 2 routers establish Level 2 adjacencies with other level 2 routers, or with level 1-2 routers. Level 2 routers know the network topology of the other level 2 routers in their AS, and those destinations reachable through them.

In some cases, a level 2 router may form level 1 adjacencies if it is the only available system and shares an area with the other level 1 router. Other than in this case, the level 2 router routes towards an area, but is unaware of the area's internal structure.

## Level 1-2 Routing

Level 1-2 routing, which is the default setting for NX64000 routers, sets the router to establish a level 1 adjacency if there is at least one common area address with the neighbor. Otherwise, if the other side is either level 2 or level 1-2, a level 1-2 router establishes a level 2 adjacency. With this setting, the router functions as both a level 1 and level 2 router, maintaining separate link-state databases, and sending separate updates for each. When a level 1-2 router establishes an adjacency with a neighbor, and the two routers have no common area address, both routers record the connection as level 2 only.

## Network Topology

An IS-IS network is divided into routing domains and areas. It is these network components that define the level of routing that occurs. **Figure 17-1** depicts the components of the IS-IS network.

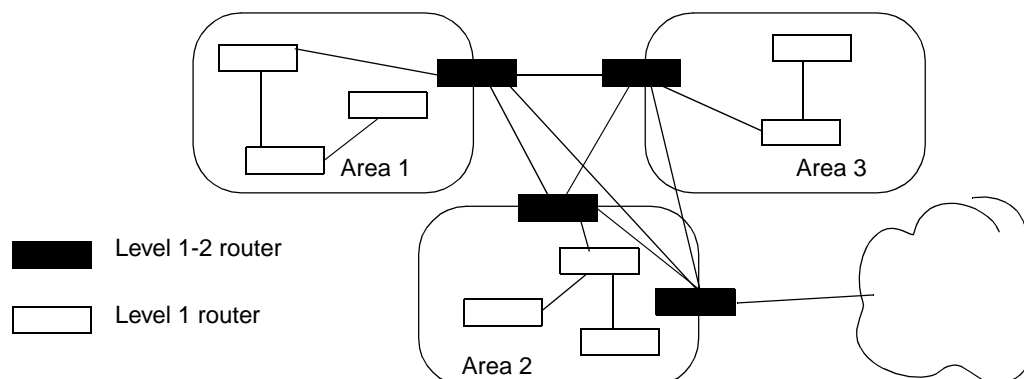
### Areas

An area is a level 1 subdomain. It is comprised of contiguous networks that all share a common area address. An area can have multiple area addresses, in which case a packet containing any one of the addresses is routed to that area. Routers within an area know that area's internal structure and know which routers are able to reach outside of the area. Routing within the area corresponds to level 1 routing. Only a level 2 router can route to an area address outside of the area. Dividing a domain into areas allows a hierarchical network structure, which minimizes distribution of routing information.

### Domains

An IS-IS routing domain is a group of interconnected areas (or subdomains) that contain a set of routers under a single administration. That is, a single routing protocol and the same set of routing metrics. The formal definition of a domain, from the "OSI Routing Framework" specifies "that all intermediate systems within a Routing Domain can determine whether an ES within the domain is reachable, and if so can derive a path to it." A routing domain is the equivalent to BGP's autonomous system (AS).

The domain uses one—and sometimes more than one—IGP to route packets internally, and/or an exterior gateway protocol (EGP) to route packets outside of the domain. The router uses IS-IS or OSPF within the domain and BGP to reach external ASs. No IS-IS routing messages are sent outside of the domain.



**Figure 17-1. The IS-IS Network Hierarchy**

## Routing Domain Types

Integrated IS-IS supports three types of routing domains:

- OSI-only routing domains route only OSI packets
- IP-only routing domains route only IP packets
- Dual routing domains route both OSI and IP packets.

► This implementation supports the IP-only routing domain. That is, it routes in a pure IP environment, but allows interconnection with third-party systems that support dual routing domains.

### IP-Only Routing Domains

In an IP-only routing domain, all routers must support IP and be configured as either IP-only or dual IS-IS routers. All OSI packets, except those necessary for the operation of the protocol, are discarded. OSI packets destined for dual routers are accepted into the routing domain, but ignored by the IP-only routers.

### OSI-Only Routing Domains

In an OSI-only routing domain, all routers must support OSI and be configured as either OSI-only or dual IS-IS routers. All IP packets, except those destined for the dual routers, are discarded.

### Dual Routing Domains

In a dual routing domain, IP-only, OSI-only, and dual routers can be mixed in the domain, but not in the areas. Each area must be either dual, pure IP, or pure OSI. For example, a pure IP area can contain IP-only and dual routers, but only IP traffic is routed by level 1 routers. To route between IP and OSI areas, you must use dual level 2 routers.

## Understanding a Link-State Protocol

Link-state protocols, like IS-IS or OSPF, exchange routing information between routers within an autonomous system (AS). An AS is a group of routers that share routing information and “speak” the same protocol. Using the shortest path first (SPF) Dijkstra routing algorithm to determine the network topology and status, link-state protocols minimize routing overhead and speed convergence time.

Link-state protocols are based on each router in a network learning the network’s complete topology. This is accomplished by each router in the network learning of its set of neighbors, which comprise the router’s local topology. The router then floods its local topology to all neighbors. Once a router has information about all its neighbors’ routes, it can calculate the route to all other routes in the AS. When a router detects a link change, it floods the information to all neighbors, and each router then recalculates its routing tables.

## Debugging and Logging Facilities

The NX-IS implementation provides debugging and debug message logging facilities to help you troubleshoot IS-IS problems. In addition, show commands plus the ping command can be used to troubleshoot connections and connectivity. For more information on troubleshooting IS-IS problems, refer to the *NX64000 Troubleshooting Guide*.

## IS-IS Packet Types and Formats

IS-IS packets are used for neighbor discovery and database synchronization. The following sections describe the purpose and basic format of each packet type.

### IS-IS Packet Types

IS-IS packets transmit information between neighbors. There are three basic packet types:

- Hello packets
- LSP packets
- Sequence number packets

### Hello PDUs

IS-IS hello packets are used for neighbor discovery and link status information. Neighbors send hello packets to each other to announce their presence, and respond to hello packets to verify connectivity. Using a user-configurable timer, you can set the interval, in seconds, between hellos. The interval determines how long a loss of a connection goes undetected. The timer comes with the usual trade offs—a short interval allows quick detection of a downed neighbor, but creates more traffic with more frequent hello packets and responses. In addition, you can set a hello multiplier. This number determines how many missed hello packets are allowed before the connection is deemed unreachable. Finally, you can block an interface from sending hello packets entirely.

Hellos are not explicitly acknowledged with any other packet, but IS-IS does ensure that the other end is receiving the hello. On a LAN, each router includes a list of all the other nodes it has seen hellos from (and hence, has adjacencies with) on the LAN in its hello packet. This way a node can tell that other node has seen its hello if it sees itself listed in the hello from that node.

There are three types of hello packets:

**Table 17-3. Hello PDU Types**

Hello PDU Type	Description
Level 1 LAN	Broadcast to all Level 1 routers on the LAN.
Level 2 LAN	Broadcast to all Level 2 routers on the LAN.
Point-to-Point	For non-broadcast media.

The packet format for these hello PDUs are generally the same, except that the LAN hellos contain a LAN ID and the point-to-point hello contains a local circuit ID. The fixed header fields in common include:

**Table 17-4. Hello Packet Additional Fixed Header Information**

Field	Description
Circuit type	A value of 1 (Level 1 only circuit), 2 (Level 2 only circuit), or 3 (both Level 1 and Level 2 circuits).
Source ID	The sending (source) router's network-layer address (system ID).
Holding time	The length of time a router holds a neighbor connection without response. This value is three times the user-configurable hello timer. When the holding time expires, the IS purges the adjacency from its database and generates a state change notification.
Packet length	Length, in octets, of this hello packet.
Local Circuit ID (Point-to-point only)	A unique ID assigned to the source interface. It becomes the ID that identifies the circuit to both ends of the link if it is the lower of the two source IDs.
LAN ID (LAN-level only)	Assigned by the designated router (DR), this is the ID of the DR with a one-octet marker to identify the specific LAN.

All types of hello packet headers also contain variable-length fields. They are:

**Table 17-5. Variable-length Hello Packet Header Fields**

Field	Description
Area addresses	The set of area addresses manually configured for this IS, as well as an octet indicating address length.
Padding	Random coding to pad packet length. Values are ignored by the destination IS.
Authentication	Indicates whether authentication is enabled. See for information on authentication. Values are 0 (reserved), 1 (cleartext password), 2254 (reserved), and 255 (routing domain private authentication method).
IS neighbors (LAN-level only)	A current set of routers on the LAN that have announced themselves via hello packets.

#### Link State PDUs (LSPs)

IS-IS uses link-state packets (LSPs) to exchange link state information, that is to transmit routing updates and synchronize routing tables. LSPs are generated and broadcast either when a link state changes or at a user-configurable interval. Each LSP has a unique sequence ID number (LSP ID), generated by the sending IS. The IS increments the number it assigns by 1 after each LSP is sent, regardless of whether the information in the LSP has changed. The higher the sequence number, the more current the LSP.

The learned information (routes, adjacencies, etc.) combines to form, for each intermediate system, an IS-IS database made of reachable addresses for the router to use as a basis for forwarding decisions. LSPs have a maximum buffer size, which applies network-wide (1498 bytes). If there is too much information to fit into a single LSP, then multiple fragments are created and each is sent as a separate LSP. They are reconstructed by the receiving system based on the source ID of the LSP, which is the same for all fragments, and an octet for a specific fragment number.

It is through the LSP database that routers learn reachability information and forward packets to the appropriate destination address. The database informs Level 1 routers of all other Level 1 routers in their area as well as the Level 2 router used for traffic with destinations outside the area. It informs Level 2 routers of whether an address is internal to the area, and therefore can be reached via Level 1 routing. Or, with an external route, which Level 2 adjacency is appropriate.

Many of the NX-IS supported IS-IS commands configure LSP controls and display LSP data. The following aspects of LSP transmission can be configured:

- Password authentication for Level 2 LSPs
- Set the retransmission interval for any or a specific LSP
- Set the minimum and maximum intervals between LSP transmissions



There are two types of LSPs—those transmitted by Level 1 routers and those transmitted by Level 2 routers. A Level 1-2 router transmits both types, depending on the receiver. The packet format for Level 1 and Level 2 LSPs are the same, with the following common fixed header fields:

**Table 17-6. LSP Packet Additional Fixed Header Information**

Field	Description
Packet length	The length of this LSP packet, in octets. This length includes all fixed header information (LSP-specific and those described in <a href="#">Table 17-8 on page 17-11</a> ).
Remaining lifetime	The age of the packet. This is the amount of time a packet is configured to be valid, less the amount of time it has been held by the current system. Also, when a system transmits a packet, it deducts the time it estimates as the trip length to the neighbor. When the value in this field reaches zero, the packet has expired, and the entry is purged from the database.
LSP ID	An ID for the LSP, which is comprised of the following components: <ul style="list-style-type: none"><li>• The system ID of the sending (source) router that generated the LSP</li><li>• Pseudonode ID (for designated routers on a LAN)</li><li>• Fragment number</li></ul>
Sequence number	The packet number, for comparison.
Checksum	The industry-standard checksum, as defined in ISO 8473. The checksum is generated by the source IS and is used to verify routing information from hop to hop. Packets with an incorrect checksum are purged, unless this action is manually overridden.
P (Level 2 routers only)	A flag that indicates that the router supports partition repair. Not supported in the NX-IS software.
Att (Level 1 LSPs generated by Level 1-2 routers)	A four-bit value, one bit of which indicates that the Level-1-2 router is connected to another area.
OL	If set to 1, indicates that the database of the source router is in overload. An LSP with this bit set to 1 is not used in route calculations to other ISs. A value of 0 indicates no overload.

**Table 17-6. LSP Packet Additional Fixed Header Information**

Field	Description
IS type	The type of router. Values are interpreted as follows: <ul style="list-style-type: none"><li>• 1 indicates a Level 1 router</li><li>• 2 indicates a Level 2 router</li><li>• 3 indicates a Level 1-2 router</li></ul>

LSP headers also contain variable-length fields. Several fields—area addresses, authentication, and IS neighbors—are similar to those described in [Table 17-5 on page 17-8](#). Additional fields and field differences are described below:

**Table 17-7. LSP Difference Variable-length Header Fields**

Field	Description
IS neighbors	A current set of adjacent routers on the LAN, and their associated costs and flags.
Endnode neighbors (Level 1 only)	OSI only: Appears once for each cost associated with a link. All neighbors (endnodes) with that cost are listed together. A separate list is included for each cost.
Prefix neighbors (Level 2 only)	OSI only: Appears once for each cost associated with a link. All neighbors (reachable address prefix neighbors) with that cost are listed together. A separate list is included for each cost.

#### Complete and Partial Sequence Number PDUs (CSNP and PSNP)

CSNPs and PSNPs are used to ensure that system databases are synchronized. In a “checks and balance” manner, they are sent out periodically to verify that neighboring routers are working with the same LSPs. While the LSPs contain the database of routes and adjacencies, the sequence number packets contain lists of current LSP ID numbers, sequence numbers, and checksums. These packets are only exchanged between neighbors, not forwarded or broadcast.

Sequence number packets can be either partial (PSNP) or complete (CSNP). A PSNP lists the most recent sequence number of one or more LSPs. It is also used to acknowledge receipt of an LSP (see [“IS-IS Adjacency Establishment and Maintenance” on page 17-14](#)) or to request an LSP due to missing or outdated information.

A CSNP contains a list of every active LSP in the database. CSNPs are exchanged either when a link first comes up, or periodically (the interval is user-configurable), to verify and ensure accurate database routes. Because the number of LSP can be too large to fit into a single CSNP packet, the protocol allows use of an LSP range to specify the appropriate set of LSPs.

There are four types of sequence number PDUs:

- Level 1 Complete Sequence Numbers PDU
- Level 2 Complete Sequence Numbers PDU
- Level 1 Partial Sequence Numbers PDU
- Level 2 Partial Sequence Numbers PDU

All SNPs use the same header fields, with the exception of the address range for CSNPs. The common fixed header fields are described below:

**Table 17-8. SNP Additional Fixed Header Information**

Field	Description
Packet length	The length of this SNP, in octets. This length includes all fixed header information (SNP-specific and those described in <a href="#">Table 17-8 on page 17-11</a> ).
Source ID	The ID of the router that generated the SNP.
Start LSP ID (CSNPs only)	For the address range described, the system ID of the first LSP in the range.
End LSP ID (CSNPs only)	For the address range described, the system ID of the last LSP in the range.

The variable-length fields of an SNP header include the LSP entry and an authentication field (described in [Table 17-5 on page 17-8](#)).

## IS-IS Packet Formats

Each of the packet formats (described in “[IS-IS Packet Types](#)” on page 17-6) uses the same IS-IS fixed header information, which is illustrated in [Figure 17-2](#). All fields are one octet (8 bits).

Network-layer protocol identifier (NLPID)
Header length
Version
ID length
Packet Type
Version
Reserved
Maximum area address

**Figure 17-2. IS-IS Fixed Header Fields**

The following table describes the fixed header fields.

**Table 17-9. Description of Fields in the IS-IS Fixed Header**

Field	Description
Protocol identifier	Intradomain routing protocol discriminator.
Header length	Length of fixed header, in octets.
Version	Version/protocol ID extension. Always set to 1.
ID length	Length of the ID field of the NSAP addresses. Values are as follows: <ul style="list-style-type: none"><li>Between 1 and 8, inclusive, indicates an ID field of the corresponding length. The NX64000 system only supports an ID field of 6 octets.</li><li>Zero indicates a 6-octet ID field length</li><li>255 indicates a 0-length ID field</li></ul>

**Table 17-9. Description of Fields in the IS-IS Fixed Header**

Field	Description
Packet Type	One of the following PDU types: <ul style="list-style-type: none"><li>• Level 1 LAN IS-to-IS hello packet</li><li>• Level 2 LAN IS-to-IS hello packet</li><li>• Point-to-Point IS-to-IS hello packet</li><li>• Level 1 link state packet (LSP)</li><li>• Level 2 LSP</li><li>• Level 1 complete sequence numbers packet (CSNP)</li><li>• Level 2 CSNP</li><li>• Level 1 partial sequence number packet (PSNP)</li><li>• Level 2 PSNP</li></ul>
Version	Version/protocol ID extension. Always set to 1.
Reserved	Set to 0, and therefore, ignored.
Maximum area address	Number of area addresses supported. A value of 0 indicates three area addresses, to be compatible with older versions of the protocol that supported only up to three. The NX64000 system only supports three.

With Integrated IS-IS, each packet type has additional common fixed header fields, although the values change depending on the packet type. See RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*, for a complete description of the coding for each type.

## IS-IS Address Structure

IS-IS requires use of a Network Service Access Point (NSAP) and network entity title (NET). These addresses, which conform to the requirements of ISO 8348, identify the area and system ID for the router. When IS-IS is used as IGP for IP networks, there is but a single difference between a NET and an NSAP—the selector byte at the end. This byte is not used by IS-IS in IP networks, and is always set to zero. So, an NSAP with a zero selector byte is an IS-IS NET address. (In the original CLNS network, the selector byte identified transport layer protocols—21 for OSI transport, 20 for Digital's proprietary NSP protocol.)

Because the NET address can be a long string of not easily recalled numerics, IS-IS provides an option to use CLNS names. These names allow you to map text names to the NET address. You can then use the name in place of the alphanumeric address in any operation that requires entry of the NET. Often, a router displays the CLNS name in place of the NET address in its output displays.

Because IS-IS tends to be constrained within a single ISP, you can choose the method used to assign a unique NSAP to your router. Links between ISPs use BGP, which knows nothing of IS-IS's NSAPs, so that address is unimportant.

An IS-IS NSAP address is divided into three parts: an area address, a system ID, and an NSAP selector. A router can support more than one NET; each one identifies an area to which the router belongs. The NET can be 8 to 20 bytes long, and is laid out as follows:

Area address	ID	Selector
--------------	----	----------

The following table describes these components:

**Table 17-10. NET Address Components**

Component	Description
Area address	The area address is used for routing between areas (Level 2 routing), can be from 1 to 13 bytes long.
System ID	The system ID is used for routing within an area. For Level 1 routing the system ID must be unique within the area, for Level 2 routing it must be unique across the backbone. System ID is 6 bytes long.
NSAP selector	The N-selector is one byte, and always set to 00 for IP routing in IS-IS.

## IS-IS Adjacency Establishment and Maintenance

IS-IS uses hello packets as keepalives to verify the connection with an adjacency, and uses LSPs to transmit routing database information. CSNPs and PSNPs synchronize databases by verifying that each IS is working with the most current, and correct LSP information.

### Routing Processes

IS-IS uses four routing processes. They are described in the following table:

**Table 17-11. IS-IS Routing Processes**

Process	Description
Decision Process	Responsible for calculating the shortest path to each destination in the domain based on the information in the link-state database. The process is run separately both for each routing level, and within each level for each supported extended metric. The Decision Process is also responsible for adjacency formation.

**Table 17-11. IS-IS Routing Processes**

Process	Description
Update Process	Responsible for LSP activity—origination and propagation. Information necessary for these operations is comes from LSPs and SNPs generated by the Receive Process.
Forwarding Process	Responsible for managing the buffers for packet forwarding. It forwards based on information in a specific Forwarding Database, which is determined by the setting metrics and QoS parameters.
Receive Process	Receives input and is responsible for taking the appropriate action (e.g., passing information to the Forwarding or Decision Process).

### Link-State Database Maintenance

When an IS receives an LSP, it checks whether this LSP is already in its link state database. If it finds any LSPs lacking, it does one of the following:

- If it is missing, the LSP is installed in the database.
- If it is in the database, it compares the sequence numbers. The LSP with the higher number is the more current.

When an IS on a point-to-point network receives an LSP, it responds with an acknowledgement in the form of an PSNP. If the sending IS-IS does not receive an acknowledgement in the configured amount of time, it resends the LSP. An IS on a broadcast network does not acknowledge receipt of an LSP. The DR on the LAN sends out CSNPs to all, and each IS compares its list of LSPs (identified by their ID and sequence number) with the list broadcast by the DR. If the IS finds its own list incorrect, it broadcasts a PSNP to the LAN for the missing LSP ID numbers. It is the job of the DR to respond to that broadcast. If the receiving IS finds that its own information is more current, it transmits an LSP with the correct information to the DR and all other ISs on the LAN. The sequence number having incremented, the receiving systems will then install that information in their databases.

### Avoiding Overload Problems

If memory becomes depleted on the router so that the system cannot reliably compute routes, it sets the Overload Bit in its LSPs. This prevents other routers from using the overloaded system as a transit router, although they still send LSPs. The overloaded router continues to update its tables. You can manually set this bit for a router as well. Because the router continues to update its database, manually setting the bit allows you to bring a router online and allow it to develop complete routing tables before relying on it for routing decisions.

### Route Selection Process

IS-IS uses both best route selection and default routes to make forwarding decisions. Both processes are described below.

## Influencing Interface Cost

Metrics influence route preference by associating a cost with a route or adjacency. This cost is used by the SPF algorithm to calculate the best path to a destination. You can assign a cost to an interface, or assign a default cost to all routes being redistributed into IS-IS. However, if the redistributed route has a cost specifically assigned to it, you cannot override that cost.

## Extended Metrics

To further influence the decision process, IS-IS supports extended metrics. Extended metrics extend the IS-IS “TLV” to 32 bits, allowing more room for IS-IS information exchange between routers. Although optional to the protocol, support of extended metrics is necessary in some cases for interoperability, as well as for MPLS operations. Configuration for extended metrics determines the number of bytes set aside for the metrics. Within these bytes are held the characteristics of a particular link.

Basically, the narrow IS-IS metrics let you tell other IS-IS routers a specific amount of information (one byte). With extended metrics, there is additional number of bytes (three bytes for IS-IS adjacencies, four bytes for IP routes). Those extra bytes contain information such as traffic engineering settings, in addition to the traditional interface cost route metric. These metric values are recorded in the LSPs.

## Preferred Routes

IS-IS selects a route in the following order:

- Routes within the area (level 1 routing)
- Routes within the routing domain using internal metrics
- Routes outside of the routing domain

## For Level 1 Routers

Within level 1 routing, if a destination matches more than one entry in the link-state database, IS-IS uses “best match” routing. That is, if there are more than one reachability entries in the database, IS-IS chooses the one with the mask containing the most “1” bits. If there are more than one equally specific addresses, the route that supports the requested type of service is selected. If more than one match still exists, the shortest path is preferred. Shortest path selections, in case of a tie, can be determined by the use of metrics. If metrics do not choose a route, the protocol can use load splitting.

When a level 1 router has to select a level 2 router to forward out-of-area packets to, it first looks for a router that supports the requested TOS, and secondly relies on shortest path.

## For Level 2 Routers

Routes learned from a level 1 router are always preferred, as they indicate internal routing. For level 2 routers that also function as level 1 routers, routes learned via their level 1 routing is always preferable to routes learned from any level 2 source. If there is no level 1 input, first choice is a route using only internal metrics. If there is more than one qualifying route, the decision process is the same as that described for level 1 routers (such as best match TOS support, shortest path).



If only a route with external metrics is available, IS-IS still evaluates in the order of best match, TOS support (from external route to the border router with the internal route), shortest path. In this case, shortest path is determined by the smaller external metric, then the smaller internal metric.

## Default Routes

You can configure an IS to announce and redistribute a default route into the IS-IS routing domain. It is used as the “route of last resort,” meaning that if a router does not have a more specific route to a destination IP address, it uses the default route.

For example, you may have a network with only one of the routers in that network connected to the Internet. Rather than announcing every route on the Internet to all the routers in your local network, you could configure the Internet-connected router to announce a default route to all the other routers in your network. Any router in your network could route packets to other nodes in the local network (since it presumably knows about all the subnets in the network from the IGP). But, when a user tries to connect to a node in the Internet, the local router would not have a specific route to this address. In that case, it would use the default route, and the packets would be routed back to the Internet-connected router. The default route is announced as 0.0.0.0 with a mask of 0.0.0.0. (any IP address matches it).

## Authentication

IS-IS supports an authentication option to control router participation and area membership. There are two types of authentication in IS-IS—authentication of hello packets and authentication of LSPs. The following sections describe these.

### Router-based Authentication

Router-based authentication is done by adding password requirements to LSPs. There are two types of authentication supported. Both include the password in the LSP header, and the receiving router checks that password to verify the adjacency. The methods are as follows:

- Simple authentication, which is a straight text-string password that is included in the LSP header. It is less secure than the MD5 option.
- MD5 authentication, which uses an encrypted string to verify the packet.

When configuring router-based authentication, you can specify whether it applies to an area (Level 1 routing) or a domain (Level 2 routing).

### Interface-based Authentication

For more fine-grained authentication, you can add password requirements to hello packets. This sets interface-based authentication. Each interface forming an adjacency must be configured with the same authentication setting and password, or packets are dropped. You can use show commands to display discarded traffic on a port-by-port basis. When an interface does not see a password from the other end, it drops the hello packets. When the configured hello time expires, the adjacency is dropped. When configuring hello-based authentication, you can specify whether to authenticate on Level 1, Level 2, or both types of adjacencies.

## Designated Routers/Pseudonodes

You can configure priorities for a router, and this value is used to elect a designated router. You can configure both a Level 1 and Level 2 priority, as each level elects its own DR. The router with the highest priority becomes DR. In the case of a tie, the router with the highest MAC address (system ID) becomes DR. If the current DR is joined online by a router with a higher priority, the higher priority router becomes the DR. If the current DR fails, a new election chooses the next DR. If a router is configured with a priority of zero, it is ineligible.

Normally, each router advertises its link to each other router that it can communicate with. However, for router with a large number of links, like those on a broadcast LAN, this can cause large amounts of traffic. To avoid the massive congestion of each IS constantly sending, responding, and resending LSPs to every other IS, IS-IS uses a pseudonode in conjunction with a designated router (DR). Each IS reports links to the pseudonode, instead of to every other IS, and the DR sends out LSPs representing the pseudonode to all the other ISs in the network.

## IS-IS Configuration

The following table lists all the IS-IS commands supported by the NX64000 switch/router. Each command listing indicates the general functions for which you would use the command. Some commands are included and described in the configuration examples that follow. All commands listed are documented in the *NX64000 Command Reference*.

**Table 17-12. IS-IS Command Usage**

Command	Basic Configuration	Authentication	Routing Decision Process	PDU Manipulation	Default Support
area-password		✓			
clns host	✓				
clns router isis					✓
clns routing					✓
default-information originate			✓		
default-metric			✓		
domain-password		✓			
ignore-lsp-errors				✓	

**Table 17-12. IS-IS Command Usage**

Command	Basic Configuration	Authentication	Routing Decision Process	PDU Manipulation	Default Support
ip router isis	✓				
isis circuit-type	✓				
isis csnp-interval				✓	
isis hello-interval				✓	
isis hello-multiplier				✓	
isis mesh-group				✓	
isis metric			✓		
isis password		✓			
isis priority			✓		
isis retransmit-interval				✓	
isis retransmit-throttle-interval				✓	
is-type	✓				
log-adjacency-changes				✓	
lsp-gen-interval				✓	
lsp-refresh-interval				✓	
metric-type			✓		
net	✓				
passive-interface				✓	
redistribute			✓		
router isis	✓				

**Table 17-12. IS-IS Command Usage**

Command	Basic Configuration	Authentication	Routing Decision Process	PDU Manipulation	Default Support
set-overload-bit				✓	
show clns	✓			✓	
show clns host	✓				
show clns interface	✓	✓	✓	✓	
show clns is-neighbors	✓				
show clns neighbors	✓				
show clns protocol	✓		✓		
show clns route	✓		✓		
show clns traffic			✓	✓	
show isis database	✓		✓	✓	
show isis mesh-groups				✓	
show isis spf-log			✓		
spf-interval			✓		
summary-address			✓		
tunnel destination				✓	
update-queue-depth					✓

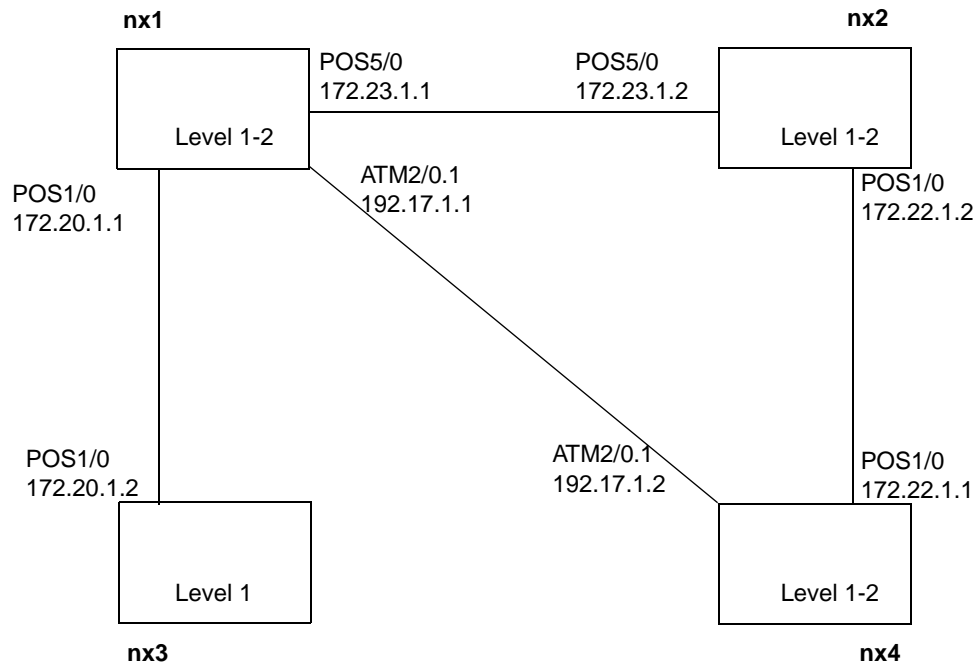
## Basic IS-IS Configuration Tasks

The following sections illustrate some basic IS-IS configuration tasks. Refer to the first part of this chapter for an explanation of the protocol operation.

In general, to configure IS-IS on an NX64000 router, you must minimally configure the following:

- Load the protocol
- Define the router's area address and system ID, which will determine whether the routers form level 1 or level 2 adjacencies
- Enable the protocol on the router and on each configured interface

The figure below illustrates the IS-IS network as configured in this guide:



**Figure 17-3. IS-IS Configuration Example**

## Loading and Enabling IS-IS

After configuring interfaces (for information about configuring interfaces see [Chapter 7, “Cards and Interfaces”](#)), load and enable IS-IS on the router. To complete these steps, you must load the protocol, enter IS-IS router configuration mode, and assign a NET.

NX1:

```

nx1# configure terminal
nx1(config)# load isi
nx1(config)# router isis
nx1(config-router)# net 49.0001.0000.0000.0001.00

```

NX2:

```

nx2# configure terminal
nx2(config)# load isi
nx2(config)# router isis
nx2(config-router)# net 49.0001.0000.0000.0002.00

```

NX3:

```
nx3# configure terminal
nx3(config)# load isi
nx3(config)# router isis
nx3(config-router)# net 49.0001.0000.0000.0003.00
```

NX4:

```
nx4# configure terminal
nx4(config)# load isi
nx4(config)# router isis
nx4(config-router)# net 49.0001.0000.0000.0004.00
```

The following tables explains the commands in the preceding example:

Configuration Line	Description
<code>load isi</code>	Loads IS-IS from software release 1.7.0
<code>router isis</code>	Enables IS-IS on the router and enters IS-IS router configuration mode.
<code>net 49.0001.0000.0000.0004.00</code>	Assigns network entity title to the router. The area ID portion of the NET, 49.0001 in this case, define the area that the router belongs to. NX1, NX2, NX3, and NX4 all belong to the same area. The system ID, 0000.0000.0004 in this case, must be unique within the area. The n-selector must always be 00.

## Enabling IS-IS on the Interface

Once you have enabled IS-IS on the router and assigned a NET, you must enable it on the specific interfaces on which it will run. The following example only shows enabling IS-IS on NX1 interfaces, but the process is the same for each router, specific to the interfaces configured.

- The `ip router isis` command provides optional arguments which are only valid for configuring MPLS tunnel interfaces. For more information on this type of interface, see [Chapter 14, “Multi-protocol Label Switching \(MPLS\) Configuration.”](#)

NX1:

```
nx1# configure terminal
nx1(config)# interface pos1/0
nx1(config-if)# ip router isis
nx1(config-if)# exit
nx1(config)# interface pos5/0
nx1(config-if)# ip router isis
nx1(config-if)# exit
nx1(config)# interface atm2/0.1
nx1(config-if)# ip router isis
nx1(config-if)# exit
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<b>interface interface-name</b> <b>pos1/0</b> <b>pos5/0</b> <b>atm2/0.1</b>	Enters interface configuration mode for the interface specified. This example uses a previously configured interface (otherwise you would have to specify an encapsulation mode and IP address).
<b>ip router isis</b>	Enables IS-IS on the interface.
<b>exit</b>	Returns you to the previous prompt level.

## Configuring the Routing and Circuit Levels

You can assign each router to a particular type of routing, either level 1 only for small networks, or level 1-2 to add hierarchy. Level 1-2 is the default. You can also set the circuit type for each interface, which controls the type of adjacencies the interface can form. Options for circuit type are level 1-2, level 1 only, and level 2 only. By default a circuit is type level 1-2.

The following example sets NX3 to a level 1 only router. Interfacepos1/0 is then set to level 1 only, since it is a level 1 only router. NX4 is a level 1-2, but it's POS interface is set to level 1 only. NX1 and NX2 use the default router type (level 1-2) and default circuit type (level 1-2), so they do not require any additional configuration. NX1 forms a level 1 adjacency with NX3, and NX2 forms a level 1 adjacency with NX4. NX1 and NX2 form a level 1 adjacency with each other, but Level 2 with any router outside of the area. NX1 and NX4 form a level 2 adjacency over the ATM link.

NX3:

```
nx3(config)# router isis
nx3(config-router)# is-type level-1
nx3(config-router)# exit
nx3(config)# interface pos1/0
nx3(config-if)# isis circuit-type level-1
```

NX4:

```
nx4(config)# interface pos1/0
nx4(config-if)# isis circuit-type level-1
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<b>router isis</b>	Enters IS-IS router configuration mode. (IS-IS was enabled in a previous step.)
<b>is-type level-1</b>	Sets the router to a level 1 only router, allowing it to communicate only with other level 1 only or level 1-2 routers within the area.
<b>interface pos1/0</b>	Enters interface configuration mode for the interface pos1/0.
<b>isis circuit-type level-1</b>	Configures the interface so that it can only form level 1 adjacencies.
<b>exit</b>	Returns you to the previous prompt level.

## Configuring Authentication

IS-IS authentication can be enabled on both the area level and the interface level. The following examples set passwords for each.

### Configuring LSP-based Passwords

Area-level passwords configure verification to the router's Level 1 LSPs. Domain-level passwords configure verification to the router's Level 2 LSPs. When you enable an LSP-based password—for either Level 1, level 2, or both—the router only accepts LSPs that include that password. Enabling area and domain passwords is done from the router configuration prompt.

NX1:

```
nx1(config)# router isis
nx1(config-router)# area-password "see you later alligator" md5-pwd
nx1(config-router)# domain-password "in a while crocodile" md5-pwd
```

NX2:

```
nx2(config)# router isis
nx2(config-router)# area-password "see you later alligator" md5-pwd
nx2(config-router)# domain-password "in a while crocodile" md5-pwd
```

NX3:

```
nx3(config)# router isis
nx3(config-router)# area-password "see you later alligator" md5-pwd
nx3(config-router)# domain-password "in a while crocodile" md5-pwd
```

NX4:

```
nx4(config)# router isis
nx4(config-router)# area-password "see you later alligator" md5-pwd
nx4(config-router)# domain-password "in a while crocodile" md5-pwd
```



The following table explains the commands in the preceding example:

Configuration Line	Description
<code>router isis</code>	Enters IS-IS router configuration mode. (IS-IS was enabled in a previous step.)
<code>area-password "see you later alligator" md5-pwd</code>	Requires any received Level 1 LSP to include the password <code>see you later alligator</code> . Authentication type is MD5, which is a more secure, encrypted format. Each router in this configuration is configured with the password to ensure full database synchronization.
<code>domain-password "in a while crocodile" md5-pwd</code>	Requires any received Level 2 LSP to include the password <code>in a while crocodile</code> . Authentication type is MD5, which is a more secure, encrypted format. Each router in this configuration is configured with the password to ensure full database synchronization.

## Configuring Interface-level Passwords

Interface-level passwords configure verification to the router's hello packets. The same password must be set for each end of an adjacency. If an interface is not configured with a matching password, it cannot accept the hello packet, it therefore does not respond, and the adjacency is dropped. Enabling interface-based password authentication is done from the interface configuration prompt. Passwords are set separately for each configured interface and each level.

NX1:

```
nx1(config)# interface pos1/0
nx1(config-if)# isis password lionsandtigers
nx1(config-if)# isis password bearsohmy level-2
nx1(config-if)# exit
nx1(config)# interface pos5/0
nx1(config-if)# isis password aroundtheworld
nx1(config)# interface atm2/0.1
nx1(config-if)# isis password level1pass
nx1(config-if)# isis password level2pass level-2
```

NX2:

```
nx2(config)# interface pos1/0
nx2(config-if)# isis password lionsandtigers
nx2(config-if)# isis password bearsohmy level-2
nx2(config-if)# exit
nx2(config)# interface pos5/0
nx2(config-if)# isis password aroundtheworld
```

NX3:

```
nx3(config)# interface pos1/0
nx3(config-if)# isis password lionsandtigers
```

NX4:

```
nx4(config)# interface pos1/0
nx4(config-if)# isis password lionsandtigers
nx4(config)# interface atm2/0.1
nx4(config-if)# isis password level1pass
nx4(config-if)# isis password level2pass level-2
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<b>interface interface-name</b> <b>pos1/0</b> <b>pos5/0</b> <b>atm2/0.1</b>	Enters interface configuration mode for the interface specified.
<b>isis password password</b> <b>lionsandtigers</b> <b>level1pass</b>	Requires any received level 1 hello packets to include the configured password. (Level 1 is the default.) All POS1/0 interfaces are configured with the same password so that they can maintain their adjacency. NX1 and NX2 also include a hello-based password statement for interface POS5/0.
<b>isis password password level-2</b> <b>bearsohmy</b> <b>level2pass</b>	Requires any received level 2 hello packets to include the configured password. Level 2 is specified in the command. Because interfaces POS1/0 on NX3 and NX4 are level 1 only routers, their configuration does not include a level 2 password.

## Manipulating the Routing Decision Process

There are several ways to influence the routing decision process, whether by changing metrics or by manipulating the routing table itself. Examples of metrics, route redistribution, aggregating addresses, and configuring passive interfaces are described in the following sections.

### Changing the Default Metric

By default, an interface has a default metric of 10. That is, the cost associated with using a particular interface is 10. To influence the direction of traffic, set the metric on an interface to a higher or lower value. Higher values are less preferred. Metrics are set independently for each routing level. Level 1 is the default.

In the following example, the ATM interface on NX1 and NX4 is being saved for interarea (level 2), not level 1 traffic. The metric for level 1 routing is set to be higher than the cost of traveling through NX2 (metric of 10 plus 10), so that packets from NX4 will not use interface ATM2/0.1 for level 1 traffic.

NX1:

```
nx1(config)# interface atm2/0.1
nx1(config-if)# isis metric 35
```

NX4:

```
nx4(config)# interface atm2/0.1
nx4(config-if)# isis metric 35
```

## Route Redistribution

Redistribution alters the routing scheme to allow routes from another protocol into the IS-IS routing table. For example, you could redistribute static routes into IS-IS:

NX1:

```
nx1(config)# router isis
nx1(config-router)# redistribute static
```

## Using Summary Addresses

A summary address allows you to aggregate addresses on level 2 routers, thereby minimizing the routing table size and the number of routing updates. Route aggregation allows you to advertise many routes with a single route table entry. When configured for route aggregation, the system recognizes a less-specific match as the routing table entry for multiple more-specific routes. In the following example, NX1 is configured to summarize all addresses in the 172.0.0.0 range.

NX1:

```
nx1(config)# router isis
nx1(config-router)# summary-address 172.0.0.0 255.0.0.0
```

## Blocking Adjacency Formation

You can set an IS-IS interface to learn routes but not to advertise itself. This could be useful for testing or isolation purposes. By making an interface passive in this way, you allow external networks attached to that interface to be advertised into IS-IS, without that interface actively participating. The passive interface continues to send out LSPs, which would include its learned routes but does not include its own IP address.

For example, you may want to set a loopback interface on an area border router between IS-IS and OSPF to advertise the networks it has learned about through OSPF into the IS-IS network, but not want the loopback itself advertised:

NX1:

```
nx1(config)# router isis
nx1(config-router)# passive-interface loopback0
```

## Verifying IS-IS Configuration

The NX-IS software provides a number of **show** commands that let you verify your IS-IS configuration. The following table lists the type of information you can view from each show command:

**Table 17-13. IS-IS Commands for Verifying Configuration**

Action	Command
Display global information, including: <ul style="list-style-type: none"><li>• number of interfaces configured</li><li>• NET</li><li>• timer information</li><li>• remaining packet lifetime</li><li>• routing level setting</li><li>• area ID</li></ul>	<b>show clns</b>
Display configured name mappings for the local router (set with the <b>clns host</b> command).	<b>show clns host</b>
Display all relevant information for the specified interface, including: <ul style="list-style-type: none"><li>• circuit type and status</li><li>• metrics, priority setting, ID, and active adjacencies for level 1 and level 2</li><li>• hello and retransmit intervals and hello multiplier</li><li>• password configuration</li></ul>	<b>show clns interface</b>
Display detailed information about IS-IS adjacencies, including: <ul style="list-style-type: none"><li>• system ID</li><li>• interface name and state</li><li>• circuit type and ID</li><li>• priority setting</li><li>• area address and uptime</li></ul>	<b>show clns is-neighbors</b>

**Table 17-13. IS-IS Commands for Verifying Configuration**

Action	Command
Display information about all neighbors, including: <ul style="list-style-type: none"> <li>• system ID</li> <li>• subnetwork point of attachment</li> <li>• name of the interface over which the system ID was learned</li> <li>• state of the connection</li> <li>• time left before the adjacency times out</li> <li>• circuit type</li> <li>• protocol used to learn of the adjacency</li> <li>• area address, IP address, and uptime</li> </ul>	<code>show clns neighbors</code>
Display router-specific information about IS-IS, including: <ul style="list-style-type: none"> <li>• system ID</li> <li>• level of routing</li> <li>• manually configured and learned area addresses of router's adjacencies</li> <li>• IS-IS enabled interfaces</li> <li>• aggregate addresses</li> </ul>	<code>show clns protocol</code>
Display, for each interface and at each level, sent/received counts and sourced/flooded counts for: <ul style="list-style-type: none"> <li>• hellos</li> <li>• LSPs</li> <li>• CSNPs</li> <li>• PSNPs</li> </ul>	<code>show clns traffic</code>
Display the contents of the IS-IS database, including: <ul style="list-style-type: none"> <li>• LSP ID, sequence number, checksum, and hold time</li> <li>• Attach bit, P bit, and overload bit settings</li> <li>• area and IP addresses that sent the LSP</li> </ul>	<code>show isis database</code>
Display members of a mesh group.	<code>show isis mesh-groups</code>
Display history of SPF calculations, which includes: <ul style="list-style-type: none"> <li>• time of calculation occurrence and duration</li> <li>• nodes affected</li> <li>• LSP ID that triggered the calculation and reason</li> </ul>	<code>show isis spf-log</code>



## BGP Configuration

The Border Gateway Protocol (BGP) has become the protocol of choice for exchanging information between autonomous systems (ASs). It is the standard exterior gateway protocol, commonly accepted as the replacement to the now-defunct Exterior Gateways' Protocol (EGP). BGP bridges the gap between networks running different interior gateway protocols (for example, OSPF and IS-IS).

BGP is an inter-autonomous system routing protocol that exchanges reachability information with other BGP systems. A BGP speaker (a BGP router that sends messages) advertises its routing table to peers (adjacent BGP routers) in neighboring ASs. Because BGP uses this “hop-by-hop” paradigm, it is perfectly suited for providing the routing mechanism for the Internet. In addition, as it verifies and consolidates the routing information it receives, it ensures that only the optimal path to a network is installed and, therefore, advertised. Because “optimal” is relative to a speaker's view and place in the network, each speaker continually makes installment decisions and updates its tables accordingly.

### Key Features

The BGP implementation in NX-IS supports:

- Control reachability information sent between ASs
- Redistribute packets to the interior gateway protocol (IGP) within an AS
- Redistribute connected routes, IGP routes, and static routes into BGP
- Maintain a large database of routing information (RIB)
- “Dampen” routes which are not stable to provide an overall stable RIB
- Break your network into manageable pieces and provide a scalable hierarchy.

## Technology Concepts

Basic to understanding of BGP are:

- Autonomous systems
- External BGP
- Internal BGP
- BGP route decision and installation process
- Routing information flow
  - Peer groups
  - Route reflectors
  - Confederation
  - Aggregation
  - Redistribution
  - Route filtering

## Standards and RFCs

This implementation of BGP is based on the following Request For Comments (RFCs), enabling standardized BGP on the systems:

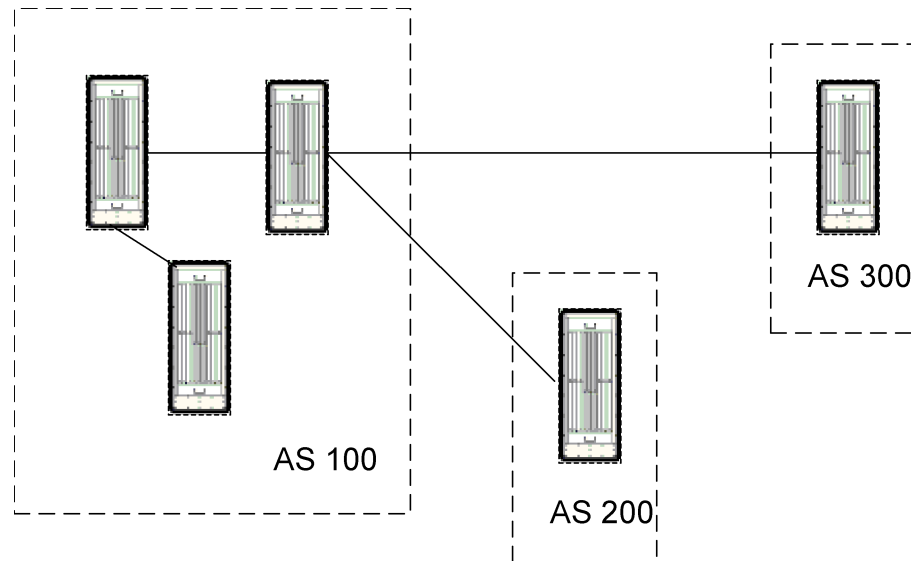
**Table 18-1. RFCs Implementing Standardized BGP**

RFC	Title
RFC 1771	A Border Gateway Protocol 4 (BGP-4)
RFC 1966	BGP Route Reflection An alternative to full mesh IBGP
RFC 1997	BGP Communities Attribute
RFC 2439	BGP Route Flap Damping

## Autonomous Systems

An AS refers to a set of routers under a single administration. The AS uses one, and sometimes more than one IGP to route packets within the AS, and/or an exterior gateway protocol (EGP) to route packets to other ASs. An AS can have as few as one router and, theoretically, has no upper limit on the number of routers it contains. Admittance to the AS is determined by the ability to exchange routing information and to maintain connectivity. The following figure illustrates three ASs that internetwork:





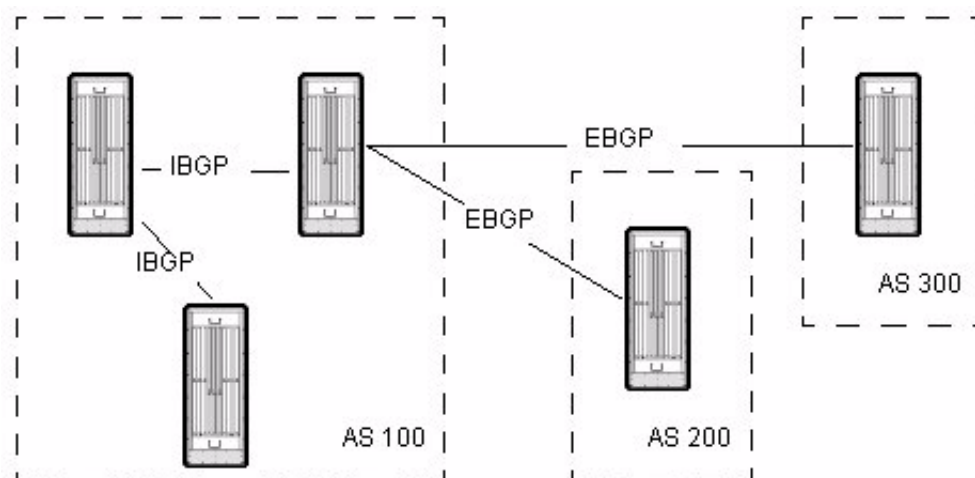
**Figure 18-1. Autonomous Systems in an Internetwork**

#### External BGP

Routers that are in different ASs, and that exchange BGP updates, are running External Border Gateway Protocol (EBGP). BGP updates allow these routers to maintain a consistent view of the network topology. The routers on the edge of the AS, the gateway routers, represent their AS to other gateway routers in other ASs. Each functions as an entry point to its AS. See [Figure 18-2](#) for an illustration of external BGP.

#### Internal BGP

Routers that are within the same AS, and that exchange BGP updates are running Internal Border Gateway Protocol (IBGP). Through BGP you can assign which router peers with external ASs. That router will pass the network information to allow all routers within the AS to share a single network-wide view of the topology.



**Figure 18-2. External vs. Internal BGP**

## BGP Route Decision and Installation Process

BGP uses TCP as its transport protocol. Any two routers that have an active TCP session for the purpose of exchanging BGP updates are “peers” or “neighbors.” Before a BGP router advertises its routing table with a peer in an external AS, it verifies reachability information with peers within the AS. In either case, BGP peers initially exchange their full routing tables, but subsequently only send updates as needed. Because BGP information is exchanged only on change, it minimizes bandwidth requirements.

When a BGP speaker receives a route from a peer, it first determines if it is a known or unknown destination. If it is unknown, the speaker inserts the route into its routing table and advertises the information to its peers. If it is known, the speaker determines whether the route to the destination is preferable to the route currently stored in the table. If it is preferable, the old route is replaced and the new route is advertised to all peers. If it is not a better route, the update is dropped.

- The operations documented in this section describe the general process of route installation into the RIB. In practice on the NX64000 system, a received route must first pass the decision criteria defined by the applied routing policies. See the “Route Filtering Configuration” chapter for information on policies.

## Update Messages

BGP uses updates to form a picture of the network (see “[Message Types](#)” on [page 18-9](#) for more information on BGP packet types and message headers.) BGP peers exchange updates to discern network reachability. Each update lists information on a valid path—both hops and attributes—as well as lists routes that were attached to the path but are no longer reachable.

After establishing a connection, each router sends a series of update messages to inform the new peer of all routes reachable through the router. When an update is received, each router compares any listed paths with the paths in its current routing table. If the new path is preferable to the path in the routing table, the router replaces the route in its table and sends its neighbors an update message.

## Neighbor Statements

Routers in different ASs that use BGP to communicate and exchange information are called neighbors. Neighbors that are within the same AS, internal neighbors, need not be directly connected. If they are not, however, they must be fully meshed, part of a confederation, or using route reflection. (See [“Route Reflectors” on page 18-13](#) and [“Confederation” on page 18-14](#) for more information.) External neighbors, those that are in a different AS, must be directly connected or use the `neighbor ebgp-multihop` command to allow peering.

Using neighbor statements, you not only establish neighbor relationships, but assign configuration characteristics. These characteristics can be applied to a specific neighbor identified by its IP address or a group of neighbors—a peer group. They allow you to more carefully define the nature of the relationship, such as:

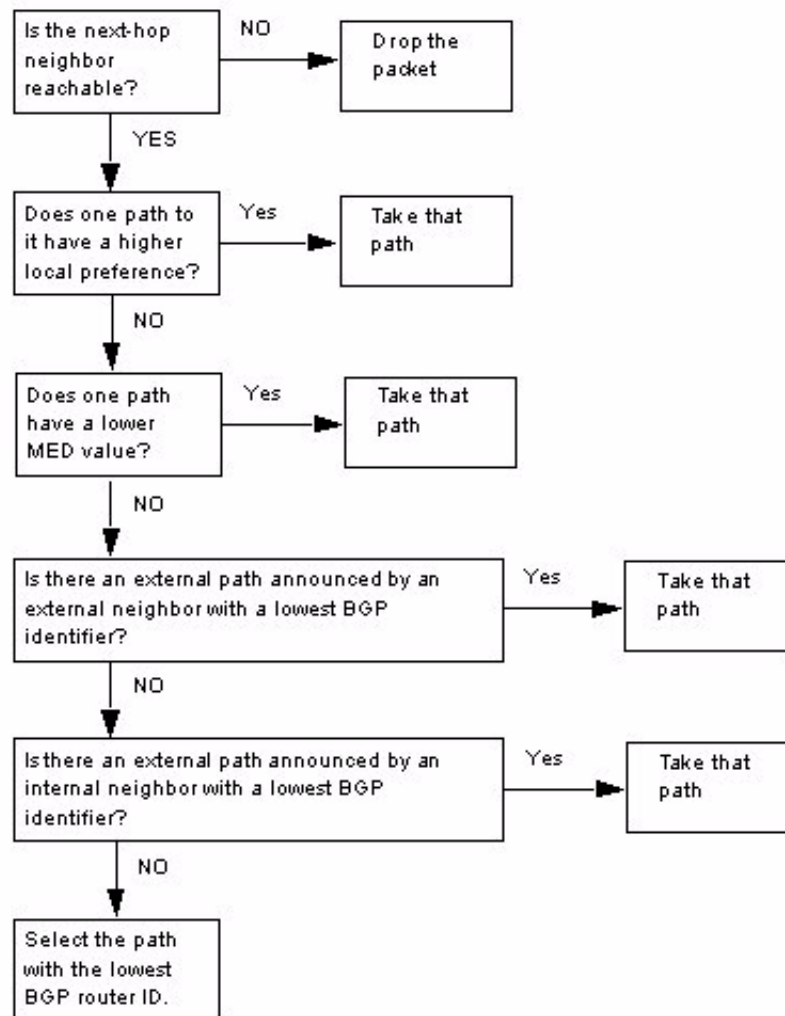
- define or override attributes of a neighbor
- allow or disallow certain types of connections
- set advertisement and update intervals

## Path Selection

BGP collects path and network information from its neighbors via update messages. Because BGP stores path attribute information with each route, BGP can detect and avoid routing loops and choose the best path to a destination. That choice is made based on the criteria deemed most important by the network administrator.

Path information is comprised of two parts—the AS path and the path attributes. The AS path is a list derived from neighbor updates that describes the route to a destination. Path attributes, discussed in more detail in the next section, are the additional pieces of route information that complete the topology picture allowing the router to make the best path choice.

When BGP learns more than one route to a destination, it must figure out which route to submit to the route manager. To determine the preferred route, BGP follows the decision making process below:



**Figure 18-3. BGP Path Selection Process**

Before routing packets on a newly received route, each router performs two checks of the route:

1. The router checks that the local AS is not part of the path, otherwise a loop will result.
2. The router checks that the route maintains a stable state for a period of time before declaring it valid.

The path selection process is run each time a router updates a path in its routing table. When a router receives an update instructing it to remove a path, it does so and replaces the path with a new path determined by the above process.

## Path Attributes

Path attributes serve as “tie-breakers” when more than one path is available to a destination. Some attributes are transitive—that is, they apply throughout a network; others only make sense locally. The following are BGP well-known attributes, that is, attributes that all BGP implementations must recognize.

### AS Path Attribute

The AS path is a list of each AS traversed in getting to the local router. BGP prepends each AS to the beginning of the list so that the further down the list the AS, the further away the router. The AS path list is critical in preventing routing loops. It is possible to override this attribute, for purposes of creating export route filters. In this case, the AS path of a route matching the filter criteria is replaced with the path you specify.

There are two types of AS path segments—sets and sequences. Sets are an unordered group of ASs. Sequences are an ordered group.

### Origin Attribute

The origin attribute helps describes the route’s source or origin. It can be one of three values—IGP, EGP, or incomplete. The table below describes the origin settings:

**Table 18-2. BGP Origin Attributes**

Value	Description
IGP	The route originated within the AS (was learned from the Interior Gateway Protocol).
EGP	The route originated outside of the local AS (was learned from the Exterior Gateway Protocol).
Incomplete	The route’s origin is unknown or undetermined (the route was redistributed into BGP).

### Next-hop Attribute

The next-hop attribute is simply the IP address of the next gateway that the local router will use to reach a certain destination. Two requirements of BGP for next-hop are:

1. A router cannot advertise to a peer a route that it originates and in which that peer is the next hop.
2. A router cannot accept route in its routing table that lists itself as the next hop.

### Local Preference Attribute

The local preference attribute is the determining factor in best path selection when there are multiple paths to the same destination. Using the `bgp default local-preference` command, you set the default preference value for routes that the system advertises within the local AS. (The local preference attribute is used with IBGP only.) Routes with a higher preference are

preferred. For example, Routers 1 and 2 both advertise a route to network A. Router 1 advertises the route with a preference of 100, the default. Router 2 advertises a route with a preference of 200. Router 2's route is considered the preferred route to network A by other routers within the AS.

#### Multi-exit Discriminator (MED) Attribute

The MED path attribute, an optional attribute, helps external peers within the same AS derive the preferred path into an AS when there are multiple entry points into it. A lower MED value is preferred over a higher value. You can enable MED comparison of routes from within the same AS (`bgp compare-med`) or for routes from neighbors in different ASs as well as from within the same AS (`bgp always-compare-med`).

#### Community Attribute

A community is a group of destinations that share some common attribute and can therefore permit routing decisions to be imposed based on the group's identity. Each destination can belong to multiple communities, and by default, all destinations belong to the general Internet community. All prefixes with the community attribute belong to the communities listed in the attribute.

The following are "well-known communities," which are reserved community numbers that dictate a specific action:

**Table 18-3. BGP Community Attributes**

Community	Description
Internet	Contains all destinations by default.
No Export (0xFFFFF01)	Prohibits routes with this attribute from being advertised outside of an AS or confederation (a large domain viewed as several smaller domains).
No Advertise (0xFFFFF02)	Prohibits routes with this attribute from being advertised to other BGP peers.
Local-AS or No Export Subconfed (0xFFFFF03)	Prohibits routes with this attribute from being advertised to other ASs.

By default, the system does not send the communities attribute to BGP neighbors.

## Debugging and Logging Facilities

The NX-IS implementation provides debugging and debug message logging facilities to help you troubleshoot BGP problems. In addition, show commands plus the ping command can be used to troubleshoot neighbor connections. For more information on troubleshooting BGP problems, refer to the *NX64000 Troubleshooting Guide*.

## BGP Packet Format

Each BGP packet is preceded by a fixed size header. The header consists of a 16-bit marker field, a two-bit length field, a one-bit type field, and, optionally, data. The marker field contains authentication data; the length field indicates the total length of the packet; the type field indicates the message type; the data field contains the optional content.

Marker (16 bytes)	Length (2 bytes)	Type (1 byte)	Data (variable, up to 4077 bytes)
----------------------	---------------------	------------------	---

**Figure 18-4. BGP Header Format**

### Message Types

As defined in RFC 1771, *A Border Gateway Protocol 4 (BGP-4)*, there are four type codes defined for BGP:

- Open
- Update
- Notification
- Keepalive

After a connection is established, each peer sends an Open message. Upon receipt, the router sends a keepalive message confirming the connection. From that point, peers can exchange update, keepalive, and notification messages. These fields are described in the following sections. If two concurrent TCP connections are established between two peers, the session with the higher BGP identifier prevails (see the *collision-detection* command in the *NX64000 Command Reference*).

### Open Message

A BGP Open message contains the header information described above plus the following fields:

Version (1 byte)	AS (2 bytes)	Hold Time (2 bytes)	BGP Identifier (4 bytes)	Parameter Length (1 byte, opt.)	Parameters (variable, opt.)
---------------------	-----------------	---------------------------	-----------------------------	---------------------------------------	--------------------------------

**Figure 18-5. BGP Open Message Fields**

The minimum length of an Open message is 29 bytes, which includes 19 bytes from the packet header, plus those illustrated above. The following table describes the Open message fields:

**Table 18-4. Description of Open Message Fields**

Field	Description
Version	An unsigned integer indicating the BGP version number. The current version number is 4.
AS	A two-byte unsigned integer indicating the sender's autonomous system number.
Hold Time	A two-byte unsigned integer indicating the maximum number of seconds the router will keep a connection active without having received a message from the peer.
BGP Identifier	A four-byte unsigned integer representing the IP address of the sender. This address is determined on startup and is the same for every local interface and every BGP peer.
Parameters Length	A one-byte unsigned integer indicating, in bytes, the total length of the Parameters field. A zero in this field identifies that no optional parameters are present.
Parameter	A variable-length field that contains information on authentication code and data currently in use.

## Update Message

Update messages carry network reachability information between BGP peers. The update can both advertise a preferred route and request to remove outdated routes. The information contained in these messages is used by the routers to construct a picture of the network topology.

The minimum length of the Update message is 23 octets. Each Update can advertise only one route, although that route can be described by several path attributes. An Update message can also withdraw multiple routes, in which case the Path Attributes and Network Layer Reachability Information fields are empty.

A BGP Update message contains the header information described above plus the following fields:

Unfeasible Route Length (2 bytes)	Withdrawn Routes (variable)	Total Path Attribute Length (2 bytes)	Path Attributes (variable)	Network-layer Reachability Information (variable)
-----------------------------------	-----------------------------	---------------------------------------	----------------------------	---

**Figure 18-6. BGP Update Message Fields**



The following table describes the fields:

**Table 18-5. Description of Update Message Fields**

Field	Description
Unfeasible Routes Length	A two-byte unsigned integer indicating the number of bytes in the Withdrawn Routes field. A value of 0 indicates that no routes are being withdrawn with this Update, and therefore, the Withdrawn Routes field is not present.
Withdrawn Routes	A variable-length field listing, by IP address, the routes being withdrawn with this Update message. Routes are withdrawn when they are deemed either no longer reachable or inferior to an alternate route.
Total Path Attribute Length	A two-byte unsigned integer indicating the number of bytes in the Path Attributes field. A value of 0 indicates that no Path Attributes are included in this Update, and therefore, the Path Attributes field is not present.
Path Attributes	<p>A variable-length field listing the attributes of the route. (The attribute types are described in more detail in <a href="#">“Path Attributes” on page 18-7.</a>) The following attribute types are available:</p> <ul style="list-style-type: none"><li>• Origin—describes the route’s source (IGP, EGP, or Incomplete).</li><li>• AS_Path—describes an unordered or order list of the ASs the packet has traveled.</li><li>• Next-hop—the IP address of the router that should be used as the next hop to a destination.</li><li>• MED—a value that establishes a preference for an exit point from an AS when more than one exists.</li><li>• Local_pref—informs other speakers within the same AS of the local routers preference for a route.</li><li>• Atomic_aggregate— indicates that from a set of overlapping routes, the local router has selected a less specific route.</li><li>• Aggregator—contains the last AS number and the IP address of the BGP peer that formed the aggregate route.</li></ul>
Network-layer Reachability Information (NLRI)	A variable-length field containing a list of IP address prefixes for valid routes. All attributes listed in the Path Attributes field apply to any address prefix in this field.

## Keepalive Message

Keepalive messages are exchanged between peers to verify reachability status. BGP uses the hold time interval setting to establish frequency of keepalive messages. Usually, a keepalive message is sent every one-third of the hold time, with a minimum time of one per second. If the hold time interval is zero, keepalive messages are not sent. A keepalive message is 19 bytes and contain only the BGP header.

## Notification Message

BGP sends a notification message as soon as it detects an error on a connection, and then it immediately closes the connection. When the transport protocol connection closes, all resources for that BGP connection are deallocated. Routing table entries associated with the remote peer are marked as invalid. A BGP notification message contains the header information described above plus the following fields:

Error Code (1 byte)	Error Subcode (1 byte)	Error Data (variable)
------------------------	---------------------------	--------------------------

**Figure 18-7. BGP Notification Message Fields**

A notification message has a minimum length of 21 bytes. The following table describes the notification message fields:

**Table 18-6. Description of Notification Message Fields**

Field	Description
Error Code	A one-byte unsigned integer indicating the error code, which represents the error's type. The code and type possibilities are: <ul style="list-style-type: none"><li>• 1— Message Header Error</li><li>• 2— OPEN Message Error</li><li>• 3— UPDATE Message Error</li><li>• 4— Hold Timer Expired</li><li>• 5— Finite State Machine Error</li><li>• 6— Cease (close connection although there are no fatal errors)</li></ul>
Error Subcode	A one-byte field specifying more error information. An error code can have more than one error subcode associated with it. Each error code type has its own listing of subcodes. (See RFC 1771 for more information.)
Error Data	A variable-length field describing the reason for the notification message.

## Controlling Routing Information Flow

BGP supports several features that optimize network efficiency by minimizing routing tables and updates. The following sections describe some of those features.

### Peer Groups

A peer group is a logical association of neighbors that share the same configuration attributes, such as update policies and passwords. You use peer groups to apply multiple attributes to a group of routers. As you assign members to a peer group, they inherit the parameter settings from the group.

### Route Reflectors

Originally, for BGP routing table distribution to work, all BGP peers within an AS had to be fully meshed so that any router receiving external routing information could redistribute it internally. This caused fairly severe scalability problems, so route reflection (RR) was developed. RR supports a partially meshed IBGP topology by allowing a peer receiving an update from the sending speaker to forward it on to a non-meshed peer within the AS. This peer is the router reflector.

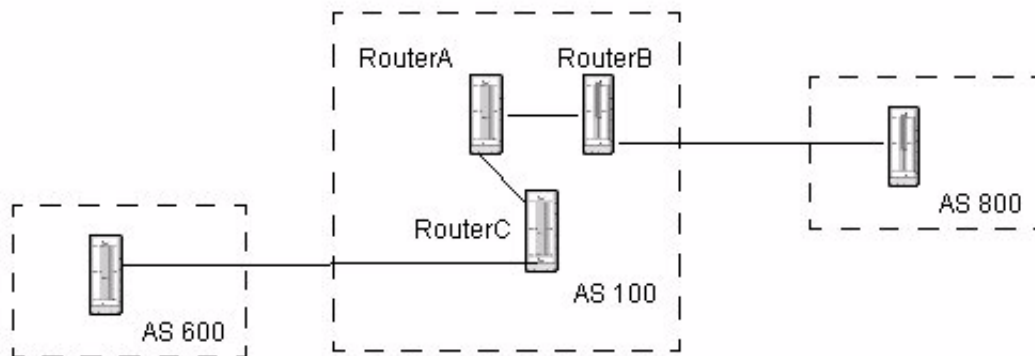
There are two types of RR peers—client and non-client. A route reflector and its client peers form a cluster, which allows for a non-fully meshed topology. Clients do not peer with RRs outside of their clusters. Non-client peers must be fully meshed.

An RR behaves as described in the following table when it selects a best path:

**Table 18-7. Route Reflector Behavior**

Route received from	Action
Non-Client peer	Reflect to all other clients.
Client peer	Reflect to all the non-client peers. Reflect to all client peers except the originator.
EBGP peer	Send to all the client and non-client peers.

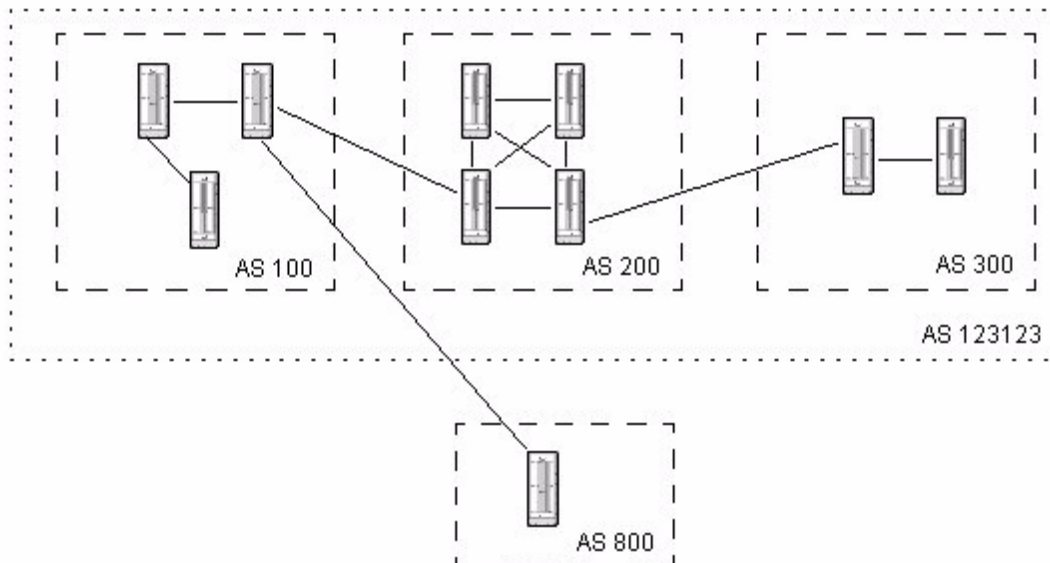
**Figure 18-8** illustrates a backbone with route reflection taking place. Although RouterB and RouterC are not directly connected, RouterA will communicate updates from one to the other.



**Figure 18-8. Route Reflection Eliminate the Need for Full Mesh in an AS**

### Confederation

An AS confederation is a supernet. That is, it is a grouping of ASes that advertise themselves as a single AS to any peer outside of the grouping. Peers receiving advertisements from a router with the same confederation ID process the packet as they would if the peer had the same AS number. The confederation uses a confederation identifier, which acts as an AS number for the for all AS members and represents the group externally. A member uses its routing domain identifier (the internally visible AS number) to communicate with peers within the confederation.



**Figure 18-9. An Example of a BGP Confederation, AS123123**

## Aggregation

Route aggregation allows you to advertise many routes with a single route table entry. When configured for route aggregation, the system recognizes a less-specific match as the routing table entry for multiple more-specific routes.

For example, setting aggregation for the 12.0.0.0/8 network entry causes the router to use that route for all networks that begin with 12. This can significantly reduce the size of the routing table and minimizes the number of routing table updates.

## Redistribution

Redistribution alters the routing scheme to redistribute routes from other routing protocols into BGP. To keep track of network routes, BGP uses both a main routing table (which contains the best path learned through all routing protocols) and a BGP routing table (which contains only BGP-learned best paths). Although the BGP table updates with each advertisement containing a preferable route, when the main table is updated (for example, from OSPF), the BGP table is not notified. To get paths from the main table into the BGP table, they must be redistributed into BGP.

- The best practice when redistributing routes is to redistribute from an IGP, such as IS-IS or OSPF, into BGP. Redistributing BGP into an IGP is not recommended. It requires extremely well-executed filters to prevent serious harm to your network. For example, a miss-configured filter could allow the entire Internet routing table to be redistributed into an IGP running on a router in the core of your network, causing a network breakdown.

## Route Filtering

BGP uses policy-based route filtering to control routing updates. For specific information on route filtering, see [Chapter 19, “Route Filter Configuration.”](#) Generally, route filters define the list of network prefixes that are allowed in (received) or out (advertised) from an interface.

## Import Policies

Import policies determine whether a route should be accepted and added to the routing table. Inbound filters add stability and security to the network. You can define filters based on such criteria as local preferences, AS patterns, peer ASs, communities, networks, and prefix-lengths.

## Export Policies

Export policies determine whether a BGP speaker should advertise a route. You can define advertising criteria based on many factors, such as AS path, IP address, next-hop, route source, and origin to name a few.

## Redistribution Policies

Redistribution policies control outgoing routes being sent from the local router to non-BGP routers. Examples of redistribution criteria include coefficients, administrative distance, OSPF or IS-IS levels, metrics, and router addresses.

## BGP Configuration

The following table lists all the BGP commands supported by the NX64000 switch/router, as well as the route filtering commands. Each command listing indicates the general functions for which you would use the command. Some commands are included and described in the configuration examples that follow. All commands listed are documented in the *NX64000 Command Reference*.

**Table 18-8. BGP Command Usage**

Command	Enabling BGP	Administration	Neighbor Administration	Flow Control: Redistribution	Flow Control: Peer Groups	Network Configuration	Attributes: Routing Decision	Route Filtering Policy
bgp always-compare-med							✓	
bgp client-to-client reflection						✓		
bgp cluster-id						✓		
bgp compare-med							✓	
bgp confederation identifier						✓		
bgp confederation peers						✓		
bgp dampening		✓						
bgp default local-preference							✓	
bgp fast-external-fallover		✓						
bgp recursion		✓						
clear ip bgp	✓	✓						✓
collision-detection		✓						
commit	✓	✓						
default-information originate				✓				
default-metric							✓	
ip as-path access-list				✓				✓
ip community-list				✓				✓

**Table 18-8. BGP Command Usage**

Command	Enabling BGP	Administration	Neighbor Administration	Flow Control: Redistribution	Flow Control: Peer Groups	Network Configuration	Attributes: Routing Decision	Route Filtering Policy
neighbor advertisement-interval			✓					
neighbor as-origination-interval			✓					
neighbor default-originate			✓				✓	
neighbor description			✓					
neighbor ebgp-multihop			✓					
neighbor maximum-prefix			✓			✓		
neighbor next-hop-self			✓				✓	
neighbor peer-group			✓		✓			
neighbor remote-as	✓		✓					
neighbor remove-private-as			✓					
neighbor retry-timer			✓					
neighbor route-reflector-client			✓			✓		
neighbor send-community			✓				✓	
neighbor shutdown	✓		✓					
neighbor soft-reconfiguration inbound			✓					
neighbor timers			✓					
neighbor update-source			✓		✓			
neighbor version			✓					
neighbor weight			✓				✓	
network				✓			✓	
permit-internal-into-ibgp				✓				

**Table 18-8. BGP Command Usage**

Command	Enabling BGP	Administration	Neighbor Administration	Flow Control: Redistribution	Flow Control: Peer Groups	Network Configuration	Attributes: Routing Decision	Route Filtering Policy
redistribute				✓				✓
router bgp	✓							
show ip as-path-access-list		✓		✓				✓
show ip bgp		✓						
show ip bgp build-info		✓						
show ip bgp cidr-only		✓						
show ip bgp community		✓		✓				✓
show ip bgp dampened-paths		✓						
show ip bgp flap-statistics		✓						
show ip bgp neighbors		✓	✓					
show ip bgp neighbors events		✓	✓					
show ip bgp neighbors timers		✓	✓					
show ip bgp paths		✓						
show ip bgp peer-group		✓			✓			
show ip bgp regexp		✓						
show ip bgp rejected		✓						
show ip bgp summary		✓						
show ip bgp unreachable		✓						
show ip community-list		✓		✓				✓
standard-compliant		✓						
synchronization				✓				



**Table 18-8. BGP Command Usage**

Command	Enabling BGP	Administration	Neighbor Administration	Flow Control: Redistribution	Flow Control: Peer Groups	Network Configuration	Attributes: Routing Decision	Route Filtering Policy
table-map				✓				✓
timers bgp		✓						
<i>BGP Route Filtering (see the “Route Filtering Configuration” chapter)</i>								
match as-path								✓
match community-list								✓
match interface								✓
match ip-address								✓
match ip next-hop								✓
match ip route-source								✓
match metric								✓
match route-type								✓
match tag								✓
neighbor distribute-list								✓
neighbor filter-list								✓
neighbor route-map								✓
route-map								✓
set as-path								✓
set automatic-tag								✓
set community								✓
set dampening								✓
set ip next-hop								✓

**Table 18-8. BGP Command Usage**

Command	Enabling BGP	Administration	Neighbor Administration	Flow Control: Redistribution	Flow Control: Peer Groups	Network Configuration	Attributes: Routing Decision	Route Filtering Policy
set level								✓
set local-preference								✓
set metric								✓
set metric-type								✓
set next-hop								✓
set origin								✓
set tag								✓
set weight								✓
show route-map								✓

## Basic BGP Configuration Tasks

The following sections illustrate the settings used to configure BGP in this document's network example. Refer to the first part of this chapter for information on the features.

In general, to configure BGP, you must do the following:

1. Enable the protocol.
2. Define neighbors and their parameters.
3. Define route filters.
4. Redistribute routes.

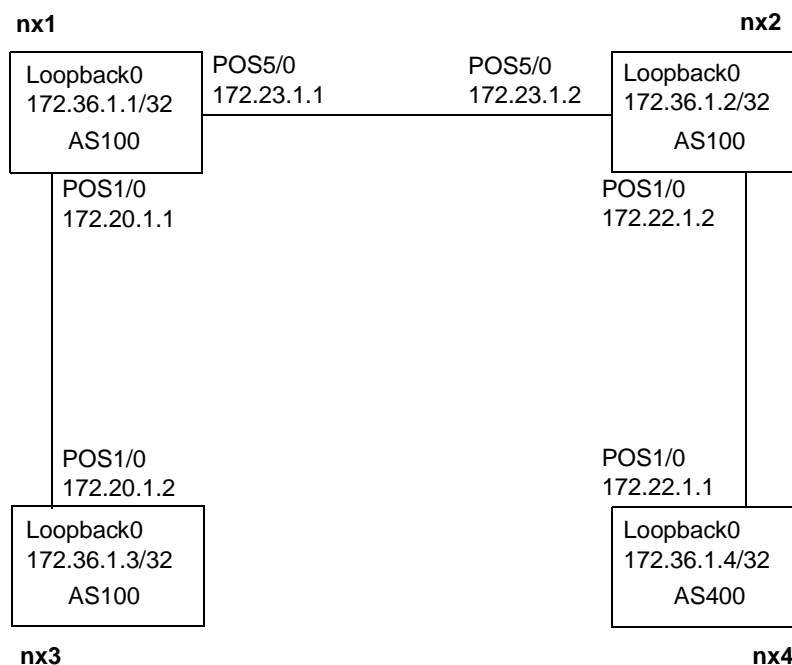
In the configuration described in the following sections, the general configuration is accomplished by:

1. Enabling BGP and assigning the router to an autonomous system.
2. Defining neighbors and their attributes.
3. Defining incoming and outgoing filters with distribute lists and access lists.
4. Defining route maps and peer groups for redistribution.
5. Defining network configuration parameters for network management:

- Route reflectors
  - Aggregation
6. Defining attributes for routing decision criteria.

► You must configure route filtering policy before BGP can begin advertising routes to or learning routes from neighbors. Because the default routing policy action is to deny all routes, no routes will be exchanged until you define policies.

The figure below illustrates the BGP network as configured in this guide:



**Figure 18-10. BGP Network Example**

## Enabling BGP

After configuring interfaces, load and enable BGP on the router. Be sure that you have configured a loopback interface on each router for BGP to use to form neighbor relationships. In the configurations below, NX1, NX2, and NX3 enable BGP and are members of AS 100. They are running IBGP. NX4 enables BGP and is assigned to AS 400, and is running EBGP.

If internal routes, such as local or OSPF routes, are redistributed into BGP, they are only advertised to external peers. Generally, it is not desirable to advertise internal routes to internal peers since they already should have been advertised via IGP. Because NX-IS denies all incoming and outgoing routes by default, you must explicitly allow announcements of internal networks to internal peers using the `permit-internal-into-ibgp` command.

NX1:

```
nx1# configure terminal
nx1(config)# hostname nx1
nx1(config)# load bgp
nx1(config)# router bgp 100
nx1(router-bgp)# permit-internal-into-ibgp
```

NX2:

```
nx2# configure terminal
nx2(config)# hostname nx2
nx2(config)# load bgp
nx2(config)# router bgp 100
nx2(router-bgp)# permit-internal-into-ibgp
```

NX3:

```
nx3# configure terminal
nx3(config)# hostname nx3
nx3(config)# load bgp
nx3(config)# router bgp 100
nx3(router-bgp)# permit-internal-into-ibgp
```

NX4:

```
nx4# configure terminal
nx4(config)# hostname nx4
nx4(config)# load /bgp
nx4(config)# router bgp 400
nx4(router-bgp)# permit-internal-into-ibgp
```

The following table explains each line in the preceding example:

Configuration Line	Description
<b>hostname nx1</b>	Assigns a hostname to the router.
<b>load bgp</b>	Loads BGP from software release 1.6.0
<b>router bgp 100</b>	Enables BGP on the router and assigns it to an AS, in this case, 100. Note that the NX4 is assigned to AS 400.
<b>permit-internal-into-ibgp</b>	Allows the router to announce IGP-learned routes to internal peers.

## Neighbor Statements

Neighbor statements set configuration parameters that define the relationship between the local router and the specified neighbor. The following table lists the possible parameters that you can set and indicates which are described in this chapter:

**Table 18-9. BGP Neighbor Statements**

Command	Description	Example Included
<code>neighbor advertisement-interval</code>	Sets the interval between updates.	
<code>neighbor as-origination-interval</code>	Sets the interval between default-route advertisements.	
<code>neighbor default-originate</code>	Configures the router to send the default route 0.0.0.0.	
<code>neighbor description</code>	Associates a text string with the neighbor.	
<code>neighbor distribute-list</code>	Sets up filters for incoming or outgoing BGP routes based on an access list.	yes
<code>neighbor ebgp-multihop</code>	Configures the router to allow connections to peers on non-direct connect networks.	
<code>neighbor filter-list</code>	Sets up filters for incoming or outgoing BGP routes based on an IP AS-path access list.	yes
<code>neighbor maximum-prefix</code>	Sets the limit on prefixes the router can receive from that neighbor.	
<code>neighbor next-hop-self</code>	Configures the router to advertise itself as the next hop for a neighbor.	
<code>neighbor peer-group</code>	Creates a peer group or adds a neighbor.	yes
<code>neighbor remote-as</code>	Add an entry to the BGP neighbor table and identify the AS in which the neighbor resides.	yes
<code>neighbor remove-private-as</code>	Configures the router to send external updates without private AS numbers.	
<code>neighbor retry-timer</code>	Sets the interval between connection attempts.	
<code>neighbor route-reflector-client</code>	Configures the router as a route reflector and the neighbor as a client.	yes
<code>neighbor send-community</code>	Configures the router to include the communities attribute in updates.	
<code>neighbor shutdown</code>	Disables a neighbor or peer group without deleting it from the configuration.	

**Table 18-9. BGP Neighbor Statements**

Command	Description	Example Included
<b>neighbor soft-reconfiguration inbound</b>	Prevents the router from requiring a reset of the BGP state with the neighbors to which it is peered every time it receives a policy or route change from that neighbor. This state reset is time consuming and can potentially disrupt the rest of the network. This parameter is enabled by default, and cannot be disabled.	
<b>neighbor timers</b>	Sets the keepalive and holdtime timers	
<b>neighbor update-source</b>	Permits IBGP sessions to use any operational interface for TCP connections to the neighbor.	yes
<b>neighbor version</b>	Sets the version that these peers will use to communicate.	
<b>neighbor weight</b>	Overrides weight values inherited from a neighbor or peer group.	yes

In the configuration example below, NX1, NX2, and NX3 are running IBGP. NX4 is running EBGP. (See [Figure 18-10 on page 18-21](#) for an illustration of the network.) The routers neighbor as follows:

- NX1 neighbors with the loopback0 interface on NX2 and NX3.
- NX2 neighbors with the loopback0 interface on NX1 and with pos1/0 on NX4.
- NX3 neighbors with the loopback0 interface on NX1.
- NX4 neighbors with pos1/0 on NX2.

NX1:

```
nx1(config)# router bgp 100
nx1(router-bgp)# neighbor 172.36.1.2 remote-as 100
nx1(router-bgp)# neighbor 172.36.1.3 remote-as 100
```

NX2:

```
nx2(config)# router bgp 100
nx2(router-bgp)# neighbor 172.36.1.1 remote-as 100
nx2(router-bgp)# neighbor 172.22.1.1 remote-as 400
```

NX3:

```
nx3(config)# router bgp 100
nx3(router-bgp)# neighbor 172.36.1.1 remote-as 100
```

NX4:

```
nx4(config)# router bgp 400
nx4(router-bgp)# neighbor 172.22.1.2 remote-as 100
```

The following table explains the commands in the preceding example:

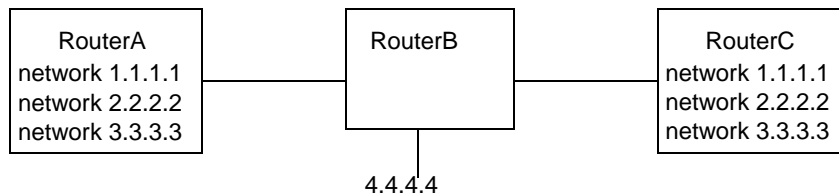
Configuration Line	Description
<code>router bgp 400</code>	Enters router-bgp configuration mode, enables BGP on the router, and assigns it to an AS.
<code>neighbor address remote-as as</code>	Defines a neighbor router.

## Controlling Route Advertisements

You control advertisements through a combination of the `neighbor distribute-list` and `access-list` commands. You also use the `network` command to specify which networks to advertise. The following sections provide examples using of these commands.

### Configuring Which Networks to Announce

The `network` command defines which networks a router is allowed to announce to its neighbors. In the following illustration, RouterA's configuration file defines networks 1.1.1.1, 2.2.2.2, and 3.3.3.3 with the `network` command. RouterB, which is connected to 4.4.4.4, does not have the network defined in its configuration, and therefore cannot advertise it. The result is that RouterC only knows of networks 1.1.1.1, 2.2.2.2, and 3.3.3.3. Even though it learns of them through RouterB, it does not learn network 4.4.4.4, which is connected to RouterB.



**Figure 18-11. The Network Command Defines Advertised Networks**

In the example, NX1 advertises its loopback2 interface (172.37.1.1) and the EBGp 192.17.x.x network. NX2 advertises its loopback2 interface (172.37.1.2). NX3 advertises its loopback2 interface (172.37.1.3) and the EBGp 172.21.x.x network. NX4 advertises its two loopback interfaces—172.36.1.4 and 172.37.1.4—and the EBGp 172.21.x.x network.

NX1:

```

nx1(router-bgp)# network 172.37.1.1 mask 255.255.255.255
nx1(router-bgp)# network 192.17.0.0 mask 255.255.0.0
  
```

NX2:

```

nx2(router-bgp)# network 172.37.1.2 mask 255.255.255.255
  
```

NX3:

```
nx3(router-bgp)# network 172.37.1.3 mask 255.255.255.255
nx3(router-bgp)# network 172.21.0.0 mask 255.255.0.0
```

NX4:

```
nx4(router-bgp)# network 172.36.1.4 mask 255.255.255.255
nx4(router-bgp)# network 172.37.1.4 mask 255.255.255.255
nx4(router-bgp)# network 172.21.0.0 mask 255.255.0.0
```

The following table explains the `network` command:

Configuration Line	Description
<code>network address mask mask</code>	Defines the network number that the router advertises. The mask is optional, and if omitted, a default mask based is assigned.

## Using Distribute Lists

Distribute lists work in conjunction with access lists to create filters that the router applies to incoming and/or outgoing routes. A distribute list is similar to a filter list, but filters on network numbers instead of AS numbers. In our network example, NX2 defines an incoming and outgoing distribute-list for each neighbor. A distribute list filters BGP advertisements for specific neighbors or peer groups. The filtering criteria is stored in the access list which is referred to in the `neighbor distribute-list` command:

```
neighbor ip-addr distribute-list access-list-# {in | out}
```

For NX1 and NX3, incoming and outgoing advertisements are filtered using access-list 2. NX4 uses access-list 2 for incoming and access-list 3 for outgoing advertisements. See [“Using Access Lists” on page 18-27](#) for a definition of access-lists 2 and 3.

NX2:

```
nx2(router-bgp)# neighbor 172.36.1.1 distribute-list 2 in
nx2(router-bgp)# neighbor 172.36.1.1 distribute-list 2 out
nx2(router-bgp)# neighbor 172.36.1.3 distribute-list 2 in
nx2(router-bgp)# neighbor 172.36.1.3 distribute-list 2 out
nx2(router-bgp)# neighbor 172.22.1.1 distribute-list 2 in
nx2(router-bgp)# neighbor 172.22.1.1 distribute-list 3 out
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<code>neighbor address distribute-list 2 in</code> <code>172.36.1.1</code> <code>172.36.1.3</code> <code>172.22.1.1</code>	Controls NX2’s ability to propagate networks to the specified neighbor. Because access-list 2 permits any, all networks can be propagated.



Configuration Line	Description
<code>neighbor address distribute-list 2 out</code> <code>172.36.1.1</code> <code>172.36.1.3</code> <code>172.22.1.1</code>	Configures NX2's ability to learn networks from the specified neighbor. Because access-list 2 permits any, all networks can be learned.
<code>neighbor 172.22.1.1 distribute-list 3 out</code>	Configures NX2's ability to learn networks from NX4.

## Using Access Lists

Access lists regulate routing updates on an interface. They are called by the `neighbor distribute-list` command, and define an IP address for matching and an action of permit or deny if match should take place. Access to an address is denied by default, and must be explicitly granted in an access list.

Access list 2 permits incoming and outgoing advertisements between internal neighbors (NX1 and NX3) and permits incoming advertisements from NX4. It uses access list 3 to prohibit outgoing advertisements of networks 172.20.x.x, 172.36.x.x, 192.17.x.x, and 192.18.x.x (but permit all others) to NX4. To understand the which access-list applies to which neighbor, see the `neighbor distribute-list` command description above.

NX2:

```
nx2(router-bgp)# access-list 2 permit any
nx2(router-bgp)# access-list 3 deny 172.20.0.0 0.0.255.255
nx2(router-bgp)# access-list 3 deny 172.36.0.0 0.0.255.255
nx2(router-bgp)# access-list 3 deny 192.17.0.0 0.0.255.255
nx2(router-bgp)# access-list 3 deny 192.18.0.0 0.0.255.255
nx2(router-bgp)# access-list 3 permit any
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<code>access-list 2 permit any</code>	Configures access list 2 to forward any packet whose source is not currently prohibited by a deny statement for this access list.
<code>access-list 3 deny address mask</code> <code>172.20.0.0 0.0.255.255</code> <code>172.36.0.0 0.0.255.255</code> <code>192.17.0.0 0.0.255.255</code> <code>192.18.0.0 0.0.255.255</code>	Prohibit outgoing advertisements of networks 172.20/16, 172.36/16, 192.17/16, and 192.18/16. This command is used in conjunction with, among others, the <code>neighbor distribute-list</code> command above to block redistribution of these networks to NX4.

Configuration Line	Description
<code>access-list 3 permit any</code>	Configures access list 3 to forward any packet whose source is not currently prohibited by a deny statement for this access list. (See deny statements above.)

## Using Filter Lists and AS-Path Access Lists

Filter lists work in conjunction with AS-path access lists to create filters that the router applies to incoming and/or outgoing routes. A filter list is similar to a distribute list, but filters on AS paths instead of network numbers. In the example below, the filter is applied to those incoming routes advertised by neighbor 192.17.1.2 (NX4). The `neighbor filter-list` command calls AS-path access list 50, which is used for AS-path filtering. Both statements use regular expressions to create pattern matches. The deny statement translates, “deny anything whose AS path list includes first AS 400, then AS 655403.” This configures the NX1 to get only those routes coming directly from the NX4. The permit statement translates to permit anything.

NX1:

```
nx1(router-bgp)# neighbor 192.17.1.2 filter-list 50 in
nx1(router-bgp)# exit
nx1(config)# ip as-path access-list 50 deny ^400 655403
nx1(config)# ip as-path access-list 50 permit .*
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<code>neighbor 192.17.1.2 filter-list 50 in</code>	Specifies the neighbor for which you want to filter routes and the access-list to apply against that neighbor.
<code>exit</code>	Returns to the previous configuration mode.
<code>ip as-path access-list 50 deny ^400 655403</code>	Defines access-list 50 to deny any routes that include in their AS path list AS 400 followed by AS 655403.
<code>ip as-path access-list 50 permit .*</code>	Defines access-list 50 to permit any list of AS numbers in the AS path list.

## Using Filtering and Route Redistribution for Flow Control

There are several ways to manipulate flow control. Generally, you can either filter incoming and outgoing routes for redistribution or manage network configuration issues. The following sections use redistribution, route maps, peer groups, route reflectors, and filter lists to control route information.

## Using Route Maps

Route map INSIDEMAP permits redistribution of any route that matches the criteria specified for the map. In the following configuration, a route whose destination address matches those addresses listed in access list 2 can be redistributed. Access list 2 specifies permission to redistribute any IP address. Further use of access lists in BGP is described in the section [“Configuring Route Reflectors” on page 18-31](#).

- The specifics of route maps are described in [Chapter 19, “Route Filter Configuration.”](#) Access lists are fully described in [Chapter 12, “Access List Configuration.”](#)

NX1:

```
nx1# configure terminal
nx1(config)# route-map INSIDEMAP permit 10
nx1(config-route-map)# match ip address 2
nx1(config-route-map)# exit
nx1(config)# access-list 2 permit any
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<code>route-map INSIDEMAP permit 10</code>	Creates (or modifies) route map INSIDEMAP, assigns it a sequence number of 10 (should there be other route maps with the same name), and permits redistribution of any route that matches the match criteria.
<code>match ip address 2</code>	Allows redistribution of any route that matches the routes listed in access list 2.
<code>exit</code>	Exits route-map configuration mode.
<code>access-list 2 permit any</code>	Configures access list 2 to forward any packet whose source is not currently prohibited by a deny statement for this access list. (Any IP address is permissible as source host.)

## Using Peer Groups for Route Redistribution

Peer groups are managed using BGP neighbor statements as well. The `neighbor peer-group` command both creates a peer group and then assigns members to it. After creating a peer group, assign attributes to the group. As you assign members, they inherit the attributes. Most neighbor statement commands accept a peer-group name in place of an IP address; those commands are described in the section on neighbor statements.

In the configuration example below, the NX1 creates a peer group called INSIDEPG and then assigns neighbors 172.36.1.2 (loopback0 on NX2) and 172.36.1.3 (loopback0 on NX3) to the peer group. Members of the peer group belong to AS 100, use a route map named INSIDEMAP to both incoming and outgoing routes, and use soft reconfiguration on inbound routes. See [“Using Route Maps” on page 18-29](#) for a description of using the route-map command.

NX1:

```
nx1# configure terminal
nx1(config)# router bgp 100
nx1(router-bgp)# neighbor INSIDEPG peer-group
nx1(router-bgp)# neighbor INSIDEPG remote-as 100
nx1(router-bgp)# neighbor INSIDEPG route-map INSIDEMAP out
nx1(router-bgp)# neighbor INSIDEPG route-map INSIDEMAP in
nx1(router-bgp)# neighbor INSIDEPG update-source loopback0
nx1(router-bgp)# neighbor 172.36.1.2 peer-group INSIDEPG
nx1(router-bgp)# neighbor 172.36.1.3 peer-group INSIDEPG
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<code>neighbor INSIDEPG peer-group</code>	Creates a peer group named INSIDEPG.
<code>neighbor INSIDEPG remote-as 100</code>	Defines members of INSIDEPG as belonging to AS 100.
<code>neighbor INSIDEPG route-map INSIDEMAP out</code>	Assigns members of INSIDEPG to use route map INSIDEMAP for redistributing outgoing routes.
<code>neighbor INSIDEPG route-map INSIDEMAP in</code>	Assigns members of INSIDEPG to use route map INSIDEMAP for redistributing incoming routes.
<code>neighbor INSIDEPG update-source loopback0</code>	Configures members of INSIDEPG to use interface loopback0 for TCP connections to neighbors.
<code>neighbor 172.36.1.2 peer-group INSIDEPG</code>	Assigns neighbor 172.36.1.2 to peer group INSIDEPG.
<code>neighbor 172.36.1.3 peer-group INSIDEPG</code>	Assigns neighbor 172.36.1.3 to peer group INSIDEPG.

## Managing Network Configuration

BGP offers several features that allow for a more efficient network configuration. By grouping routers, you can control route advertisements and allow for non-fully meshed networks.

## Configuring Route Reflectors

Because this network does not form a fully meshed topology within the IBGP, you must enable route reflection to ensure accurate routing tables. In the following example, the local router, NX1, is the route reflector serving clients NX2 and NX3. These routers become a cluster, identified as cluster 111.

NX1:

```
nx1(router-bgp)# neighbor 172.36.1.2 route-reflector-client
nx1(router-bgp)# neighbor 172.36.1.3 route-reflector-client
nx1(router-bgp)# bgp cluster-id 111
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<b>neighbor address route-reflector-client</b> <b>172.36.1.1</b> <b>172.36.1.3</b>	Specifies the client of the local router, making the router a route reflector. Clients must be physically connected to the route reflector, but not each other.
<b>bgp cluster-id 111</b>	Assigns a cluster ID to the router. The cluster ID is used in routing updates within the cluster.

## Configuring Route Aggregation

You can configure static route aggregation on the NX switch/router. The advantage to static aggregation is that the router is impervious to flapping of the network at the other end of the static route. However, due to its static nature, should the network cease to be available, the router will continue to advertise the address.

The **network** command does not cause the router to advertise every route it is configured to announce, the router must learn the route from somewhere. In the following example, no router will advertise 192.0.0.0/8 to NX4. Instead, neighbors will advertise more specific routes, for example, 192.0.0.0/16 or 192.0.0.0/24. To get NX4 to advertise the aggregated address 192.0.0.0/8, you must invent a static route and use the **network** command to configure BGP to advertise the address. Note that the next hop for the static route is "null 0", i.e., the "bit bucket". NX4 is just using the route as a way to get the aggregated route advertised, so the next-hop doesn't matter. When it comes to actually routing packets to 192.x.x.x addresses, the more specific routes are used. Also note, to prevent the router from advertising the more specific routes along with the aggregated route, you must configure outbound route filters for any routes you do not want advertised.

NX4:

```
nx4(config)# ip route 192.0.0.0 255.0.0.0 null0
nx4(config)# access-list 101 deny ip 192.0.0.0 0.255.255.255 255.128.0.0
0.127.255.255
nx4(config)# access-list 101 permit ip any any
nx4(config)# router bgp 100
nx4(router-bgp)# network 192.0.0.0 mask 255.0.0.0
nx4(router-bgp)# neighbor 172.22.1.2 remote-as 400
```

```
nx4(router-bgp)# neighbor 172.22.1.2 distribute-list 101 out
nx4(router-bgp)# exit
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<code>ip route 192.0.0.0 255.0.0.0 null 0</code>	Defines a static route which represents the aggregated address.
<code>access-list 101 deny ip 192.0.0.0 0.255.255.255 255.128.0.0 0.127.255.255</code>  <code>access-list 101 permit ip any any</code>	Configures route filters (via extended access lists) to prevent advertisement of more specific routes of the 192 network (e.g., 192.1.0.0, 192.2.0.0, etc.) and permit all others.
<code>router bgp 100</code>	Enters BGP router configuration mode.
<code>network 192.0.0.0 mask 255.0.0.0</code>	Specifies that the router should advertise the route 192.0.0.0, but for it to do so, it must learn the route from somewhere. (See the <code>ip route</code> command description above.)
<code>neighbor 172.22.1.2 remote-as 100</code>	Defines a neighbor router.
<code>neighbor 172.22.1.2 distribute-list 101 out</code>	Applies access list 101 to outbound advertisements headed to neighbor 172.22.1.2
<code>exit</code>	Leaves BGP configuration mode and returns to terminal configuration mode.

## Using Attributes

Attributes help the router make routing decisions when multiple otherwise-equal paths exist to a destination. Many of the attributes are configured using `set` and `match` route filtering commands (described in [Chapter 19, “Route Filter Configuration”](#)).

In the example, route maps are set both directly and using a route-map statement. The route map sets the routes going from NX1 to NX4 to 200. The neighbor weight statement sets routing preferences, with higher values being preferred.

NX1:

```
nx1(router-bgp)# neighbor 192.17.1.2 route-map MEDVAL out
nx1(router-bgp)# neighbor 192.17.1.2 weight 30000
nx1(router-bgp)# exit
nx1(config)# route-map MEDVAL permit 10
nx1(config-route-map)# set metric 200
```

The following table explains the commands in the preceding example:

Configuration Line	Description
<code>neighbor 192.17.1.2 route-map MEDVAL out</code>	Assigns interface 192.17.1.2 to use route map MEDVAL when redistributing outgoing routes.
<code>neighbor 192.17.1.2 weight 30000</code>	Sets the weight of all routes learned from NX4 to 30000, increasing chances of it being a preferred route.
<code>route-map MEDVAL permit 10</code>	Creates (or modifies) route map MEDVAL, assigns it a sequence number of 10 (should there be other route maps with the same name), and permits redistribution of any route that matches the match criteria.
<code>set metric 200</code>	Sets a metric value (the MED for BGP routes) of 200 to routes that have MEDVAL route-map applied.

## Verifying BGP Configuration

The NX-IS software provides a number of show commands that let you verify configuration and neighbor status. The following tables lists the type of information you can view from each show command:

**Table 18-10. BGP Commands for Verifying Configuration**

Action	Command
List AS path access lists.	<code>show ip as-path-access-list</code>
Display the BGP routing table.	<code>show ip bgp</code>
Display the date and time of the latest build.	<code>show ip bgp build-info</code>
Display routes using CIDR.	<code>show ip bgp cidr-only</code>
Display routes based on their BGP communities.	<code>show ip bgp community</code>

**Table 18-10. BGP Commands for Verifying Configuration**

Action	Command
Display BGP dampened routes.	show ip bgp dampened-paths
Display BGP flap statistics for the path or address specified.	show ip bgp flap-statistics
Display information about the TCP and BGP connections to neighbors.	show ip bgp neighbors
Display detailed status information about neighbor traffic.	show ip bgp neighbors events
Display KeepAlive and HoldTime timer information.	show ip bgp neighbors timers
Display all the BGP paths in the database.	show ip bgp paths
Display information about BGP peer groups.	show ip bgp peer-group
Display routes matching the regular expression argument.	show ip bgp regexp
Display received routes that are rejected (not installed as valid routes) by the import route filter policy configuration.	show ip bgp rejected
Display summary status of all BGP connections.	show ip bgp summary
Display unreachable routes.	show ip bgp unreachable
Display routes belonging to specified BGP communities.	show ip community-list

## Interoperability Issues

If you use a Cisco 12000 GSR with the NX64000 switch/router, you should be aware of the following differences. This table does not include command syntax or valid value range differences.



Feature	Cisco IOS Implementation	NX-IS Software Implementation
Route installation	All routes are accepted by default. You must define policies to deny routes.	All routes are denied by default. A received route must first pass the decision criteria defined by the applied routing policies before it is installed in the RIB.
Announcement of internal routes to internal peers	Cisco always advertises redistributed routes to both internal and external peers unless you configure policies to deny them.	NX-IS only advertises redistributed internal routes to external peers, because usually IGP advertises to internal peers. You must explicitly allow announcements of internal networks to internal peers using the <code>permit-internal-into-ibgp</code> command.
Route aggregation	Supported with aggregate address command. See Cisco documentation.	NX-IS does not currently support an aggregate address command. Static route aggregation can be done with the <code>network</code> command. See <a href="#">“Configuring Route Aggregation” on page 18-31</a> for details.
Route maps for peers	Neighbors in a peer group keep their own inbound route maps regardless of whether a route map was explicitly associated with the peer group.	A neighbor configured as a peer group member takes the associated peer group settings. That is, inbound and outbound route maps override any individual associations.
MED value comparison for routes within an AS	Compares MED values for routes received from IBGP peers. Comparison cannot be disabled.	MED comparison is disabled by default. Enable it with the <code>bgp compare-med</code> command.
MED value comparison for routes within and outside of an AS	Enable MED value comparison for IBGP and EBGp peers with the <code>bgp always-compare-med</code> command.	You must first enable IBGP MED value comparison with the <code>bgp compare-med</code> command. Then, execute the <code>bgp always-compare-med</code> command to include EBGp routes.
neighbor soft-reconfiguration inbound	Disabled by default.	Enabled by default. This command cannot be disabled in the NX-IS software.
synchronization	Enabled by default.	Disabled by default.

Feature	Cisco IOS Implementation	NX-IS Software Implementation
<code>neighbor update-source</code>	Router uses primary IP address when trying to establish a session with a neighbor and multiple IP addresses are configured on the same Ethernet port. Router switches to secondary IP address after multiple failures.	Router does not bring up a BGP session when command is set on a shut down interface.  Router uses the first configured IP interface and does not switch over to any other IP interfaces.
<code>neighbor maximum-prefix</code>	Terminates BGP session and sets neighbor state to <i>idle</i> . Neighbor remains idle until a <code>clear ip bgp</code> command is issued.	Terminates BGP session and sets neighbor state to <i>active</i> . Neighbors can become established without issuing a <code>clear ip bgp</code> command.
<code>permit-internal-into-ibgp</code>	Not supported.	Because the NX denies all routes by default, this command allows you to permit internal announcement of IGP routes.
<code>show ip bgp neighbor</code>	Not supported for internal neighbors.	Supported for internal neighbors.

## Route Filter Configuration

Route filtering controls the flow of information through a router by allowing an administrator to permit or deny certain routes. It defines what comes in and what goes out of the router. Inbound filtering controls routing updates coming into the local router. Outbound filtering controls which routes the local router advertises.

Policies are the rules an administrator uses to determine which routes are acceptable to the local router. Route filtering is an implementation mechanism for a specific policy. In other words, you can configure route filtering to select a subset from some set of routes, and take an action on that subset. That action includes permitting or denying the route's inbound or outbound access, and, optionally, manipulating the route's attributes.

Route filtering is done primarily within BGP, but is also used by interior gateway protocols (IGPs). All commands and technology in this chapter applies to BGP. Much, but not all, applies to IGPs as well. See [“Other Protocols” on page 19-11](#) for information specific to OSPF and IS-IS.

Route filtering “validates” the routes in a BGP update so that only those routes desired by the network administrator are installed in the routing information base (RIB). It also is used to control which routes are redistributed into and out of BGP. Route filtering differs from packet filtering in that it controls the flow of information by controlling what is installed in the routing table. Packet filtering, which can also be part of an administrator's policy, permits or denies access to individual data packets based on identifying information in the IP header. (See [Chapter 12, “Access List Configuration”](#) for more information on packet filtering.)

## Technology Concepts

Basic to understanding route filtering are:

- The filtering process
- Inbound vs. outbound control
- Matching and changing criteria
- Route maps
- Access, filter, and distribute lists
- Peer groups and filters

## Understanding the Filtering Process

The local router follows a three-step process before accepting or advertising a route. The process involves applying filtering rules to identify the route, prescribing an action of permitting or denying the route, and optionally, manipulating the route's attributes. Note that while the process is described for a single route, it is each route, individually, in a routing update that the local router is evaluating. The process, in more detail, is described in the following steps.

- ▶ By default, the NX64000 system denies all routes into or out of the NX64000 router. A route must first pass the decision criteria defined by the applied routing policies before it is installed in the RIB.
- 1. The route is compared to the first of the filtering rules established for the local router. The rule is a statement of matching criteria, based on IP address, AS information, or various attributes. If the route matches, it moves to step 2. If the route does not match, it is compared to the next filtering rule, and the next, until either a match is made or the rules are exhausted. As soon as a match is made, the process moves to step 2. If a match is never made, the route is discarded.
- 2. Once the route has matched one of the filter rules, an action is taken on it. Included in the description of the rule is an action of either permitting or denying the route. If the route is permitted, it moves to step 3. If the route is denied, it is discarded.
  - ▶ When a route is received, it is tested, in succession, against each rule in the list until a match is found. Once the route is identified, the process moves to the next step. Therefore, the order of the rules in the list is quite important. For example, a sweeping “deny all” statement at the beginning of the list prohibits you from permitting specific routes with subsequent filter rules. If no match is found, the “deny all” rule applies.
- 3. Once the route is permitted it can, optionally, have one or some of its attributes changed. If the rule does not include an instruction to “set” an attribute, the route is either written to the BGP table (inbound routes) or forwarded (outbound routes). If the rule does include a set command, the specified attribute is changed and the route is processed.

## Filtering Inbound and Outbound Updates

Route filtering can be applied to either inbound or outbound updates. With inbound filtering, the router is checking updates received from peers, and only installing those routes that pass the filter rules. With outbound filtering, the router is only advertising those routes that pass the filter rules. The most common basis for filtering is on an IP address/mask combination or on an AS path or list of paths.

### Filtering on IP Address

The IP address and mask combination defines a route's Network Layer Reachability Information (NLRI). This is the prefix, or range of prefixes, that identify the route. By filtering on an address/mask combination you can identify routes from a specific router or interface.

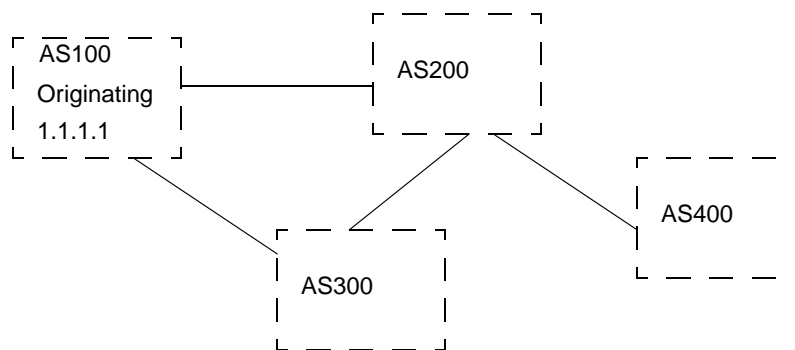
The NLRI is a variable-length field that is contained in a BGP UPDATE message. Multiple prefixes can be listed in this field. However, because the UPDATE message also contains the Path Attributes field, all prefixes in the NLRI field must share common path attributes. In that way, a single UPDATE message can contain all prefixes with common attributes, thereby reducing processing overhead.

When filtering on the IP address, you can use one of the following commands:

- `neighbor distribute-list`
- `match ip address`

## Filtering on AS Path

Filtering on as AS path allows you to identify a route based on, for example, its originating AS or a particular path through the network. The AS path is comprised of the autonomous system numbers of each AS the prefix passed through on its way to the local router. As a prefix passes through an AS, the AS number (ASN) is appended to the path list. In this way, BGP can prevent routing loops by not installing a route that includes its origin AS as part of the path. **Figure 19-1** illustrates this process.



**Figure 19-1. AS Path of a Prefix through the Network**

In the figure above, AS400 learns about 1.1.1.1 from two sources. PathA is comprised of AS100, AS200, and AS400. PathB contains AS100, AS300, AS200, and AS400. Assuming nothing filters out the update containing pathA's route, that would, most likely, be the preferable one.

To change a route's preference, you can prepend an AS path string to the AS-PATH attribute. By adding ASs to the list, the route becomes less desirable. If you want AS400 to use the pathB, you can define a match clause for pathA that would cause the additional ASNs to be added to the attribute.

The figure also illustrates the prevention of routing loops. If AS100 advertises itself as originating 1.1.1.1 to AS200, AS200 may then advertise it to AS300 and AS400. AS300 then lists the path for prefix 1.1.1.1 as AS100, AS200, and appends AS300 to the list. It would not advertise it back to AS200, but it may pass it to AS100. When AS100 sees itself listed in the path, it would know not to install that path since it would not be a preferable route.

When filtering on an AS path list, you can use one of the following commands:

- `neighbor filter-list`
- `match as-path`

- set as-path

## Using Regular Expressions

Regular expressions are tools that provide pattern matching for use with AS path lists. Used in conjunction with AS path lists, they allow you to identify single or multiple ASs in a route against which you can do inbound or outbound filtering.

Regular expressions can do pattern matching on both regular alphanumeric characters and special characters. Special characters are those keyboard symbols that have meaning or cause action within the regular expression tool. You can, however, match on special characters by preceding the character with a backslash (\). For example, to match on “expensive\$”, enter “expensive\\$”.

The examples in this section use alphanumeric characters to illustrate the expression. In actual use, because regular expressions are used to represent BGP AS paths, only numerics are accepted. All pattern matching is case-sensitive. When you enter a regular expression at the command line, quotation marks are optional. For example, a list within quotation marks, which identifies a path that passes through all listed ASs, such as:

```
ip as-path access-list 10 permit "300 400 500"
```

is recorded in the running configuration file as:

```
ip as-path access-list 10 permit 300_400_500
```

**Table 19-1** lists the special characters the NX-IS software supports for use with regular expressions. **Table 19-2** provides examples of each special character.

**Table 19-1. Regular Expression Special Characters**

Name	Symbol	Description
period	.	Match any single character.
asterisk	*	Match the preceding character zero or more times.
plus sign	+	Match the preceding character one or more times.
vertical bar		Match the preceding or following character (a logical OR).
caret	^	Match the beginning of a line (i.e., only the networks advertised from the neighbor identified immediately after the caret.)
dollar sign	\$	Match the end of a line (i.e., only the networks originated by the neighbor identified immediately after the dollar sign).

**Table 19-1. Regular Expression Special Characters**

Name	Symbol	Description
underscore	_	Match one of the following: <ul style="list-style-type: none"><li>• comma ,</li><li>• left brace {</li><li>• right brace }</li><li>• left parenthesis (</li><li>• right parenthesis )</li><li>• beginning of line</li><li>• end of line</li><li>• space</li></ul>
brackets	[]	Match on characters listed between the brackets (any single value in a range).
hyphen	-	Specifies that the characters specified represent a range.

**Table 19-2. Examples of Regular Expression Special Characters**

Symbol	Example	Explanation
.	sn.p 12.5	Matches a single character, such as “i” or “a” in snip or snap. Does not match multiple characters found in snoop. Matches AS numbers 1235 or 1245, but not 12345.
*	r.*r	Match an occurrence of “r” followed by any number of any characters followed by another “r.” These could include router, roar, rr, and rizzmatizzer.
+	3+	Matches 3, 333, 33333.
^	^router ^333	Matches “router A” but does not match “engineering owns router A.” Matches AS path list “333 444 555” but not “33 44 55” (an AS_PATH starting with 333)
\$	router\$ 1234\$	Matches “marketing’s router” but does not match “marketing and engineering routers.” Matches AS path list “12 34 1234” but not “1234 34 12” (an AS path ending with 1234).

**Table 19-2. Examples of Regular Expression Special Characters**

Symbol	Example	Explanation
	105 107	Matches any route with a 105 or 107 anywhere in the AS path.
_	_4_ _4292_100_	Matches only networks that have passed through AS 4. Matches any route that has passed through an AS 100-to-AS 4292 link.
[]	b[ai]t [0-9] [a-zA-Z] [^123a-c]	Match bat or bit, but not bet or bait. To indicate a range, use the hyphen. Match any number zero through nine. Match any upper- or lower-case letter. To exclude characters from the match, use the caret as the first symbol inside the bracket. Matches any characters except 1, 2, 3, a, b, c.
-	a-gH-T	Match lowercase letters a through g and uppercase letters H through T.
<b>Examples of Regular Expressions with Multiple Special Characters</b>		
Multiple	^4_[0-9]*\$	Matches networks originated by all directly attached ASs of AS 4.
Multiple	.*	Matches any networks.
Multiple	^\$	Matches only the local AS.
Multiple	^64111\$	Matches any route with only AS 64111 in the path.
Multiple	_.77._	Matches an AS path that includes one or more of AS x77x, where x is any single numerical character from 0 to 9.
Multiple	_6+\$	Matches only networks originated by AS 6, 66, 666, 6666 (an AS path ending with an AS number comprised of all 6s).
Multiple	(_200_)+	Matches any route that matches AS 200 or AS 200 with path stuffing.
Multiple	(91)+	Matches any route with at least one 91 somewhere in the AS path.
Multiple	[92]+	Matches any route with at least one 9 or one 2 somewhere in the AS path.



## Matching Criteria and Setting Attributes

Route filtering uses specific command types to select criteria (match commands) and change attributes (set commands) for a given route. These commands are stored in route maps, which serve as instruction sheets for route filtering.

### Using Match Commands

Match commands, used in route maps, specify the criteria for an update to match. A route must be permitted by all match statements in the route map in order to remain a candidate for redistribution. Consider it as a logical AND between match conditions. In the following example, a route must match one of the AS path lists in IP AS-path access list 5 and one of the IP addresses in access list 10 before the next hop can be set to 1.1.1.1.

```
nx1(config)# route-map LEARNHOW permit 100
nx1(config-route-map)# match as-path 5
nx1(config-route-map)# match ip address 10
nx1(config-route-map)# set next-hop 1.1.1.1
nx1(config-route-map)# exit
```

However, a match statement can contain multiple references, in which case a route need only match one reference to be a candidate for redistribution. Consider it as a logical OR. In the following example, a route must match one of the AS path lists in IP AS-path access list 5 and match one of the IP addresses in either access list 10, 11, or 12 before the next hop can be set to 1.1.1.1.

```
nx1(config)# route-map LEARNHOW permit 100
nx1(config-route-map)# match as-path 5
nx1(config-route-map)# match ip address 10 11 12
nx1(config-route-map)# set next-hop 1.1.1.1
nx1(config-route-map)# exit
```

### Using Set Commands

It is only if a route meets the match criteria that you can change its attributes. Set commands specify the change to be made to an attribute of a route when the criteria is met. While routes are compared against match commands until a single, first match is made, the route can have multiple set commands applied against it (several attributes changed).

By changing attributes, you affect the decision process. For example, you can selectively set the next hop for specific routes, thereby redirecting them from the otherwise expected path. Or, you can set the community attribute so that all routes matching the criteria are members of the same community.

## Understanding Route Maps

Route maps are used to control access of routes and modify routing information, as well as to define the rules for route redistribution. They are sets of rules that are used to identify packets so that you can direct them as you choose. For example, an ISP can configure “packets belonging to group A can go here, but group B is not allowed,” to define customer routes vs. peer routes. In essence, route maps are the instruction sheets on which the rules are stored. They are individually applied to neighbors to permit or deny access to incoming and outgoing updates.

An alternate method of applying the same filters is through use of filter lists and distribute lists, described in the next section, [“Using Lists to Identify Match Criteria.”](#) However, if you want to include attribute manipulation in your instructions, you must use route maps to do so. Filter lists and distribute lists can only be used to identify match criteria.

- If you are not including attribute manipulation in your instructions, system response is faster when applying filter lists and distribute lists instead of route maps.

You create a route map with the `route-map` command. Supply a map name, and optionally, a permit or deny statement and a sequence number. If you do not specify whether to permit or deny the routes matching the criteria in that map, the routes are permitted by default. The sequence number allows you to maintain multiple route maps with the same name. The number indicates the position of that version of the route map. Each version, or instance, can contain entirely different match criteria and set actions.

Route maps work in conjunction with access lists and AS-path lists to set up filters to permit or deny routes. The following example shows the relationship between route maps and their associated lists:

```
nx1# configure terminal
nx1(config)# route-map LEARNHOW permit 100
nx1(config-route-map)# match as-path 5
nx1(config-route-map)# match ip address 10
nx1(config-route-map)# exit
nx1(config)# ip as-path access-list 5 permit 200
nx1(config)# access-list 10 permit 2.0.0.0
nx1(config)# access-list 10 permit 3.0.0.0
```

In the example, you enter the `route-map` command with the map name `LEARNHOW`, a statement to permit, and a sequence number of 100. When you configure a route map, you enter route map configuration mode. There are two match statements in the route map.

- Match on AS paths. The list of ASs to match on is contained in AS-path access list 5. That list instructs that any pattern containing AS 200 is a match.
- Match on IP addresses. The list of addresses to match is contained in access list 10. That list instructs matching on IP addresses 2.0.0.0 and 3.0.0.0.

You apply a route map to updates to or from a neighbor using the `neighbor route-map` command. The command identifies the neighbor by either IP address or peer group name, specifies which route map to apply, and defines whether to apply the map to incoming updates from the neighbor or outgoing updates to the neighbor.

```
nx1# configure terminal
nx1(config)# router bgp 300
nx1(router-bgp)# neighbor 1.1.1.1 remote-as 600
nx1(router-bgp)# neighbor 1.1.1.1 route-map LEARNHOW in
```

In the example, the local router applies a route map named `LEARNHOW` to incoming updates from neighbor 1.1.1.1.

You apply a route map to an IGP using the `redistribute` command. In the following example, the system filters BGP routes through the route-map `LEARNHOW` before injecting them into OSPF.

```
nx1(config)# router ospf
```

```
nx1(config-router)# redistribute bgp 100 route-map LEARNHOW
```

## Using Lists to Identify Match Criteria

There are two general categories of lists used in route filtering. One is access lists, which contains the criteria by which some subset of routes is defined, i.e., it's a set of filters. These sets of rules filter on IP address and AS-path lists and permit or deny redistribution. Until you apply an access list to an element, however, it is without effect. You can apply an access list in a route map, or, within the second category of lists.

The other type of lists refer to one of the access list types. Filter lists use AS-path lists as their set of rules, and distribute lists use IP address lists. These types of lists are used from within the `router-bgp` prompt level, and are therefore unavailable to OSPF and IS-IS. While they apply the same two access lists mentioned above to updates for filtering, they do not provide any mechanism for changing attributes if the match criteria is met. The following sections described each type.

### Matching Using IP Address Access Lists

For purposes of route filtering, IP address access lists, referred to simply as access lists, identify source addresses as match criteria. Access lists can be either standard (numbered one through 99) or extended (numbered 100 through 199). Using the standard access list, you can identify a route based on its source address using a prefix/mask combination. With an extended access list, you can identify a range of networks to filter on. (Extended access lists permit packet (not route) filtering on additional criteria as well. See [Chapter 12, "Access List Configuration"](#) for more information about access lists.

An access list is identified by number, and each rule is added individually using the `access-list` command from the CLI. Rules are entered into the access list in the order that you entered them from the CLI. Rules are evaluated sequentially until a match is found. For this reason, order entry can be very important. If no match is found, the default final rule of an access list is a "deny all" statement.

### Matching Using AS-Path Access Lists

AS-path access lists identify routes based on their autonomous system paths. AS numbers are added individually to a list using the `ip as-path access-list` command. AS numbers are appended to a route's AS\_Path attribute, allowing filtering a segment of or an entire AS path list. (See ["Filtering on AS Path" on page 19-3](#) for more information.)

You can enter multiple autonomous system number for each filter rule, either individually or using regular expressions. (See ["Using Regular Expressions" on page 19-4](#) for more information on this pattern matching tool.) The following example individually adds two rules to AS-path access list number 99. The first denies any route that travelled through AS 200, 400, 600, and then 800. The second permits routes that travelled through AS 100 or 300 or 500.

```
nx1# configure terminal
nx1(config)# ip as-path access-list 99 deny 200 400 600 800
nx1(config)# ip as-path access-list 99 permit 100
nx1(config)# ip as-path access-list 99 permit 300
nx1(config)# ip as-path access-list 99 permit 500
```

The following example uses regular expressions to add two rules to AS-path access list number 99. These rules configure to deny any AS path starting with 400 and followed by 655 and permit everything else.

```
nx1# configure terminal
nx1(config)# ip as-path access-list 50 deny ^400 655
nx1(config)# ip as-path access-list 50 permit .*
```

## Matching Using Distribute Lists

When your filtering criteria is simply to identify routes based on IP address and permit or deny them, you can use the `neighbor distribute-list` command. The command, executed from the `router-bgp` prompt level, applies an IP prefix-based access list to a route exchange with a BGP neighbor. The distribute list calls an IP access list for route matching, but does not allow you to change attributes (use `set` commands) on those routes. The following example shows the relationship between the two list types. For updates coming from neighbor 1.1.1.1 in AS 100, the local router applies access list 99. This list denies any routes from network 4.0.0.0/8, and permits everything else.

```
nx1# configure terminal
nx1(config)# access-list 99 deny 4.0.0.0 0.255.255.255
nx1(config)# access-list 99 permit any
!
!
nx1(config)# router bgp 900
nx1(router-bgp)# neighbor 1.1.1.1 remote-as 100
nx1(router-bgp)# neighbor 1.1.1.1 distribute-list 99 in
```

## Matching Using Filter Lists

To permit or deny routes based on AS path information, and forego attribute manipulation, use the `neighbor filter-list` command (executed from the `router-bgp` prompt level). This command applies an AS-path access list to route exchanges with BGP neighbors. While you cannot change attributes with `set` commands, you can assign administrative weights to routes matching the AS path list criteria. This weight attribute can then be used to favor or disfavor routes for the selection process. The metric (weight) that you assign is used only by the local router (it is not passed to any other router), and applies to inbound routes only. In the following example, the local router assigns a metric of 55 to routes received from neighbor 1.1.1.1 which passed through AS 100 and 200. However, when it advertises these routes, it does not include the metric information.

```
nx1# configure terminal
nx1(config)# ip as-path access-list 99 permit 100 200
nx1(config)# router bgp 900
nx1(router-bgp)# neighbor 1.1.1.1 remote-as 100
nx1(router-bgp)# neighbor 1.1.1.1 filter-list 99 weight 55
```

## Understanding Order of Precedence

The NX64000 system applies only a single set of filter rules to a route. If a configuration file contains multiple sets of rules, the system uses the following order of precedence to set match criteria for filters:

- Route maps
- Filter lists

- Distribution lists

If multiple route maps with the same name exist, they are applied in order according to their sequence number. Multiple route maps with different names cannot be assigned to a neighbor. Each assignment of a route map overwrites any previous assignment.

The only exception to this is when using peer groups. An individual peer configuration always overrides peer group configuration. For example, peer group GALAXY filters advertisements using route-map ROCKET. A member of GALAXY, neighbor 1.1.1.1, is configured to filter addresses using distribute list 99. Although route maps take precedence over distribute lists, because it is an individual configuration, the distribute list is used. See [“Overriding Peer Group Settings” on page 19-18](#) for an example.

## Peer Groups and Filters

A peer group is a logical association of neighbors that share the same configuration attributes, for example, route filtering policies. With a peer group you can apply multiple attributes to multiple routers “in batch,” as opposed to individually configuring each with the same settings. In the case of route filtering, you can assign routers to a peer group and then apply a defined list or route-map to the entire group.

### Overriding Settings

Even after you have assigned a router to a peer group and applied, for example, a route map to that group, you can still individually override the instructions for incoming updates described in that route map. You cannot, however, override outgoing updates. The reason for this is that members of a peer group are supposed to look the same to the outside world. Therefore their outgoing update route filtering policies (those that control what they advertise) should be the same. On the other hand, each member can be configured to accept routes as the administrator sees fit. Because each still only advertises those routes matching the group policy, the “world view” is consistent.

For example, routerA is a member of a peer group named KEEPOUT. All members of KEEPOUT use the filter rules described in route-map PRIVATE, which denies updates from routerX (2.0.0.0) and routerZ (3.0.0.0). The configuration for routerA overrides the route map and permits receipt of routes from routerZ.

## Other Protocols

While EGP route filtering is important both internally and externally (i.e., both from other BGP neighbors as well as from other protocols), IGP route filtering exists mainly for proper interaction with other protocols. For example, the metric types of each protocol are different and incompatible, and so route filters must act as translators from one protocol to another.

Because OSPF and IS-IS require synchronized databases to function properly, you cannot configure route filtering between them. You can redistribute routes from an IGP into an EGP. The EGP then applies some route filtering policy, for example, aggregation of prefixes, and advertises the routes to its peers. (This is how routers know how to reach destinations inside another AS.)



While it is theoretically possible to redistribute BGP into OSPF or IS-IS, it is commonly strongly discouraged.

IGPs (other than IBGP) are not meant to deal with the number of routes that BGP would deliver, especially if you're carrying full Internet routes. This causes a huge CPU burden and possibility of error. For example, if a route starts flapping on an OSPF router, the Dykstra algorithm will rerun every route on every OSPF router's CPU. Also, a large amount of additional memory would be required to carry 100,000+ routes in both BGP and the IGP on the redistributing router(s). Instead of redistributing BGP, you can either run IBGP on all your routers or advertise defaults through the OSPF or IS-IS.

IGPs use route maps and access lists to configure filters. The filter is then applied with the `redistribute` command, which takes the route-map name as an optional argument, and links the map to a given protocol. For example, within IS-IS you can use a route map to force redistribution of routes into level-1 (ordinarily redistribution goes only into level-2).

## Route Filtering Configuration

The following table lists all the route filtering commands supported by the NX64000 switch/router. Each command listing indicates the general functions for which you would use the command. Some commands are included and described in the configuration examples that follow. All commands listed are documented in the *NX64000 Command Reference*.

**Table 19-3. Route Filtering Command Usage**

Command	BGP-only	Filtering on IP address	Filtering on AS Path	Setting Filters	Applying Filters	Notes
<code>match as-path</code>	✓		✓	✓		
<code>match community-list</code>	✓			✓		
<code>match interface</code>				✓		Implemented for redistribution of IGP-to-BGP only
<code>match ip-address</code>		✓		✓		
<code>match ip next-hop</code>		✓		✓		Applies only to outbound filters
<code>match ip route-source</code>		✓		✓		Applies only to outbound filters
<code>match metric</code>				✓		Implemented for redistribution of IGP-to-BGP only

**Table 19-3. Route Filtering Command Usage (Continued)**

Command	BGP-only	Filtering on IP address	Filtering on AS Path	Setting Filters	Applying Filters	Notes
match route-type				✓		
match tag				✓		Implemented for redistribution of IGP-to-BGP only
neighbor distribute-list	✓	✓			✓	
neighbor filter-list	✓		✓		✓	
neighbor route-map					✓	
route-map					✓	
set as-path	✓			✓		Applies only to outbound filters
set community	✓			✓		
set dampening	✓			✓		
set ip next-hop				✓		
set level				✓		
set local-preference	✓			✓		
set metric				✓		
set metric-type				✓		
set next-hop				✓		
set origin	✓			✓		
set tag				✓		
set weight	✓			✓		
show route-map						

**Table 19-3. Route Filtering Command Usage (Continued)**

Command	BGP-only	Filtering on IP address	Filtering on AS Path	Setting Filters	Applying Filters	Notes
Filter-related commands found in other chapters						
access-list Internet Protocol (IP) Commands		✓		✓		
show access-list Internet Protocol (IP) Commands						
ip as-path access-list Border Gateway Protocol Commands	✓		✓	✓		

## Basic Route Filtering Configuration Tasks

The following sections illustrate some specific uses of route filters. Refer to the first part of this chapter for an explanation of the filtering mechanism within the NX64000 system.

In general, to configure route filters, you must define some subset of the following:

- Access lists that define IP addresses for matching against.
- IP AS path access lists that define AS paths for matching against.
- Match criteria and set (change) attribute instructions.
- Route maps for applying match and set instructions to neighbors.
- Filter lists and/or distribute lists for applying only match criteria to neighbors.
- Redistribution parameters that call a route map when using filters with an IGP.

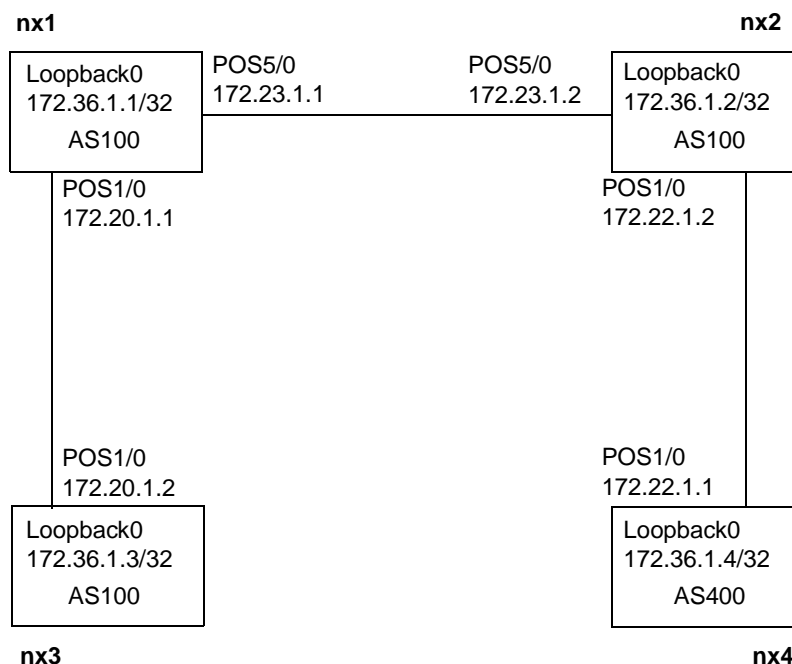
In the configuration described in the following sections, several specific filtering tasks are accomplished:

- Filtering on IP Address Using Distribute Lists
- Filtering on AS Path Using Filter Lists
- Changing Attributes to Accomplish Path Editing
- Overriding Peer Group Settings
- Redistributing IS-IS into BGP with Route Maps

► When you make changes to route filters, you must execute the `clear ip bgp` command for the new filters to take effect. This command should be run each time a filter is changed.



The figure below illustrates the BGP network as configured in this guide:



**Figure 19-2. BGP Network Example for Route Filtering**

## Filtering on IP Address Using Distribute Lists

When you do not need to change attributes of a route, the most efficient method of filtering on IP address is to use distribute lists. An inbound distribute list defines which routes are to be accepted from the specified neighbor and installed in the routing table. Outbound defines which routes can be advertised to the specified neighbor.

In the example below, NX1 uses access list 2 for inbound filtering and access list 3 for outbound filtering to/from NX4. The inbound filter permits routes from networks 172.48.1.0, 172.48.2.0, 172.48.3.0, 172.22.1.0, and 172.22.2.0, and denies all others. (There is an explicit deny at the end of all access lists.) The outbound filter permits all routes to be advertised:

```

nx1# configure terminal
nx1(config)# access-list 2 permit 172.48.1.0 0.0.0.255
nx1(config)# access-list 2 permit 172.48.2.0 0.0.0.255
nx1(config)# access-list 2 permit 172.48.3.0 0.0.0.255
nx1(config)# access-list 2 permit 172.22.1.0 0.0.0.255
nx1(config)# access-list 2 permit 172.22.2.0 0.0.0.255

nx1(config)# access-list 3 permit any

nx1(config)# router-bgp 100
nx1(router-bgp)# neighbor 172.36.1.4 remote-as 400
nx1(router-bgp)# neighbor 172.36.1.4 distribute-list 2 in
nx1(router-bgp)# neighbor 172.36.1.4 distribute-list 3 out
nx1(router-bgp)# exit
  
```

## Filtering on AS Path Using Filter Lists

When filtering on AS paths, it is often helpful to use regular expressions to simplify the filtering rules. Use the `show ip bgp regexp` command to help determine the definition of the regular expression. First view a display of what a regular expression will match, and then set filter rules accordingly. For example, first list routes advertised from a neighbor in AS 400 which AS 400 learned from AS 700. Command output is similar to the following:

```
nx1# show ip bgp regexp _400_700_
BGP table version is 88523, local router ID is 10.0.100.14
Status codes: s suppressed, d damped, * valid, > best, i - internal
               x unreachable next hop
               r administratively rejected
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network      Next Hop      Metric  LocPrf  Weight    Path
*  1.0.0.0/24      172.0.0.1        100                0      400 700 20 2 10 200 64000 e
>
*  1.0.1.0/24      172.0.0.1        100                0      400 700 20 2 90 300 e
>
*  1.0.2.0/24      172.0.0.1        100                0      400 700 1000 64000 e
>
*  2.0.0.0/24      172.0.0.1        100                0      400 700 2000 79 555 222 e
>
*  2.0.1.0/24      172.0.0.1        100                0      400 700 2000 1993 1997 e
>
*  2.0.2.0/24      172.0.0.1        100                0      400 700 4 2 59 e
>
```

If these are the routes that you want to permit, you can then use the regular expression to define your filter list. For example, allow any route that passed through an AS 400-to-AS 700 link:

```
nx1# configure terminal
nx1(config)# ip as-path access-list 222 permit _400_700_
nx1(config)# ip as-path access-list 222 deny any

nx1(config)# ip as-path access-list 333 permit any

nx1(config)# router-bgp 100
nx1(router-bgp)# neighbor 172.36.1.2 remote-as 100
nx1(router-bgp)# neighbor 172.36.1.2 filter-list 222 in
nx1(router-bgp)# neighbor 172.36.1.2 filter-list 333 out
nx1(router-bgp)# exit
```

You can verify the configuration of your AS-path access list with the `show ip as-path access-list` command:

```
nx1# show ip as-path-access-list
      AS path access list 222
        permit _400_700_

      AS path access list 333
        permit any
```

## Changing Attributes to Accomplish Path Editing

To change attributes of a route, you must use a route map. Within the map, you select match criteria, and those entries matching the criteria can then be changed. By having multiple instances of a route map, you can change an attribute to one value for a specific address and a different value for other address(es). For example, instance 10 of route map SETWEIGHT changes the weight of 172.36.1.1 to 33, while all other addresses have their weight set to 22. (Higher weight means higher preference.)

```
nx2# configure terminal
nx2(config)# access-list 100 permit 172.36.1.0 0.0.0.255
nx2(config)# route-map SETWEIGHT permit 10
nx2(config-route-map)# match ip address 100
nx2(config-route-map)# set weight 33
nx2(config-route-map)# exit
nx2(config)# route-map SETWEIGHT permit 20
nx2(config-route-map)# set weight 22
```

Another example of changing attributes would be to add ASs to the path list so that a neighbor (who judges based on path length) finds the modified route less desirable. Use the `show ip bgp` command to verify that the AS specified is added to the path list. The following example adds AS 600 AND 900 to routes traveling through AS 400:

```
nx3# configure terminal
nx3(config)# ip as-path access-list 5 permit 400
nx3(config)# route-map SETASPATH
nx3(config-route-map)# match as-path 5
nx3(config-route-map)# set as-path prepend 600 900
```

You can also set multiple attributes in a single map. For routes with AS path 100 in their path list, the following example sets the local preference to 333, sets weight to 444, increases the metric by 55, and sets the origin code to internal to the receiving AS:

```
nx3# configure terminal
nx3(config)# ip as-path access-list 5 permit 100
nx3(config)# route-map SETMANY
nx3(config-route-map)# match as-path 5
nx3(config-route-map)# set local-preference 333
nx3(config-route-map)# set weight 444
nx3(config-route-map)# set metric +55
nx3(config-route-map)# set origin igp
```

## Overriding Peer Group Settings

To override a peer-group route filtering policy, assign a set of rules with a specific IP address. While 1.1.1.1 may be a member of peer group MYPEER, you can assign different inbound filtering criteria by specifying the address. In the following example, NX1 assigns IBGP neighbor 172.36.1.2 to peer-group MYPEER and filters outbound updates through route-map PEEROUT. NX1 configures members of MYPEER to filter inbound updates through AS-path access list 10, except neighbor 172.36.1.2 overrides that setting and filters inbound updates through AS-path access list 20:

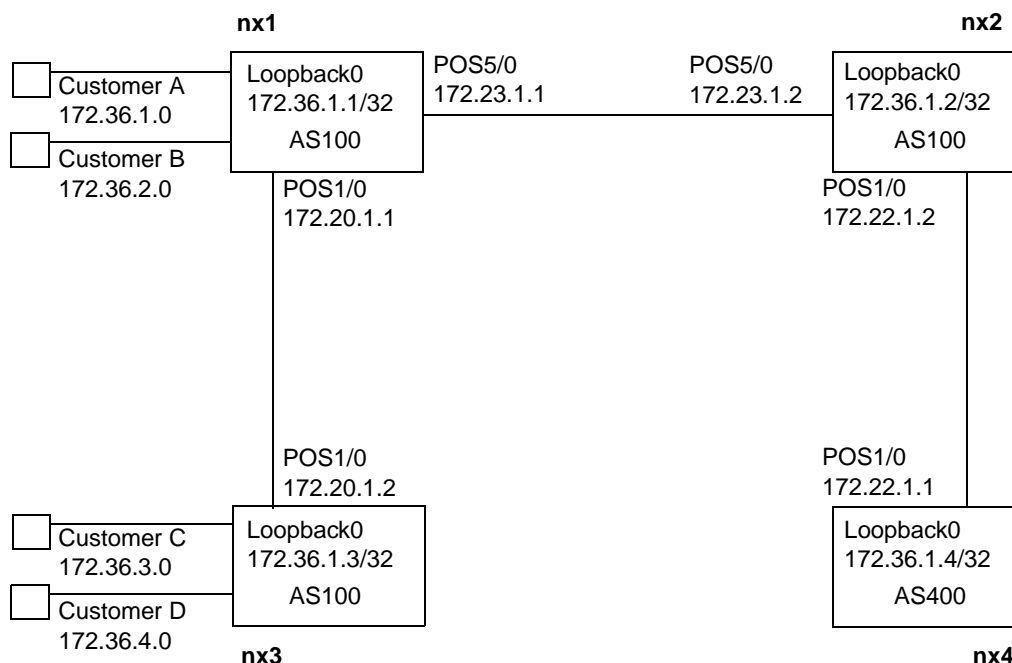
```
nx1(router-bgp)# neighbor 172.36.1.2 peer-group MYPEER
nx1(router-bgp)# neighbor MYPEER route-map PEEROUT out
nx1(router-bgp)# neighbor MYPEER filter-list 10 in
nx1(router-bgp)# neighbor 172.36.1.2 filter-list 20 in
```

## Redistributing OSPF into BGP with Route Maps

It is common to use route maps to redistribute IGP routes into BGP. For example, you may use OSPF for your internal network of publicly available addresses, and use BGP to access the outside. For the internal networks to be advertised, you must either redistribute OSPF into BGP or set up a static route on the autonomous system boundary router (ASBR) to advertise the internal network. (Internal routes are only redistributed to external peers since the IGP should take care of transmitting the routes internally.)

In the following example, NX1 and NX3 are running OSPF in addition to BGP. Each of the networks off of those routers (only the interface by which they are reached is illustrated) is also running OSPF, and all networks are in OSPF area 0. Because BGP will advertise routes to its external routers, you want to prevent most of the networks from being advertised to BGP. You do so by instructing OSPF to redistribute routes to BGP, but applying a route map containing filters that block all but networks 172.36.1.0 and 172.36.3.0.

The following figure illustrates the addition of OSPF to the BGP network:



**Figure 19-3. OSPF Addition to BGP Example Network**

The following configuration will redistribute only OSPF networks 172.36.1.0 and 172.36.3.0 to NX4.

NX2:

```
nx2# configure terminal
nx2(config)# access-list 1 permit 172.36.1.0 0.0.0.255
nx2(config)# access-list 1 permit 172.36.3.0 0.0.0.255
nx2(config)# access-list 1 deny any
nx2(config)# route-map LEARNOSPF
nx2(config-route-map)# match ip address 1
nx2(config-route-map)# exit
nx2(config)# router bgp 100
nx2(config-bgp)# neighbor 172.36.1.4 remote-as 400
nx2(config-bgp)# redistribute ospf route-map LEARNOSPF
```

## Verifying Route Filtering Configuration

The NX-IS software provides a number of show commands that let you verify configuration of filters and their components. The following table lists the type of information you can view from each show command:

**Table 19-4. Route Filtering Commands for Verifying Configuration**

Action	Command
Display all or a specific access list and its configuration.	<code>show access-list</code>
Display all or a specific AS-path access list and its configuration.	<code>show ip as-path-access-list</code>
Display routes by matching a regular expression	<code>show ip bgp regexp</code>
Display each or a specific route map name and configuration.	<code>show route map</code>

## Interoperability Issues

If you use a Cisco 12000 GSR with the NX64000 switch/router, you should be aware of the following differences. This table does not include command syntax or valid value range differences.

Feature	Cisco IOS Implementation	NX-IS Software Implementation
Route maps for peers	Neighbors in a peer group keep their own inbound route maps regardless of whether a route map was explicitly associated with the peer group.	A neighbor configured as a peer group member takes the associated peer group settings. That is, inbound and outbound route maps override any individual associations.
set as-path	Prepends a string to a string to a BGP as-path attribute to <i>inbound and outbound</i> route maps.	Prepends a string to a string to a BGP as-path attribute to <i>only outbound</i> route maps.

## Acronyms

This appendix lists the acronyms used in the documentation.

### Acronyms

Acronym	Meaning
AAL	ATM adaptation layer
ABR	area border router <i>Also: available bit rate</i>
ACR	allowable cell rate
API	Application Programming Interface
ARP	Address Resolution Protocol
AS	autonomous system
ASBR	autonomous system border router
ASE	autonomous system external
ASP	autonomous system path
ATM	asynchronous transfer mode
Bc	committed burst size
Be	excess burst size
BGP	Border Gateway Protocol
CAC	connection access control
CDV	cell delay variation
CIDR	classless interdomain routing

Acronym	Meaning
CIR	committed information rate
CLNS	connectionless network service
CP	control processor <i>Usually called the RCP (route control processor)</i> <i>Also referred to as RCC (route control card)</i>
CRC	cyclic redundancy check
CSNP	complete sequence number packets
CSU	channel service unit
CTA	clock timing adapter <i>Usually called the STA (SONET Timing/Alarm module)</i> <i>Also referred to as TAC (Timing Adapter Controller)</i>
DCE	data communications equipment
DF	don't fragment bit
DIS	designated intermediate system
DLCI	Data Link Connection Identifier
DNS	domain name service
DR	designated router
DS	differential service
DSU	data service unit
DTE	data terminal equipment
DWDM	dense wave division multiplexing
EBGP	External Border Gateway Protocol
EGP	External Gateway Protocol
ES-IS	End System-to-Intermediate System
FCS	Frame Check Sequence
FEAC	far end alarm control
FPGA	Field Programmable Gate-Array
FRU	field replacement unit



---

Acronym	Meaning
FTP	File Transfer Protocol
HDLC	high level data link control
IARP	Inverse Address Resolution Protocol
IBGP	Internal Border Gateway Protocol
ICMP	Internet Control Message Protocol
IGMP	Internet Group Multicast Protocol
IGP	Interior Gateway Protocol
IOA	input/output adapter card
IOC	input/output card <i>Usually referred to as IOP (input/output processor)</i>
IOP	input/output processor <i>Also referred to as IOC</i>
IPCP	IP Control Protocol
IRPI	Interdomain Routing Protocol
ISDN	Integrated Services Digital Network
IS-IS	Intermediate System-to-Intermediate System
Kbps	kilobits per second
LAN	local area network
LAP	Link Access Protocol
LCP	link control protocol
LLC	logical link control
LMI	local management interface
LP	local processor
LSA	link state advertisement
LSP	link state packet <i>Also: label switched path</i>
LSR	label switched router
MAC	Media Access Control

Acronym	Meaning
Mbps	megabits per second
MBS	maximum burst size
MED	multi-exit discriminator
MIB	Management Information Base
MPLS	Multiprotocol Label Switching
MRU	Maximum Receive Unit
MSDP	Multicast Source Discovery Protocol
MTU	Maximum Transfer Unit
NAS-IP	network access server IP
NBMA	non-broadcast multiple access
NET	network entity title
NIC	network interface card
NNI	Network-to-Network Interface
NPDU	network protocol data unit
NSAP	Network Service Access Point
NSSA	not so stubby area
OAM	operation, administration, and maintenance
OL	overload bit
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PCR	peak cell rate
PDU	protocol data unit <i>Also: power distribution unit</i>
PIM	Protocol Independent Multicast
PMP	point-to-multipoint
POS	Packet over SONET (Synchronous Optical Network)
PPP	Point-to-Point Protocol

Acronym	Meaning
PSNP	partial sequence numbers protocol data unit
PVC	permanent virtual circuit
PVP	permanent virtual path
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
RCC	route control card <i>Usually referred to as RCP (route control processor)</i> <i>Also referred to as CP (control processor)</i>
RCP	route control processor <i>Also referred to as CP (control processor) and RCC (route control card)</i>
RFC	request for comments
RIP	Routing Information Protocol
RSVP	resource reservation protocol
SCR	sustained cell rate
SDH	synchronous digital hierarchy
SFC	switch fabric card
SNAP	Subnetwork Access Protocol
SNMP	Simple Network Management Protocol
SNPA	subnet point of attachment
SONET	Synchronous Optical Network
SPF	Shortest Path First
STA	SONET Timing/Alarm module <i>Also referred to as CTA (Clock Timing Adapter) and TAC (Timing Adapter Controller)</i>
TAC	Timing Adapter Controller <i>Usually referred to as STA (SONET Timing/Alarm module)</i> <i>Also referred to as CTA (Clock Timing Adapter)</i>
TCP	Transmission Control Protocol

Acronym	Meaning
TTL	Time to Live (threshold)
TOS	type of service
UDP	User Datagram Protocol
VC	virtual circuit
VCD	virtual circuit descriptor
VCI	virtual channel identifier
VP	virtual path
VPC	virtual path connection
VPI	virtual path identifier
VPN	virtual private network
WAN	wide area network
WRED	weighted random early detection

## Customer Support

To report a technical problem, you may open a trouble ticket by contacting Customer Support at Lucent Technologies as indicated below.

Mailing Address	9305 Gerwig Lane, Suite J, Columbia, MD 21046
Customer Support Web Site	<a href="http://www.lucent.com/support/">http://www.lucent.com/support/</a>
Customer Support E-mail Address	<a href="mailto:nxsupport@lucent.com">nxsupport@lucent.com</a>
Customer Support Phone Number	1-866-LUCENT8 (1-866-582-3688)
FAX	410-290-3343

## Problem Reporting Information

To expedite the troubleshooting process, please have available and provide the following information. Note any deviations from initially installed component version information.

Source Information	Description
<b>Primary Contact Name:</b>	
Location:	
Phone/Pager:	
Availability:	

---

Source Information	Description
<b>Secondary Contact Name:</b>	
Location:	
Phone/Pager:	
Availability:	
System type:	
Installed HW components / versions:	
Installed SW modules / versions:	
Affected component (SW module, HW component, etc.):	
Question/Problem:	
Severity of impact:	
Supporting documentation (configuration files, log files, topology diagrams, trace files, etc.):	

## Documentation Issues

For comments on the documentation, send email to [nxdoc@lucent.com](mailto:nxdoc@lucent.com). Please include the name of the document and the document's part number (located on the title page of the manual) with your comments.