BSR 64000 Configuration and Management Guide

MOTOROLA
intelligence everywhere™

Notice

Motorola reserves the right to revise this publication and to make changes in content from time to time without obligation on the part of Motorola to provide notification of such revision or change. Motorola provides this guide without warranty of any kind, either implied or expressed, including, but not limited to, the implied warranties of mechantability and fitness for a particular purpose. Motorola may make improvements or changes in the product(s) described in this manual at any time.

TPD-0034-01
501897-001-01

Published: October, 2002

Contents

Preface

Scope xxi

Audience xxi

Documentation Set xxi

Conventions xxii

Notes, Cautions, Warnings xxiii

Contacting Support xxiv

1 Introduction

Overview 1-1

Multiservice Support 1-1

Network Management and Control 1-2

2 Using the Command Line Interface

Overview 2-1

Using a Terminal Session to Access the BSR 2-1

Using a Telnet Session to Access the BSR 2-2

Command Mode Access 2-3

- User EXEC Mode 2-5
- Privileged EXEC Mode 2-5
- Global Configuration Mode 2-5

Interface Configuration Mode 2-6
Router Configuration Mode 2-6
Route Map Configuration Mode 2-6
Cable Spectrum Group Mode 2-7
Disabling and Resetting Features 2-8
Using Command Aliases 2-9
Obtaining Help 2-10
Context-sensitive Help 2-11
Using the Command History 2-11
Setting the Command History Buffer Size 2-11
Recalling Commands 2-12
Disabling the Command History Feature 2-12
Editing Features 2-13
Navigating the Command Line 2-13
Completing a Partial Command Name 2-14
Handling Command Lines 2-14
Deleting Entries 2-15
Scrolling Down a Line or a Screen 2-15
Transposing Characters 2-16
Controlling Case 2-16
Using Output Modifiers to Limit Show Command Output 2-16

3 Configuring the System
Overview 3-1
Initial Configuration Tasks 3-1
Gathering Network Information 3-1
Required Servers 3-2
Accessing the CLI to Set System Passwords 3-3
Setting System Passwords 3-4
Specifying a Host Name 3-6
Configuring User Login Accounts 3-6
Defining a User Name without a Password 3-7
Defining a User Name with an Unencrypted Password 3-8
Defining a User Name with an Encrypted Password 3-8

Defining a Privilege Level 3-9
Defining a Group Access Level 3-9
Verifying Your User Account Login Configuration 3-10
Configuring Interfaces 3-10
Specifying System Time Information 3-11
Configuring System Log Parameters 3-11
Configuring Logging for a Remote Syslog Server 3-12
Configuring Console Logging 3-14
Setting the Logging Buffer 3-15
Clearing the Buffer 3-16
Restricting Logging Rates and SNMP Traps 3-16
Setting the Recording Mechanism for CMTS Reports 3-17
Returning to the Default CMTS Log Reporting Configuration 3-19
Logging Your CLI Session to the Syslog Server 3-19
Sending Messages to BSR Users 3-19
Configuring Server Related Parameters 3-20
Configuring DHCP Relay 3-20
Configuring DNS 3-22
Specifying DNS Name Servers 3-22
Configuring the Domain Name 3-22
Enabling Domain Lookup and Domain List 3-23
Configuring LDAP 3-23
Configuring SNTP 3-25
Configuring UDP Broadcast Relay 3-27
Configuring FTP Access 3-29
Enabling the RADIUS Client on the BSR 3-30
Configuring the RADIUS Client for Server Communication 3-30
Configuring RADIUS Client Access 3-32
Viewing RADIUS Client Statistics 3-34
Downloading Software 3-34
Before You Download Software 3-34
Downloading Image Files to NVRAM on the SRM 3-36
Downloading Image Files to Flash Memory on the SRM 3-38
Downloading Software to All Modules 3-39

Downloading Software to a Specific Module 3-40
Specifying the System Image File Boot Location 3-42
Specifying System Information 3-42
Configuring SRM and Chassis Alias Information 3-43
Configuring SRM and Chassis Asset ID Information 3-43
Configuring SRM and Chassis Serial Number Information 3-44
Saving and Viewing Your Configuration 3-44
Reseting BSR Modules 3-45
Monitoring the System 3-45
Displaying System Processing Information 3-46
Displaying System Memory Information 3-49
Displaying the System Version Information 3-50
Displaying System Buffer Information 3-52
Gathering System Information 3-53
Viewing SRM and Chassis Alias Information 3-53
Viewing the SRM and Chassis Asset ID Information 3-53
Viewing the SRM and Chassis Serial Number Information 3-54
Viewing the Chassis Status 3-54
Displaying Module Hardware Information 3-56

4 Configuring SNMP
Overview 4-1
Configuring SNMP Server Parameters 4-1
Enabling SNMP 4-8
Configuring SNMPv3 4-8
Configuring SNMP Server Identification 4-13
Configuring System Contact Information 4-14
Configuring System Location Information 4-14
Configuring the EngineID 4-14
Configuring SNMP Access Levels 4-16
Defining a Community Name 4-16
Configuring USM and VACM Security 4-18
Configuring a Group Model 4-19
Configuring a MIB View 4-20

Associating Groups to MIB Views 4-21

Configuring an SNMP Context 4-21

Configuring Packet Size 4-22

Configuring SNMP Traps 4-23

Enabling Traps 4-23

Configuring a Trap Destination 4-23

Specifying the Destination IP Address 4-24

Specifying Specific Trap Types 4-25

Specifying SNMP Trap Versions 4-25

Restricting Trap Rates 4-26

Monitoring SNMP 4-27

5 Configuring Interfaces and TCP/IP Features

Overview 5-1

About TCP/IP Level Features 5-1

Setting IP Interface Addresses 5-2

Removing an IP Address 5-6

Configuring Auto-Negotiation on the 10/100 Ethernet Module 5-6

Configuring a Loopback Interface 5-8

Configuring Tunnels on an Interface 5-10

Configuring an Unnumbered Interface 5-13

Configuring the Address Resolution Protocol 5-14

Reverse ARP 5-15

Address Resolution Using Proxy ARP 5-16

Configuring Broadcast Addressing 5-16

Defining Broadcast Address 5-17

Configuring the MTU 5-17

Configuring Static Routes 5-18

Clearing Route Table Entries 5-20

Configuring the Internet Control Message Protocol 5-20

About IRDP 5-20

Enabling IRDP 5-21

Enabling ICMP 5-23

Tracing a Route 5-24

Managing the Router 5-25
Enabling IP Source Routing 5-25
IP Accounting 5-26
Clearing Interface Counters 5-28
Clearing IP Routes 5-28
Clearing the ARP Cache 5-29
Clearing IP Traffic 5-29
Clearing DNS Entries 5-29
Gathering TCP/IP Related Information 5-29

6 Configuring the CMTS
Overview 6-1
Initial Cable Interface Configuration Tasks 6-1
Setting the IP DHCP Relay Functions 6-2
Configuring the Cable Helper and IP Helper Addresses 6-4
Enabling Host Authorization for All CMs 6-6
Disabling Host Authorization for All CMs 6-6
Creating a Static Host Authorization Entry for a Specific CM 6-6
Deleting a Static Host Authorization Entry for a Specific CM 6-7
Enabling Host Authorization for an IP Range of CPEs 6-8
Removing Host Authorization for an IP Range of CPEs 6-8
Using DHCP Lease Query Function to Secure Cable Network 6-9
Setting ARP Parameters 6-11
Configuring a Downstream Channel 6-12
Initial Downstream Configuration Tasks 6-13
Configuring the Downstream Frequency and Modulation Rate 6-13
Enforcing the Downstream Rate Limit 6-14
Enabling the Downstream Port 6-15
Managing the Downstream Channel 6-15
Configuring the Downstream Interleave Depth 6-16
Setting the Downstream Power Level 6-17
Resetting a Downstream Port 6-17
Reserving Downstream Bandwidth 6-18
Unreserving Downstream Bandwidth 6-18

Testing RF Carrier Modulation 6-19

Configuring an Upstream Channel 6-19

Initial Upstream Configuration Tasks 6-21

Setting the Upstream Frequency 6-21

Setting the Upstream Power Level 6-22

Applying an Upstream Modulation Profile 6-24

Enforcing the Upstream CM Rate Limit 6-25

Enabling an Upstream Port 6-26

Managing the Upstream Channel 6-27

Configuring Upstream CM Registration Parameters 6-27

Adjusting for Physical Delay between Cable Interface and CMs 6-31

Activating Upstream Forward Error Correction 6-32

Activating the Scrambler on the Upstream Channel 6-33

Enabling Pre-equalization 6-33

Forcing the Fragmentation of Large Upstream Packets 6-34

Disabling an Upstream Port 6-34

Configuring the Upstream Channel Descriptor 6-35

Limiting the Number of Voice Calls on an Upstream Channel 6-35

Bundling Cable Interfaces into a Single IP Subnet 6-36

Creating a Cable Bundle 6-37

Adding a Static Arp Entry to a Cable Bundle Interface 6-40

Subneting DHCP Clients on the Cable Interface 6-40

Creating a Modulation Profile 6-41

Setting Network Parameters for Cable Modems 6-43

Enabling the CM Aging Timer 6-44

Setting the Insertion Interval for CMs 6-44

Setting the Synchronization Interval 6-45

Setting CM Authentication Parameters 6-46

Restoring Previously Defined Authentication Parameters 6-47

Denying Access to a Cable Modem 6-48

Setting the Maximum Number of Hosts 6-48

Configuring Baseline Privacy 6-49

Setting TEK Privacy 6-50

Setting Authorization Key Values 6-52

Setting QoS Parameters 6-54

- Initiating a DSA 6-54
- Initiating a DSC 6-55
- Initiating a DSD 6-56
- Viewing QoS Information 6-57
  - Displaying the Packet Classifier 6-57
  - Displaying SFID and QoS Information 6-59
  - Displaying Service Flow Statistics 6-59
  - Displaying Payload Header Suppression Entries 6-60

Implementing Spectrum Management 6-60

- Configuring a Spectrum Group 6-61
  - Creating a Spectrum Group 6-62
  - Scheduling the Availability of a Spectrum Group Band 6-64
  - Scheduling the Removal of a Spectrum Group Band 6-68
  - Configuring Spectrum Data Collection 6-70
  - Configuring Spectrum Hopping Rules 6-72
  - Configuring the Spectrum Hopping Error Threshold 6-77
  - Configuring the Spectrum Hopping Flap Threshold 6-78
  - Enabling and Disabling Spectrum Roll-back 6-79
  - Configuring the Guard Band 6-80
  - Reviewing the Spectrum Group that You Created 6-80
  - Viewing Your Spectrum Group Configuration 6-83
- Applying a Spectrum Group to an Upstream Port 6-83
- Evaluating Spectrum Management Performance 6-85
  - Showing Spectrum Data 6-85
  - Viewing Spectrum Management Configuration Changes 6-86
  - Determining the Upstream Signal to Noise Ratio 6-87
  - Determining the MIB Index ID Number of an Upstream Port 6-88
  - Viewing Spectrum Management Activity 6-89
  - Viewing Spectrum Management Hopping Actions 6-90
  - Viewing the Spectrum Management Roll-back Function 6-93

Using Flap Lists 6-95

- Setting Flap List Parameters 6-95
- Using Flap Lists to Troubleshoot CM Problems 6-98

Viewing Flap List Statistics to Identify Network Health 6-99

Interpreting Flap List Statistics 6-101

Tips for Administering Flap Lists 6-106

Managing Multicast Maps 6-107

Pinging a Cable Modem at the MAC Layer 6-108

Resetting the Cable Modem 6-109

Clearing Cable Interface Counters 6-111

Gathering DOCSIS Network Information 6-111

Displaying Cable Interface Statistics 6-112

Displaying Downstream Parameters 6-114

Viewing Downstream Port Information 6-114

Displaying Upstream Parameters 6-115

Viewing Upstream Port Information 6-116

Viewing CM Information 6-117

Displaying Modulation Profiles 6-129

Displaying BPI Configuration Settings 6-130

7 Configuring Routing Policy

Overview 7-1

Defining Route Maps 7-1

Creating a Route Map 7-1

Using Match Statements to Define Routing Conditions 7-2

Using Set Statements to Define Routing Conditions 7-4

Defining Access Lists and Groups 7-6

Configuring an IP Access List 7-6

Configuring an AS-path Access-list 7-7

Configuring an IP Access Group 7-8

Creating Community Lists 7-9

Filtering Routes 7-10

Redistributing Routes 7-12

Applying a Damping Criteria 7-13

Policy-Based Routing 7-14

Configuring a Policy-Based Route Map 7-15

Defining Match Criteria 7-15

Defining the Route 7-16
Enabling Policy-Based Routing on an Interface 7-18
Enabling Local Policy-Based Routing on a Router 7-18
Gathering Routing Policy Information 7-18

8 Configuring IP Multicast Routing
Overview 8-1
Enabling IP Multicast Routing on the BSR 8-2
Configuring PIM 8-2
About PIM 8-2
Enabling PIM 8-3
Delaying Shortest Path Tree Usage for Better Throughput 8-4
Defining the PIM Domain Border 8-5
Configuring Candidate BSRs 8-5
Configuring Candidate RPs 8-7
Modifying the PIM Router-Query Message Interval 8-8
Configuring DVMRP 8-8
About DVMRP 8-9
Configuring DVMRP Routing Information 8-9
Enabling DVMRP on the Router 8-9
Configuring the DVMRP Route Expiration Threshold 8-10
Configuring the DVMRP Route Reporting Threshold 8-10
Limiting the Number of DVMRP Routes 8-10
Setting the DVMRP Prune Lifetime Value 8-11
Configuring DVMRP on a Routing Interface 8-11
Filtering Incoming DVMRP Reports 8-12
Filtering Outgoing DVMRP Routing Reports 8-12
Distributing the Default DVMRP Network to Neighbors 8-12
Adding a Metric Offset to the DVMRP Route 8-13
Setting the DVMRP Neighbor Time-out Interval 8-14
Delaying DVRMP Reports 8-14
Setting the DVMRP Probe Interval 8-15
Rejecting a DVMRP Non-pruning Neighbor 8-15
Configuring a DVMRP Summary Address 8-15

Configuring IGMP on an Interface 8-16

About IGMP 8-16

Enabling IGMP 8-17

Controlling Access to IP Multicast Groups 8-17

Changing the IGMP Version 8-18

Modifying the IGMP Host-Query Message Interval 8-18

Specifying the IGMP Querier Time-out Interval 8-19

Changing the Maximum Query Response Time 8-19

Configuring the BSR as a Static Multicast Group Member 8-20

Managing IP Multicast Routing on the BSR 8-20

Configuring an IP Multicast Static Route 8-21

Changing the Distance for a Unicast Multicast Route 8-21

Changing the Distance for a Static Multicast Route 8-22

Clearing IP Multicast Information 8-22

Removing a DVMRP Prune 8-22

Removing a DVMRP Route 8-23

Clearing IGMP Statistics 8-23

Removing the IP Multicast Cache 8-24

Gathering IP Multicast Information 8-24

Displaying General IP Multicast Information 8-24

Displaying PIM Information 8-26

Displaying DVMRP Information 8-27

Displaying IGMP Information 8-27

Displaying Reverse Path Forwarding Information 8-28

9 Configuring RIP

Overview 9-31

About RIP 9-31

Specifications 9-32

Enabling RIP 9-33

Specifying a RIP Version 9-34

Enabling or Disabling Split Horizon 9-35

Enabling Route Summarization 9-36

Applying an Offset List 9-37

Enabling RIP Authentication 9-38
Configuring Interpacket Delay 9-38
Configuring Timers 9-39
Example 9-40
Configuring a Passive Interface for RIP 9-40
Redistributing Routes into RIP 9-41
Assigning a Default Metric Value for Redistributed Routes 9-43
Gathering RIP Information 9-44
10 Configuring IS-IS
Overview 10-1
Enabling IS-IS 10-1
Redistributing Routes into IS-IS 10-4
Assigning a Default Metric Value for Redistributed Routes 10-6
Managing IS-IS on the BSR 10-6
Specifying Router-Level Support 10-7
Forcing a Default Route or Route Map into an IS-IS Domain 10-7
Configuring the Administrative Distance for IS-IS 10-8
Configuring IS-IS Area or Domain Passwords 10-8
Summarizing IP Address Ranges 10-9
Enabling the LSP Overload Bit 10-9
Configuring a Passive Interface for IS-IS 10-9
Configuring IS-IS on an Interface 10-10
Specifying the Interface Circuit Type of an IS-IS Interface 10-10
Configuring IS-IS Link-State Cost Metrics 10-11
Setting the Advertised Hello Interval 10-11
Specifying the Advertised Hello Multiplier 10-12
Setting the Advertised CSNP Interval 10-12
Setting the LSP Interval 10-13
Setting the LSP Retransmission Interval 10-14
Setting the LSP Retransmit Throttle Interval 10-14
Setting the Designated Router Priority 10-14
Assigning a Password to an IS-IS Interface 10-15
Gathering IS-IS Information 10-16

Displaying IS-IS Database Information 10-16

Displaying the Shortest Path First Log 10-17

Displaying Connectionless Network Service Information 10-19

11 Configuring OSPF

Overview 11-1

Specifications 11-1

Enabling OSPF 11-2

Redistributing Routes into OSPF 11-3

Assigning a Default Metric Value for Redistributed Routes 11-4

Configuring OSPF Area Parameters 11-5

Configuring OSPF Area Authentication Parameters 11-5

Configuring OSPF Stub Areas 11-6

Configuring OSPF Not So Stubby Area 11-8

Configuring Route Summarization between OSPF Areas 11-9

Configuring Route Summarization into OSPF Area 11-10

Managing OSPF on the BSR 11-11

Establishing a Virtual Link 11-11

Assign a Default Route for an ASBR 11-15

Controlling OSPF Link Cost Metrics 11-16

Allowing Dynamic OSPF Virtual Links 11-16

Changing OSPF Administrative Distances 11-17

Configuring Route Calculation Timers 11-18

Configuring OSPF on an Interface 11-18

Configuring General OSPF Interface Parameters 11-19

Blocking OSPF LSA Flooding 11-22

Configuring a Passive Interface for OSPF 11-22

Forcing Router ID Choice with Loopback Interface 11-23

Gathering OSPF Information 11-24

Displaying OSPF Routing Information 11-24

Showing Network Information 11-24

Showing Border Routers 11-25

Showing Neighboring Routers 11-25

Showing Virtual Links 11-27

Displaying OSPF Interface Information 11-27
Displaying OSPF Memory Information 11-28
Displaying OSPF Database Information 11-29

12 Configuring BGP
Overview 12-1
About BGP 12-1
BGP Peers 12-4
BGP Updates 12-4
BGP Sessions 12-4
Specifications 12-5
Configuring Basic BGP Connectivity 12-5
Configuring a BGP Neighbor 12-5
Advertising Networks in an AS 12-7
Configuring Advanced BGP Connectivity 12-9
Configuring BGP Peer Groups 12-10
Configuring a Routing Domain Confederation 12-12
Configuring a Route Reflector 12-15
Configuring a Cluster-ID 12-17
Restoring Route Reflection from a Route Reflection Client 12-18
Configuring Route Flap Dampening 12-19
Global Route Flap Dampening 12-19
Policy-based Route Flap Dampening 12-20
Clearing Route Flap Dampening 12-21
Shutting Down a Neighbor or Peer Group 12-22
Enabling Message Digest 5 Authentication Between Peers 12-23
Setting the Routing Updates Interval 12-23
Enabling EBGP Multihop for Neighbor and Peer Groups 12-24
Controlling the Number of Prefixes 12-25
Configuring Next Hop Processing 12-26
Configuring Next Hop Processing 12-28
Configuring Global BGP Tasks 12-29
Resetting BGP Connections 12-30
Configuring BGP Soft Reconfiguration 12-31

Enabling and Disabling Synchronization 12-32

Configuring BGP Administrative Weights 12-34

Using a Route Map 12-36

Using an AS Path Access List 12-37

Adjusting BGP Timers 12-37

Setting the Administrative Distance for a Route 12-38

Disabling Route Summarization 12-40

Configuring Aggregate Addresses 12-42

Assigning an Interface to BGP Session 12-44

Configuring a Default Route 12-46

Configuring BGP Update Flows 12-48

Configuring BGP Path Selection Algorithm 12-49

BGP Path Selection Algorithm 12-49

Configuring the Local Preference 12-50

Configuring the Origin Attribute 12-52

Configuring the AS-path Attribute 12-54

Configuring the MED Attribute 12-55

Configuring the Community Attribute 12-58

Configuring Routing Policy 12-58

Match and Set Statements 12-59

Handling Access Lists 12-61

Configuring an Access List 12-61

Configuring an AS Path Access List 12-62

Creating a Community List 12-63

Redistributing Routes into BGP 12-69

Assigning a Default Metric Value for Redistributed Routes 12-71

Monitoring BGP 12-72

13 Configuring VRRP

Overview 13-1

About VRRP 13-1

Initial VRRP Tasks 13-1

Enabling VRRP 13-2

Creating a Virtual Router 13-2

Configuring a Virtual IP Address 13-3
Specifying Authentication String 13-3
Configuring Primary IP Address 13-4
Enabling a Virtual Router 13-5
Configuring Authentication Type 13-5
Managing VRRP on the BSR 13-6
Specifying Priority 13-7
Pre-empting a Master 13-8
Specifying Advertisement Interval 13-8
Clearing Statistic Counters 13-9
Gathering Virtual Router Information 13-10
Monitoring Critical Link State 13-10
Monitoring Virtual Router Information 13-11
Monitoring Ethernet Virtual Routers 13-12
Obtaining Summary Information 13-14

14 Configuring Packet Over SONET
Overview 14-1
About SONET/SDH 14-2
Specifications 14-2
POS Features 14-2
POS Interface Configuration Tasks 14-3
Configuring the POS Interface 14-3
Configuring PPP 14-4
Configuring the Network Clock Source for SONET 14-5
Setting the Primary BITS Network Clocking Source 14-6
Setting the Secondary BITS Network Clocking Source 14-7
Deriving the Network Clocking Source from a POS Interface 14-8
Setting Clock Recovery from the Received SONET Signal 14-9
Configuring SONET 14-10
Optionally Disabling SONET Payload Scrambling 14-11
Changing the SONET Framing Type 14-12
Changing the CRC Function on the POS Interface 14-12
Defining SONET Frame Overhead Bytes 14-13

Configuring SONET Alarms 14-15
- Setting Alarm Thresholds 14-16
- Setting Alarm Reporting 14-17
- Setting the Line Alarm Indication Signal 14-18

Changing the POS Signal Rate 14-19

Specifying the POS Loopback Mode Type 14-20

Gathering POS Network Information 14-21
- Displaying PPP Link and Statistics Information 14-21
- Displaying POS Interface Information 14-22
- Displaying Physical SONET Link and Alarm Information 14-25

Index

Preface

Scope

This document describes how to install and configure the Motorola™ Broadband Services Router™ 64000 (BSR 64000™).

Audience

This document is for use by those persons who will install and configure the BSR 64000™ product. Only trained service personnel should install, maintain, or replace the BSR 64000.

Documentation Set

The following documents comprise the BSR 64000 documentation set:

- *BSR 64000 Command Reference Guide*

  This document contains the Command Line Interface (CLI) commands for managing, configuring, and maintaining the BSR 64000.

- *BSR 64000 Configuration and Management Guide*

  This document provides the instructions and procedures for configuring and managing the BSR 64000.

- *BSR 64000 Installation Guide*

  This document describes how to install the BSR 64000 product.

- *BSR 64000 Release Notes*

  These documents provide information about features not described or incorrectly documented in the main documentation set; known problems and anomalies; product limitations; and problem resolutions.

- *BSR 64000 SNMP MIB Reference Guide*

  This document describes the Simple Network Management Protocol (SNMP) MIBs; provides information that describes standard and proprietary MIB support; describes how to walk the MIBs and how to compile and load the SNMP MIBs. It also provides task examples.

- *BSR Troubleshooting Guide*

  This document provides instructions and procedures for troubleshooting the BSR product.

- *BSR 64000 Quick Start Guide*

  This document provides basic tasks used to get the BSR 64000™ out of the box, running, connected to the network, and operational.

Conventions

This document uses the conventions in the following table:

Convention	Example	Explanation
angle brackets < >	**ping** <*ip-address*> **ping 54.89.145.71**	Arguments in italic and enclosed by angle brackets must be replaced by the text the argument represents. In the example, **54.89.345.71** replaces <*ip-address*>. When entering the argument, do not type the angle brackets.
bar brackets [ ]	**disable** [*level*]	Bar brackets enclose optional arguments. The example indicates you can use the **disable** command with or without specifying a *level*. Some commands accept more than one optional argument. When entering the argument, do not type the bar brackets.

Convention	Example	Explanation
bold text	**cable relay-agent-option**	Boldface text must be typed exactly as it appears.
brace brackets {}	**page** {**on** | **off**}	Brace brackets enclose required text. The example indicates you must enter either **on** or **off** after **page**. The system accepts the command with only one of the parameters. When entering the text, do not type the brace brackets.
italic text	**boot system** <*filename*>	Italic type indicates variables for which you supply values in command syntax descriptions. It also indicates file names, directory names, document titles, or emphasized text.
`screen display`	`Wed May 6 17:01:03 2000`	This font indicates system output.
vertical bar |	**page** {**on** | **off**}	A vertical bar separates the choices when a parameter is required. The example indicates you can enter either command: **page on** or **page off** When entering the parameter, do not type the vertical bar or the brace brackets.

Notes, Cautions, Warnings

The following icons and associated text may appear in this document.

Note: A note contains tips, suggestions, and other helpful information, such as references to material not contained in the document, that can help you complete a task or understand the subject matter.

Caution: The exclamation point, within an equilateral triangle, is intended to alert the user to the presence of important installation, servicing, and operating instructions in the documents accompanying the equipment.

Warning: This symbol indicates that dangerous voltages levels are present within the equipment. These voltages are not insulated and may be of sufficient strength to cause serious bodily injury when touched. The symbol may also appear on schematics.

Contacting Support

Use the following information to contact Support:

U.S.	1-888-944-HELP
	1-888-944-4357
International	+.215-323-0044
WWW	http://www.gi.com/BUSAREA/CUSACC/websupport.html
Email	cmtssupport@motorola.com

1

Introduction

Overview

The BSR 64000™ system gives broadband carriers a competitive edge for defining, deploying, and managing broadband services. Based on Data Over Cable Service Interface Specification (DOCSIS) and Packet Cable standards, the BSR carrier-class solution allows Multiple System Operators (MSOs) to offer innovative differentiated data, voice, and multimedia services. The BSR provides the isolation, policing, and address management necessary for implementing measurable service level agreements (SLAs). It delivers traffic shaping for end-to-end SLAs across Hybrid Fiber Coax (HFC) infrastructure. Flexible management interfaces offer automated provisioning for accelerating service delivery.

The high-density BSR reduces headend congestion and streamlines operations and management, providing four times the performance for roughly a quarter of the cost and a quarter of the rack space.

Flexible interfaces for connectivity eliminate the need for discrete Cable Modem Termination System (CMTS) equipment, up converters, aggregation switches, and routers. The BSR offers unified management of routing and CMTS functions. It scales economically to meet increasing subscriber demands and the introduction of new services.

Centralized routing and distributed forwarding provide simple configuration, scalable performance, and low cost. Deployed in a distribution hub, the BSR provides an interchange point between the regional fiber network and the cable plant. In a regional headend, it interconnects the regional network with a backbone network and allows connectivity to local content servers and management systems.

Multiservice Support

The BSR enables next-generation services at the IP level: converged, data, voice, and multimedia services. Cable networks provide the foundation for innovative classes of entertainment and business services, including, but not limited to the following:

- IP telephony
- Interactive, multiplayer gaming
- On-demand music, audio and video
- Tiered data services

- Virtual Private Networks (VPNs)
- Application hosting

Network Management and Control

The BSR offers several management, control, and administration options. The BSR supports Simple Network Management Protocol (SNMP). All appropriate standard MIBs and private MIBs for monitoring and controlling the BSR value-added features are supported. The system supports the File Transfer Protocol (FTP). It can be seamlessly integrated into the existing network management infrastructure. The BSR also offers a command line interface (CLI) for ease-of-use and interoperability with legacy infrastructure. Easy-to-read diagnostic LEDs and remote management support provisioning, configuration, and problem identification.

2

Using the Command Line Interface

Overview

The BSR 64000™ command line interface (CLI) lets you enter commands at a connected terminal. You use the CLI to perform basic management tasks and to configure protocols and physical layer interfaces for the BSR. For further information on CLI commands, refer to the *BSR 64000 Command Reference Guide*. This chapter discusses the following topics:

- Using a Terminal Session to Access the BSR
- Disabling and Resetting Features
- Using Command Aliases
- Obtaining Help
- Using the Command History
- Editing Features

Using a Terminal Session to Access the BSR

You can access the CLI by connecting a terminal or PC with terminal emulation software to the BSR. The BSR supports one CLI session through its console port.

Follow these steps to start a terminal CLI session and set a password for the BSR:

1. Configure the terminal application on your PC to use COM port 1 or 2.
2. Confirm that a physical connection exists between the BSR and your terminal or PC.
3. Start your terminal or terminal application and enter its configuration mode.
4. Make sure the communications are set as shown in the table below:

Table 2-1 Console Settings

Parameter	Setting
Baud Rate	9,600
Data Bits	8
Flow Control	None

Table 2-1 Console Settings

Parameter	Setting
Parity	None
Stop Bits	1

5. Connect to the BSR. The console terminal session begins. The `MOT>` prompt displays.

6. To enter Privileged EXEC mode, use the enable command in User EXEC mode, as shown in the following example:

```
MOT>enable
```

   The `Password:` prompt displays.

7. To enter Privileged EXEC mode, press the Enter key at the password prompt. The `MOT#` prompt displays in Privileged EXEC mode.

Using a Telnet Session to Access the BSR

Once the Ethernet Interface on the BSR is assigned an IP address and the BSR password is set, the BSR can be accessed through a telnet session. Refer to Chapter 3 for more information on setting these parameters.

Note: If an IP address has not been configured for the Ethernet interface and a password has not been configured for the BSR, you can not access the BSR through telnet. The password also must be set or the telnet session is dropped.

To establish a Telnet session with the BSR, complete the following steps:

1. Start the telnet application. Enter the host name or the IP address of the BSR at the appropriate field or system prompt.

2. Press return. The following prompt displays:

```
MOT>
```

3. Enter the following case-sensitive command:

```
MOT> enable
```

This brings you to Privileged EXEC mode.

4. Press the Enter key at the password prompt. The CLI Telnet session begins.
5. To terminate the Telnet connection and exit the Telnet application when finished, enter **exit** at the prompt in Privileged EXEC mode.

Command Mode Access

The available commands depend on the command mode. Table 2-2 describes the basic modes. Figure 2-1 presents a flow chart of the modes.

Table 2-2 Command Mode Access, Prompt, and Exit Details

Mode Name	Access Means	Prompt Display	Exit Means
User EXEC	Console or Telnet	`MOT>`	To exit the CLI, enter the **logout** command. To enter Privileged EXEC mode, enter the **enable** or **login** command.
Privileged EXEC	Enter the User EXEC **enable** or **login** command.	`MOT#`	To return to User EXEC mode, enter the **disable** command. To enter Global Configuration mode, enter the **configure** command.
Global Configuration	Enter the Privileged EXEC **configure** command	`MOT(config)#`	To return to Privileged EXEC mode, enter the **exit** or **end** command or press Ctrl-Z. To enter Interface Configuration mode, enter any **interface** command. To enter Router Configuration mode, enter any **router** command, for example **router rip**.

Table 2-2 Command Mode Access, Prompt, and Exit Details

Mode Name	Access Means	Prompt Display	Exit Means
Interface Configuration	From Global Configuration mode, enter any **interface** command.	`MOT(config-if)#`	To return to Global Configuration mode, enter the **end** or **exit** command. To return to Privileged EXEC mode, press Ctrl-Z.

Router Configuration	From Global Configuration mode, enter any **router** command.	`MOT(config-bgp)#`
		`MOT(config-dvmrp)#`
		`MOT(config-isis)#`
		`MOT(config-ospf)#`
		`MOT(config-rip)#`
		(The prompt is protocol-dependent.)
			To return to Global Configuration mode, enter the **end** or **exit** command.
			To return to Privileged EXEC mode, press Ctrl-Z.

Route Map Configuration	From Global Configuration mode, enter any **route map** command.	`MOT(config-rmap)#`	To return to Global Configuration mode, enter the **end** or **exit** command. To return to Privileged EXEC mode, press Ctrl-Z.
Cable Spectrum Group	From Global Configuration mode, enter any **cable spectrum-group** command.	`MOT(config-spcgrp:`*<group-name>*`)#`	To return to Global Configuration mode, enter the **end** or **exit** command. To return to Privileged EXEC mode, press Ctrl-Z.

You begin a CLI session in User EXEC mode, which has a limited set of commands. To access other commands, switch to Privileged EXEC mode. In Privileged EXEC mode, you can enter any EXEC command or switch to Global Configuration mode. Configuration changes that you make in EXEC mode remain in effect until you reboot the system, unless you save your running configuration to your startup configuration.

In Global Configuration mode, you can make changes to the active configuration. If you save the configuration, your changes remain in effect after you reboot the BSR. In Interface Configuration mode, you enable operation features on a per interface basis. In Router Configuration mode, you can enable routing protcol features.

User EXEC Mode

When you telnet into the BSR, you are in the CLI User EXEC mode. The prompt `MOT>` indicates User EXEC mode. User EXEC commands are a subset of commands available in Privileged EXEC mode. User EXEC commands allow you to perform basic tests and list system information.

Privileged EXEC Mode

If a Privileged EXEC mode password exists, the CLI prompts you to enter it to gain access. The case-sensitive password does not appear on the screen. Privileged EXEC mode includes the User EXEC commands as well as the **configure** command, which you can use to access the remaining command modes and high-level testing commands, such as **debug** (available in Privileged EXEC mode).

To enter Privileged EXEC mode, login in and enter the **enable** command at the `MOT>` prompt. The prompt changes to `MOT#`. For security purposes, if no **enable** password exists, you can enter Privileged EXEC mode from console.

Global Configuration Mode

Global Configuration commands apply to features that affect the entire system. These commands apply to system features and enable routing functions. To enter Global Configuration mode, enter the **configure** command from Privileged EXEC mode. The prompt changes to `MOT(config)#`. To return to Privileged EXEC mode, enter the **end** or **exit** command or press Ctrl-Z.

Interface Configuration Mode

You enable features on a per-interface basis. Interface Configuration commands modify the operation of an interface such as an Ethernet port. Interface Configuration commands always follow a Global Configuration command, which defines the interface type. From Global Configuration mode, enter Interface Configuration mode by entering any **interface** command, such as the following:

```
MOT(config)#interface cable 3/0
```

The prompt changes to `MOT(config-if)#`, To exit Interface Configuration mode and return to Global Configuration mode, enter the **exit** command. To exit configuration mode and return to Privileged EXEC mode, use the **end** command or the **exit** command or type Ctrl-Z.

Router Configuration Mode

In Router Configuration mode you can enable routing protocol features. From Global Configuration mode, enter Router Configuration mode by entering any **router** command. The prompt changes in relation to the specific protocol. (See Figure 2-1.) For example, if you enter the **router bgp** command, the prompt changes to `MOT(config-bgp)#`. To exit Router Configuration mode and return to Global Configuration mode, enter the **exit** command. To exit and return to Privileged EXEC mode, use the **end** command or press Ctrl-Z.

Route Map Configuration Mode

In Route-map Configuration mode, you can establish route maps with the conditions for redistributing routes from one routing protocol to an another. From Global Configuration mode, enter Route-map Configuration mode by entering any route map command. The prompt changes to MOT(config-rmap)#. To exit Route-map Configuration mode and return to Global Configuration mode, enter the **exit** command. To exit and return to Privileged EXEC mode, use the **end** command or press Ctrl-Z.

Cable Spectrum Group Mode

The spectrum management system monitors the upstream signal integrity, and collects upstream spectrum information. In Cable Spectrum Group Mode, you can configure a spectrum group, apply a spectrum group to an upstream port, and evaluate spectrum performance. From Global Configuration mode, enter Cable Spectrum Group mode by entering any cable spectrum-group command. The prompt changes to `MOT(config-spcgrp:<group-name>)#`. To exit Cable Spectrum Group mode and return to Global Configuration mode, enter the **exit** command. To exit and return to Privileged EXEC mode, use the **end** command or press Ctrl-Z.

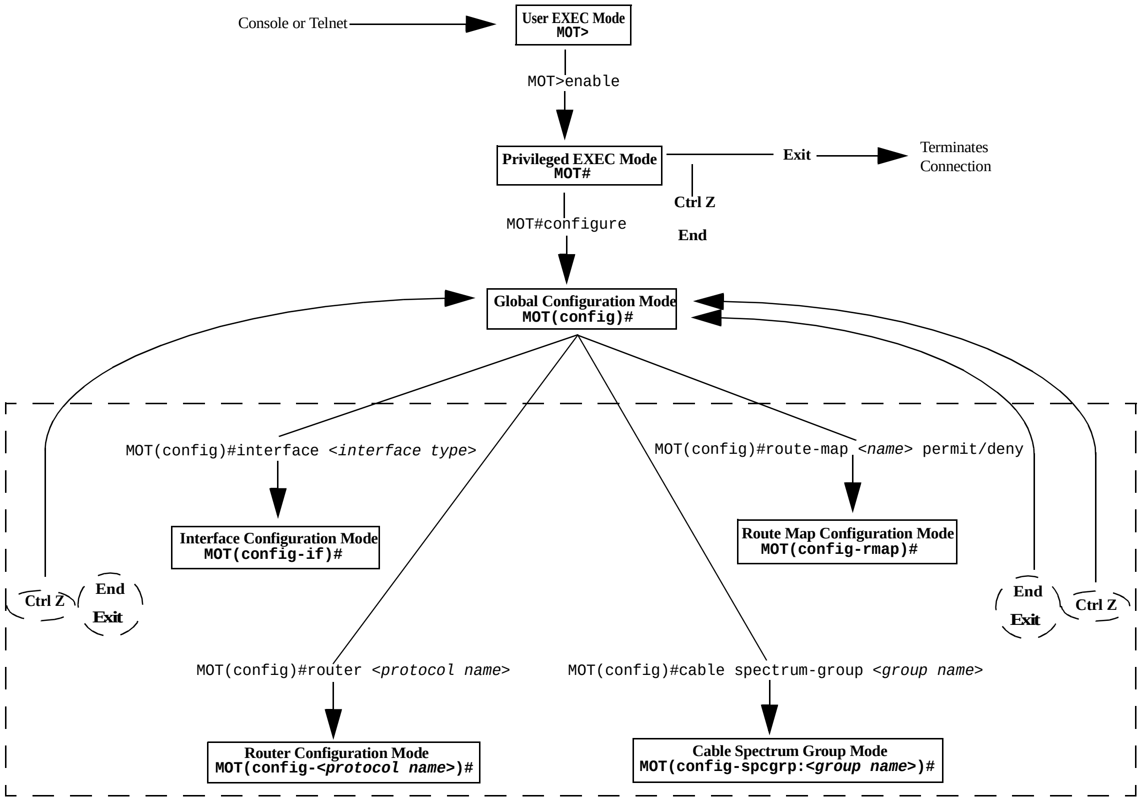

Console or Telnet
User EXEC Mode
MOT>
MOT>enable
Privileged EXEC Mode
MOT#
Exit
Terminates Connection
Ctrl Z
End
MOT#configure
Global Configuration Mode
MOT(config)#
MOT(config)#interface <interface type>
Interface Configuration Mode
MOT(config-if)#
MOT(config)#route-map <name> permit/deny
Route Map Configuration Mode
MOT(config-rmap)#
Ctrl Z
End
Exit
End
Exit
Ctrl Z
MOT(config)#router <protocol name>
Router Configuration Mode
MOT(config-<protocol name>)#
MOT(config)#cable spectrum-group <group name>
Cable Spectrum Group Mode
MOT(config-spcgrp:<group name>)#

Figure 2-1 CLI Command Flow Chart

Disabling and Resetting Features

Use the **no** form of a command to disable a feature. Use the command without the keyword **no** to re-enable a disabled feature or to enable a feature that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command, and use **ip routing** to re-enable it.

Configuration commands also have a default form, which returns the command setting to its default. Most commands are disabled by default, so the default form is the same as the **no** form. However, some commands are enabled by default and have variables set to certain default values. In these cases, the default command enables the command and sets variables to their default values.

Using Command Aliases

You can create your own alias for a command. Use the information in this section to handle aliases.

1. To display a list of all aliases, use the **show aliases** command, as shown below:

   ```
   MOT#show aliases
   ```

2. To create an alias for a command in User EXEC mode, use the **alias exec** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#alias exec {<WORD> <WORD>}
   ```

   where:

   WORD is the alias name for the command

   WORD is the name of the command being replaced by the alias

3. To create an alias for a command in Privileged EXEC mode, use the **alias priv** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#alias priv {<WORD> <WORD>}
   ```

   where:

   WORD is the alias name for the command

   WORD is name of the command being replaced by the alias

4. To create an alias for a command in Global Configuration mode, use the **alias conf** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#alias conf {<WORD> <WORD>}
   ```

   where:

   WORD is the alias name for the command

WORD is the name of the command being replaced by the alias

5. Use the **no alias** command to delete an alias. For example:

   `MOT(config)#`**no alias** [**exec** | **priv** | **conf**] <*WORD*>

   where:

   WORD is the alias name for the command

   The alias is removed for the associated command.

Examples

The following example creates an alias for the **enable** command, accessible from the Privileged EXEC mode.

`MOT>`**alias exec en enable**

This example creates an alias for the **router rip** command for use within the Interface Configuration mode.

MOT(config-if)#**alias conf rr router rip**

When invoked, this alias moves you to Router RIP mode.

Obtaining Help

Enter a question mark (?) at the prompt to display help for available commands. You can enter the question mark with the complete command or its unique abbreviation. For example, to obtain help for the **show users** command, you can enter **show users ?**.

If the CLI detects an error a the command line, it positions a caret symbol (^) at the error.

Context-sensitive Help

Display a list of command-associated keywords and arguments by using the context-sensitive help features. To get help for a specific command mode, command, keyword, or argument, use the entries in Table 2-3.

Table 2-3 Context-sensitive Help Details

Entry	**Result**
help	Displays brief help system description.
<abbreviated command>?	Displays commands that begin with the abbreviated entry. Do not enter a space before the question mark.
?	Lists all commands available for the current mode.
<command> **?**	Lists associated keywords for the command. Be sure to enter a space before the question mark.
<command> **keyword ?**	Lists associated arguments for the keyword.

Using the Command History

The CLI provides a history or record of command entries. This feature redisplays long or complex commands or entries, including access lists. Use the command history feature to complete the following tasks:

- Setting the Command History Buffer Size
- Recalling Commands
- Disabling the Command History Feature

Setting the Command History Buffer Size

By default, the history buffer stores ten command lines. To change the number of stored command lines for the current terminal session, use the **history size** command in User EXEC or Privileged EXEC mode, as shown below:

```
MOT>history size [<1-256>]
```

where:

1-256 is the number of lines of the history buffer.

Use the **no history size** command to reset the number of lines saved in the history buffer to the default, ten lines, as shown below:

```
MOT>no history size
```

Recalling Commands

To recall commands from the history buffer, perform one of the following actions:

- Press Ctrl-P or the up arrow key **-** Use this action to recall commands, displaying the most recent command first. Repeat the key sequence to display successively older commands.
- Press Ctrl-N or the down arrow key **-** After recalling commands, use this action to display more recent commands. Repeat the key sequence to display successively more recent commands.
- **show history** command **-** Enter this command in User EXEC or Privileged EXEC mode to display the last several commands.

Note: The arrow keys function is available only on ANSI-compatible terminals, such as VT100s.

Disabling the Command History Feature

The command history feature is automatically enabled. To disable it during the current session, use the **no history** command in User EXEC or Privileged EXEC mode, as shown below:

```
MOT>no history
```

Editing Features

You can enter CLI commands in uppercase, lowercase, or a combination of cases. Passwords and some identifiers, such as file names or route maps, are case sensitive. You can abbreviate commands and keywords to a number of characters that represent a unique abbreviation. Enter the command line at the system prompt, and then press the Enter key to execute the command. This section describes how to do the following:

- Navigating the Command Line
- Completing a Partial Command Name
- Handling Command Lines
- Deleting Entries
- Scrolling Down a Line or a Screen
- Transposing Characters
- Controlling Case
- Using Output Modifiers to Limit Show Command Output

Navigating the Command Line

Table 2-4 describes the key sequences you can use to move the cursor on the command line to make corrections or changes.

Table 2-4 Cursor Movement Keys

Pressing ...	Function
Ctrl B	Moves the cursor back one character.
Left arrow	Moves the cursor back one character.
Ctrl-F	Moves the cursor forward one character.
Right arrow	Moves the cursor forward one character.
Ctrl-A	Repositions the cursor to the beginning of the command line.
Ctrl-E	Repositions the cursor to the end of the command line.

Table 2-4 Cursor Movement Keys

Pressing ...	Function
Esc B	Moves the cursor back one word.
Esc F	Moves the cursor forward one word.

Completing a Partial Command Name

Press the Tab key to complete a partial entry. To invoke this feature, enter the first few letters and press the Tab key. This completes the command name.

```
MOT>his [Tab]
```

results in

```
MOT>history
```

Handling Command Lines

The CLI provides a wraparound feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. If you enter a command and the system displays a message on your screen, you can easily recall your current command line entry. To scroll back to the beginning of the command line or to recall the current command line entry, use the keys in Table 2-5.

Table 2-5 Command Line Control Key Sequences

Pressing ...	Function
Ctrl-A	Scrolls to the beginning of the line. You can then verify that you entered the command correctly.
Ctrl-B repeatedly	Scrolls to the beginning of the command entry.
Ctrl-C	Ends the Telnet session.
Ctrl-L	Redisplays the current command line.

Table 2-5 Command Line Control Key Sequences

Pressing ...	Function
Ctrl-R	Redisplays the current command line.
Left arrow key repeatedly	Scrolls back to the beginning of the command entry.

Deleting Entries

Table 2-6 describes actions to delete command entries.

Table 2-6 Deletion Keys

Press ...	Function
Backspace key	Erases the character to the left of the cursor.
Ctrl-D	Deletes the character at the cursor.
Ctrl-K	Deletes each character, from the cursor to the end of the command line.
Ctrl-U	Deletes each character, from the cursor to the beginning of the command line.
Ctrl-W	Deletes the word to the left of the cursor.
Delete key	Erases the character to the left of the cursor.
Esc D	Deletes each character, from the cursor to the end of the word

Scrolling Down a Line or a Screen

Using the help facility to list commands available in a particular mode may result in a list longer than the terminal screen can display. If the `more` prompt appears at the bottom of a screen, this indicates that more information is available. Use the keys in Table 2-7 to obtain the additional information.

Table 2-7 Scroll Keys

Press ...	Function
Enter key	Scrolls down one line.
Space bar	Scrolls down one screen.

Transposing Characters

You can transpose characters by pressing Ctrl-T. This transposes the character to the left of the cursor with the character at the cursor.

Controlling Case

To capitalize or lowercase letters, use the keys in Table 2-8.

Table 2-8 Case Control Keys

Press ...	Function
Esc C	Capitalizes the character at the cursor.
Esc L	Changes the word at the cursor to lowercase.
Esc U	Capitalizes letters from the cursor to the end of the word.

Using Output Modifiers to Limit Show Command Output

Filters or "output modifiers" display specific show command information using the "pipe" character (|) and entering the **begin**, **exclude**, and **include** parameters.

For example:

show ip ospf network [|] [**begin** | **exclude** | **include**] <*text*>

where:

| turns on filters

begin indicates to start with the line that matches

exclude excludes lines that match.

include include lines that match

text is the text string to match.

3

Configuring the System

Overview

This chapter describes the initial configuration procedures necessary to configure the BSR 64000™ system using its command line interface (CLI). For further information on the CLI commands described in this chapter, refer to the *BSR 64000 Command Reference Guide*. This chapter discusses the following topics:

- Initial Configuration Tasks
- Configuring System Log Parameters
- Sending Messages to BSR Users
- Configuring Server Related Parameters
- Downloading Software
- Specifying the System Image File Boot Location
- Saving and Viewing Your Configuration
- Reseting BSR Modules
- Monitoring the System

Initial Configuration Tasks

This section describes the initial basic configuration tasks for configuring the BSR:

- Gathering Network Information
- Required Servers
- Accessing the CLI to Set System Passwords
- Specifying a Host Name
- Configuring User Login Accounts
- Configuring Interfaces
- Specifying System Time Information

Gathering Network Information

Before you begin the initial configuration of the BSR, you should determine the following information:

- Interface IP address(es) and subnet mask(s)
- Time of Day Server IP address
- DHCP Server IP address
- Cable Modem (CM) authentication string or hexadecimal key information contained in the CM configuration file. You must have this information when you configure authentication parameters on the BSR.

Required Servers

The following servers are required for the basic operation of the BSR on your network, and must be configured to allow cable modems to range and register properly on the HFC network:

- DHCP
- TFTP

Note: The CM configuration file must be stored on the TFTP server.

The following DHCP options are necessary:

- IP address
- Router address
- TFTP server address
- Bootfile for the CM configuration file

The following servers can be configured to operate the BSR on your network for management, provisioning, troubleshooting and billing purposes:

- LDAP
- Event (Syslog) Server
- Provisioning Server
- DNS

For more information on installing the servers, refer to the vendor server software documentation.

Accessing the CLI to Set System Passwords

Follow these steps to access the CLI from a console session in order to configure password privileges for enabled modes and telnet:

Note: Make sure that the serial cable is connected properly and the terminal application is configured correctly. Refer to the BSR 64000 Installation Guide for more information.

1. Start your terminal or terminal application to connect to the BSR. Refer to Chapter 2 for more information on configuring your terminal or terminal application.
2. Power on the BSR 64000.

Warning: Do not interrupt the boot process.

3. The terminal session begins and the password prompt displays. The password is a null value by default. Press the **Enter** key. The `MOT>` prompt displays.
4. To enter Privileged EXEC mode, use the **enable** command in User EXEC mode, as shown below:

   ```
   MOT>enable
   ```

   The `Password` prompt displays.
5. To enter Privileged EXEC mode, press the Enter key at the password prompt. The password is a null value by default.
6. Use the **configure** command to enter Global Configuration mode in order to set system passwords, as shown below:

```
MOT#configure
```

The `MOT(config)#` prompt displays.

Setting System Passwords

System passwords should be set immediately. System passwords can contain up to 31 uppercase or lowercase alphanumeric characters and a numeric character cannot be the first character. Spaces are valid password characters. The user must enter the correct password to gain access to the BSR and privileged-level commands.

Note: Access to a telnet session is denied if the password for both the console and telnet is not set.

Follow these steps to configure the BSR system passwords:

1. To set the password for a console (terminal) session that allows access to the BSR in User EXEC mode, use the **password console** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#password console {0 | 7} <WORD>
   ```

   where:

   0 indicates that the following password is unencrypted (clear text).

   7 indicates that the following password is encrypted.

   WORD is the user-defined password that is no more than 31 characters.

2. To set the password for a telnet session that allows access to the BSR in User EXEC mode, use the **password telnet** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#password telnet {0 | 7} <WORD>
   ```

   where:

   0 indicates that the following password is unencrypted (clear text).

   7 indicates that the following password is encrypted.

WORD is the user-defined unencrypted password for the BSR that is no more than 31 characters.

3. To set the Privileged EXEC password, use the **enable password** command, as shown below:

```
MOT(config)#enable password {0 | 7} <WORD>
```

where:

0 indicates that the following password is unencrypted (clear text).

7 indicates that the following password is encrypted.

WORD is the user-defined unencrypted password for the BSR that is no more than 31 characters.

4. Automatic encryption is disabled by default. If you want to encrypt all currently unencrypted passwords and all future passwords entered on the BSR, use the **service password-encryption** command in Global Configuration mode, as shown below:

```
MOT(config)#service password-encryption
```

If you want to turn off the service password encryption feature so that passwords entered in the future are no longer encrypted, use the **no service password-encryption** command in Global Configuration mode, as shown below:

```
MOT(config)#no service password-encryption
```

Note: The **no service password-encryption** command does not unencrypt passwords that are already encrypted. If you want to unencrypt encrypted passwords, you must change them manually.

5. The **show running-config** command is used to determine if the password name and encryption has been set. Use the **show running-config** command in Privileged EXEC mode to verify your configuration, as shown below:

```
MOT(config)#show running-config
```

Note: The **show running-config** command output identifies the system password with the number 0 if it is unencrypted. If the system password is encrypted, it is identified with the number 7.

Specifying a Host Name

To optionally assign or change your BSR system network name, use the **hostname** command in Global Configuration mode, as shown below:

```
MOT(config)#hostname <WORD>
```

where:

WORD is the new system network name.

After you execute this command, the Command Line Interface (CLI) prompt changes to the new host name, as shown below:

```
newhostname(config)#
```

Configuring User Login Accounts

Define a unique system login account for each user requiring access to the command line interface. You can a define system login account with different levels of security access to the system. The **username** command allows you to define a complete system login including the user name, password, access-level, and user group.The following commands are used for defining a user account:

username nopassword

username password

username privilege

username user-group

Table 3-1 gives a brief description of each parameter required to configure a user login account. The sections that follow describe the procedural details for defining each parameter.

Table 3-1 User Login Account Parameters

Parameter	Description
username	Defines the name of the user account. A user name comprises a unique set of up to 16 case-sensitive characters.
nopassword	Defines no password for the user account.
password	Defines the password for the user account. A password comprises a unique set of up to 31 case-sensitive characters. Password can be specified to appear encrypted or unencrypted in the running-config file.
privilege	Defines user account privileges. Read-only privileges allow a user access to the Privileged EXEC command line mode only. Read-write privileges allow a user access to all command line modes

user-group	Defines a user account group access level to CLI command sets.
	isp = internet service provider
	mso = multiservice operator
	sysadmin = System Administrator

Defining a User Name without a Password

If you want to define a user account with no password, use the following command in Global Configuration mode:

```
MOT(config)#username <WORD> nopassword
```

where:

WORD is the user account login name.

For example:

```
MOT(config)#username newuser nopassword
```

Defining a User Name with an Unencrypted Password

Follow these steps to define a user account with an unencrypted password:

1. Use the **username password** command in Global Configuration mode to define an unencrypted password for a user account, as shown below:

   ```
   MOT(config)#username <WORD> password <WORD>
   ```

   where:

   WORD is the user account login name.

   WORD defines the user login account password.

   For example:

   ```
   MOT(config)#username newuser password mypassword
   ```

Defining a User Name with an Encrypted Password

Follow these steps to define a user account that is encrypted:

1. Use the username password command in Global Configuration mode to define a password for a user account that is encrypted, as shown below:

   ```
   MOT(config)#username <WORD> password [0 | 7] {<WORD>}
   ```

   where:

   WORD defines the user login name.

   0 specifies that an unencrypted password follows.

   7 specifies that an encrypted (hidden) password follows.

   WORD defines the user login account password.

2. Automatic encryption is disabled by default. If you want to encrypt the user account, use the **service password-encryption** command in Global Configuration mode as shown below:

   ```
   MOT(config)#service password-encryption
   ```

   For example:

   ```
   MOT(config)#username newuser password mypassword
   ```

   ```
   MOT(config)#service password-encryption
   ```

Defining a Privilege Level

To define a privilege level for a user account, use the following command in Global Configuration mode:

```
MOT(config)#username <name> privilege [ro | rw]
```

where:

name is the user account login name

ro defines a privilege level of read-only that restricts this user to Privileged EXEC command mode access only

rw defines a privilege level of read-write that allows this user access to any command mode

For example:

```
MOT(config)#username newuser privilege rw
```

Defining a Group Access Level

To define a group access level for a user account, use the following command in Global Configuration mode:

```
MOT(config)#username <name> user-group {isp <num: 1,1> | mso | sysadmin}
```

where:

name is the user account login name

user-group is one of the groups shown below. The ISP and MSO groups have access to a specific set of CLI commands.

User Group	Command Line Access
sysadmin	All CLI commands
ISP	Most CLI commands including routing commands but excluding cable commands.
MSO	Most CLI commands including cable commands but excluding routing commands.

For example:

MOT(config)#**username newuser user-group mso**

Verifying Your User Account Login Configuration

Use the **show running-config** command in Privileged EXEC mode to verify your user account configuration, as shown below:

MOT(config)#**show running-config**

In the following example, user account passwords have not been encrypted:

Note: The **show running-config** command output identifies the user account password with the number 0 if it is unencrypted. If the user account password is encrypted, it is identified with the number 7.

```
no service password-encryption
!
username root user-group sysadmin
username root password 0 root
username manuf user-group sysadmin
username manuf password 0 river
username diag user-group sysadmin
username diag password 0 delta
username ispuser user-group isp 1
username ispuser privilege rw
username ispuser password 0 ispuser
username msouser user-group mso
username msouser privilege rw
username msouser password 0 msouser
```

Configuring Interfaces

You must configure the interfaces on the BSR in order for the BSR to transmit and receive data and communicate with other network devices. Refer to Chapter 5 for more information on configuring interfaces.

Specifying System Time Information

Follow these steps to set system time information:

To set the time zone, use the **clock timezone** command in Global Configuration mode, as shown below:

```
MOT(config)#clock timezone {<WORD> <Hours_offset>
```

where:

WORD is the name of the time zone.

Hours_offset is the number of hours offset from Universal Time Coordinated (UTC); valid entries are -23 to +23 hours.

To set the system clock, type **Ctrl Z** or **exit** to return to Privileged EXEC mode and enter either of the **clock set** commands, as shown in the examples below. These examples show how to manually set the clock to 4:30 a.m. on May 1, 2001:

```
MOT#clock set 04:30:00 1 May 2001
```

```
MOT#clock set 04:30:00 May 1 2001
```

Configuring System Log Parameters

The BSR can be set to generate log messages when the configuration changes and when certain network or device events occur. This section describes how to configure the system log parameters.

The tasks described in this section involve some parameters you may want to change, such as the logging type and the severity level of the information that is logged. The tasks for configuring the system log include the following:

- Configuring Logging for a Remote Syslog Server
- Configuring Console Logging
- Setting the Logging Buffer
- Restricting Logging Rates and SNMP Traps
- Setting the Recording Mechanism for CMTS Reports

- Logging Your CLI Session to the Syslog Server

Configuring Logging for a Remote Syslog Server

You can configure up to three remote syslog servers. Follow these steps to configure logging parameters for a remote syslog server:

1. To configure system log parameters for a remote syslog server, use the **logging** command in Global Configuration mode, as shown below:

```
MOT(config)#logging {<A.B.C.D>}
```

where:

A.B.C.D is the IP address of the syslog server

For example:

```
MOT(config)#logging 10.10.10.53
```

Note: Use the **no logging** command if you want to disable logging on the remote syslog server.

2. To identify the logging facility, use the **logging facility** command in Global Configuration mode, as shown below:

```
MOT(config)#logging facility {local0 | local1 | local2 | local3 | local4 | local5
| local6 | local7}
```

If you do not identify the logging facility using this command, the system defaults to **local7**.

Note: Remember to match this setting in the config file of your syslog server.

3. To set the severity level of messages to be logged to the remote syslog servers, use the **logging trap** command in Global Configuration mode, as shown below:

```
MOT(config)#logging trap {emergencies | alerts | critical | errors |
warnings | notifications | informational | debugging}
```

where:

emergencies logs emergency conditions where the system is unusable (severity level 0).

alerts logs conditions where immediate action is needed (severity level 1).

critical logs critical conditions (severity level 2).

errors logs error conditions (severity level 3).

warnings logs warning conditions (severity level 4).

notifications logs normal but significant conditions (severity level 5).

informational logs informational descriptive system information (severity level 6).

debugging logs debugging messages (severity level 7).

The following example configures the remote syslog server to log all messages from **warnings** (severity level 4) on up to **emergencies** (severity level 0):

```
MOT(config)#logging trap warnings
```

4. To enable logging on the remote syslog server, use the **logging on** command.

```
MOT(config)#logging on
```

5. To verify that the syslog server parameters are set correctly, use the **show running-config** command in Privileged EXEC mode.

Configuring Console Logging

1. To set the severity level of messages to be logged to the local console, use the **logging console** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#logging console {emergencies | alerts | critical | errors |
   warnings | notifications | informational | debugging}
   ```

   where:

   emergencies logs emergency conditions where the system is unusable (severity level 0).

   alerts logs conditions where immediate action is needed (severity level 1).

   critical logs critical conditions (severity level 2).

   errors logs error conditions (severity level 3).

   warnings logs warning conditions (severity level 4).

   notifications logs normal but significant conditions (severity level 5).

   informational logs informational descriptive system information (severity level 6).

   debugging logs debugging messages (severity level 7).

   For example:

   ```
   MOT(config)#logging console notifications
   ```

2. To enable logging on your local console, use the **logging on** command.

   ```
   MOT(config)#logging on
   ```

   To verify that the syslog server parameters are set correctly, use the **show running-config** command in Privileged EXEC mode.

Setting the Logging Buffer

Buffering is used to allow space on the internal logging buffer on the BSR or syslog server for the latest messages. If buffering is enabled, messages are overwritten to allow space for the latest messages when the internal buffer reaches maximum capacity, which is 16 Megabytes.

1. To set the logging buffer size, use the **logging buffered** command in Global Configuration mode, as shown below:

`MOT(config)#`**logging buffered** <*4096-5242880*>

where:

4096-5242880 is the logging buffer size expressed in bytes.

2. To specify what logged information is buffered, use the **logging buffered** command, in Global Configuration mode, as shown below:

`MOT(config)#`**logging buffered** {**emergencies | alerts | critical | errors** | **warnings | notifications | informational | debugging**}

where:

emergencies logs emergency conditions where the system is unusable (severity level 0).

alerts logs conditions where immediate action is needed (severity level 1).

critical logs critical conditions (severity level 2).

errors logs error conditions (severity level 3).

warnings logs warning conditions (severity level 4).

notifications logs normal but significant conditions (severity level 5).

informational logs informational descriptive system information (severity level 6).

debugging logs debugging messages (severity level 7).

3. Use the **show log** command in Privileged EXEC or Global Configuration mode to see messages logged in the internal buffer. The oldest message is displayed first.

Clearing the Buffer

1. Clear the log using the **clear log** command in all modes except Privileged EXEC mode, as shown below:

   MOT#**clear log**

2. Use the **show log** command to verify that the log has been cleared, as shown below:

   MOT(config)#**show log**

Restricting Logging Rates and SNMP Traps

You can restrict logging rates and SNMP traps to save space on your device. When the rate of logging messages exceeds the configured limit, logging stops.

Follow these steps to restrict the rate of messages being logged:

1. To specify the number of logged messages allowed per number of seconds, use the **logging rate-limit** command in Global configuration mode as shown below:

   MOT(config)#**logging rate-limit** [*<0-2147483647> <1-2147483647>*]

   where:

   0-2147483647 is the number of messages.

   1-2147483647 is the number of seconds at which the specified number of syslog and trap messages are logged.

   The following example indicates that the rate-limit on logged messages is 10 messages per second, for example:

   MOT(config)#**logging rate-limit 10 1**

2. To automatically re-enable logging when the logging rate falls below the rate-limit restriction, use the **logging rate-limit auto-restart** command in Global Configuration mode, as shown below:

```
MOT(config)#logging rate-limit auto-restart
```

Note: To disable rate limitations, use the **no logging rate-limit** command in Global Configuration mode.

Setting the Recording Mechanism for CMTS Reports

Refer to Table 3-2 and Table 3-3 for the available logging options used with the **logging reporting** command in Global Configuration mode. These logging options allow you to select the report type and storage location for CMTS reports, as shown below:

```
MOT(config)#logging reporting {emergencies | alerts | critical | errors |
warnings | notifications | informational | debugging} {all-clear | all-set | local |
local-localvol | local-syslog | local-syslog-localvolatile | local-trap |
local-trap-localvolitile | localvol | syslog-localvol | trap-localvol |
trap-syslog-localvol}
```

For example, if you want to log critical report messages to local non-volatile memory (NVRAM), use the **logging reporting critical local** command, as shown below:

```
MOT(config)#logging reporting critical local
```

Table 3-2 describes each of the available logging report options:

Table 3-2 Logging Report Options

Report	Description
emergencies	Logs emergency conditions where the system is unusable (severity level 0).
alerts	Logs conditions where immediate action is needed (severity level 1).
critical	Logs critical conditions (severity level 2).
errors	Logs error conditions (severity level 3).
warnings	Logs warning conditions (severity level 4).
notifications	Logs normal but significant conditions (severity level 5).

Table 3-2 Logging Report Options

Report	Description
informational	Logs informational descriptive system information (severity level 6).
debugging	Logs debugging messages (severity level 7).

Table 3-3 describes the location where report messages are logged:

Table 3-3 Report Location Options

Report Location	Description
all-clear	Unsets all logging locations for the report.
all-set	Sets all logging locations for the report.
local	Log messages for the report go to local-nonvolatile memory (NVRAM).
local-localvol	Log messages for the report go to local NVRAM and local-volatile or "dynamic" memory (DRAM).
local-syslog	Log messages for the report go to local NVRAM and the syslog server.
local-syslog-localvol	Log messages for the report go to local NVRAM and local DRAM and the syslog server.
local-trap	Log messages for the report go to local NVRAM. SNMP traps are also sent to an SNMP manager.
local-trap-localvol	Log messages for the report go to local NVRAM and DRAM memory. SNMP traps are also sent to an SNMP manager.
local-trap-syslog	Log messages for the report go to local DRAM and a syslog server. SNMP traps are also sent to an SNMP manager.
localvol	Log messages for the report go to local DRAM.
syslog-localvol	Log messages for the report go to the syslog server and local DRAM.
trap-localvol	Log messages for the report go to local DRAM. SNMP traps are also sent to an SNMP manager.
trap-syslog-localvol	Log messages for the report go to the syslog server and local DRAM. SNMP traps are also sent to an SNMP manager.

Returning to the Default CMTS Log Reporting Configuration

If you want to return to the default CMTS log reporting configuration, use the **logging reporting default** command in Privileged EXEC mode, as shown below:

```
MOT#logging reporting default
```

Logging Your CLI Session to the Syslog Server

Use the **logging session** command in Privileged EXEC mode to log your CLI session to your syslog server, as shown below:

```
MOT#logging session
```

Sending Messages to BSR Users

Use the following commands to send messages to BSR users:

- Issue the **broadcast** command to broadcast a message to all connected users at any given moment, as shown in Privileged EXEC mode, as shown below:

  ```
  MOT#broadcast {<WORD>}
  ```

  where:

  WORD is the message intended for broadcast.

- Issue the **banner motd** command in Global Configuration mode to specify the message-of-the-day (MOTD) that is displayed for all connected users once they successfully login to the BSR, as shown below. The MOTD is not configured by default.

  ```
  MOT(config)#banner motd [<WORD> | <1, 10>]
  ```

  where:

  1, 10 is the MOTD line number from 1 to 10. Up to 10 MOTD lines can be configured.

  WORD is the MOTD text.

Example

The following example configures a MOTD. The pound sign (#) is the delimiting character.

MOT(config)#**banner motd The router will be rebooted at 12 a.m.**

Use the **no motd** command to delete the MOTD banner.

Configuring Server Related Parameters

This chapter describes how to configure server-related parameters on the BSR in order to establish proper communication between the BSR and the different types of servers that are connected to the BSR.

Use the following sections to configure server-related parameters on the BSR:

- Configuring DHCP Relay
- Configuring DNS
- Configuring LDAP
- Configuring SNTP
- Configuring UDP Broadcast Relay
- Configuring FTP Access
- Enabling the RADIUS Client on the BSR

Configuring DHCP Relay

This section describes how to configure the BSR to forward UDP broadcasts, including IP address requests, from Dynamic Host Configuration Protocol (DHCP) clients. You can configure the BSR to act as a DHCP relay agent. In this case, a locally attached host can issue a DHCP or BOOTP request as a broadcast message. If the router sees this broadcast message, it relays the message to a specified DHCP or BOOTP server.

The DHCP client-server protocol enables devices on an IP network (the DHCP clients) to request configuration information from a DHCP server. DHCP Relay configures the BSR to forward UDP broadcasts, including IP address requests, from DHCP clients.Configure the BSR to be a DHCP relay agent if you have locally attached hosts and a distant DHCP or BOOTP server.

To configure the BSR for DHCP relay, do the following:

1. Enter Interface Configuration mode for the Ethernet interface.

   `MOT(config)#`**interface ethernet** <*x*>/<*y*>

   where:

   x is the module slot number with an available Ethernet interface.

   y is the Ethernet interface number.

2. Issue the the **ip helper-address** command in Interface Configuration mode to forward default UDP broadcasts including IP configuration requests to the DHCP server, as shown below:

   `MOT(config-if)#`**ip helper-address** <*A.B.C.D*>

   where:

   A.B.C.D is the destination address.

Example

   `MOT(config-if)#`**ip helper-address 200.200.200.1**

   You can use the cable helper-address command in Interface Configuration mode to configure DHCP relay on the cable interface. Refer to "Subneting DHCP Clients on the Cable Interface" on page 6-1 for more information.

3. Enable the Ethernet interface and the configuration change with the **no shutdown** command.

   `MOT(config-if)#`**no shutdown**

4. Exit Interface Configuration mode.

   `MOT(config-if)#`**exit**

5. Exit Global Configuration mode.

   `MOT(config)#`**exit**

6. Verify that the information was entered correctly by displaying the running configuration in Privileged EXEC mode.

   ```
   MOT#show running-config
   ```

Configuring DNS

Domain Name System (DNS) maps host names to IP addresses. For example, it allows you to reference the host *motorola.com* instead of having to remember that its IP address is 198.93.23.13.

Configuring DNS involves the following tasks:

- Specifying DNS Name Servers
- Configuring the Domain Name
- Enabling Domain Lookup and Domain List

Specifying DNS Name Servers

Use the **ip name-server** command in Global Configuration mode to specify a Domain Name Server (DNS) that helps the BSR match DNS host names with their IP addresses, as shown below:

```
MOT(config)#ip name-server {<A.B.C.D>}
```

where:

A.B.C.D is the IP address of the Domain Name Server (DNS).

For example:

```
MOT(config)#ip name server 192.168.1.253
```

Configuring the Domain Name

For each BSR, you should configure the name of the domain in which the BSR is located. This is the default domain name that is appended to host names that are not fully qualified. To configure the domain name, use the **ip domain-name** command in Global Configuration mode.

```
MOT(config)#ip domain-name <name>
```

where:

name is the default domain name.

For example:

```
MOT(config)#ip domain-name motorola.com
```

Enabling Domain Lookup and Domain List

DNS servers provide forward lookups, which determine the IP address of a provided device name. This is the most common kind of lookup performed. DNS servers also provide a domain list function which completes unqualified host names.

1. To enable IP domain name system hostname translation, use the **ip domain-lookup** command in Global Configuration mode. This feature is enabled by default.

   ```
   MOT(config)#ip domain-lookup
   ```

2. To create a domain list of up to six (6) host names to complete unqualified host names, use the **ip domain-list** command in Global Configuration mode. If the primary domain-name fails to resolve, the software uses these names.

   ```
   MOT(config)#ip domain-list <WORD>
   ```

   where:

   WORD indicates the domain name to use to resolve unqualified host names when the primary domain fails to resolve.

3. Verify that the information was entered correctly by displaying the running configuration in Privileged EXEC mode.

   ```
   MOT#show running-config
   ```

Configuring LDAP

Lightweight Directory Access Protocol (LDAP) servers provide a way to name, manage, and access collections of attribute-value pairs. LDAP servers consist of entries that hold information about some thing or concept, such as a person or organization. Every entry in an LDAP server belongs to one or more object classes.

- Specifying the primary or secondary LDAP server addresses

- Starting the LDAP client
- Specifying the start of the search-tree

1. Use the **ldap server** command in Global Configuration mode to configure a primary or secondary LDAP server address, as shown below:

   ```
   MOT(config)#ldap server primary <A.B.C.D> port <1-1024>
   ```

   where:

   A.B.C.D is the LDAP server IP address.

   1-1024 is the port number of the LDAP server.

   For example:

   ```
   MOT(config)#ldap server primary 192.168.1.253 port 389
   ```

   Use the following additional options to further define **ldap server** parameters:

Option	Description
ldap server binddn	Distinguished LDAP server name required to bind to this server.
nobinddn	Distinguished LDAP server name not required to bind to this server.
nopassword	Password not required
password	Password

2. To start the LDAP client, use the **ldap client** command in global configuration mode.

   ```
   MOT(config)#ldap client
   ```

Note: If the primary LDAP server has not been specified, the following message appears when attempting to start the LDAP client:

```
Please configure Primary LDAP server address before starting
the client.
```

3. To specify the portion of the LDAP tree where the configuration is located, use the **ldap search-base** command in Global Configuration mode.

   ```
   MOT(config)#ldap search-base <WORD>
   ```

   where:

   WORD is the distiguished location name of entry from which to start a search.

4. Verify that the information was entered correctly by displaying the running configuration in Privileged EXEC mode.

   ```
   MOT#show running-config
   ```

Configuring SNTP

Simple Network Time Protocol (SNTP) provides system time with high accuracy, but it does not provide the complex filtering and statistical mechanisms of the Network Time Protocol (NTP). Configure the local router to operate in client mode with the remote system at the specified address. In this mode, the local router can be synchronized to the remote system, but the remote system never can be synchronized to the local router.

1. Configure the SNTP server with the **sntp server** command on Global Configuration mode.

   ```
   MOT(config)#sntp server {<224.0.1.1> | <A.B.C.D> | <Hostname>}
   ```

   where

   224.0.1.1 is the NTP Multicast server IP address.

   A.B.C.D is the IP address of the server.

   Hostname is the DNS name of the server.

Note: When the server address is 224.0.1.1, the IANA assigned multicast address for NTP, the client operates in anycast mode. It transmits a request to this multicast address and waits for replies. It then *binds* to the first server that replies. All subsequent transactions happen in unicast mode.

For example:

```
MOT(config)#sntp server 192.168.1.253
```

```
MOT(config)#sntp server sntpd.motorola.com
```

2. Authenticate SNTP time sources with the **sntp authenticate** command in Global Configuration mode.

   ```
   MOT(config)#sntp authenticate
   ```

Note: If you configure the BSR to operate in authenticated mode, you must also configure an authentication key and a trusted key.

3. Configure an authentication a key for the trusted time source with the **sntp authentication-key md5** command on Global Configuration mode. You configure SNTP authentication keys so that the BSR can send authenticated packets. The key must be identical between a set of peers sharing the same key number.

   ```
   MOT(config)#sntp authentication-key {<1-4294967295>} md5 <WORD>
   ```

   where:

   1-4294967295 is the SNTP authentication key.

   WORD is the authentication key, which is from 1 to 8 alphanumeric characters.

4. Configure an SNTP broadcast service to listen to SNTP broadcasts with the **sntp broadcast client** command in Global Configuration mode.

   ```
   MOT(config)#sntp broadcast client
   ```

5. Configure an SNTP broadcast delay, which is, with the **sntp broacastdelay** command in Global Configuration mode.

   ```
   MOT(config)#sntp broadcastdelay <1-999999>
   ```

   where:

   1-999999 is the estimated round-trip delay in microseconds.

6. Configure a key number for trusted time sources with the **sntp trusted-key** command in Global Configuration mode. For SNTP, configure the keys you are allowed to use when you configure the BSR to synchronize its time with other systems on the network.

   `MOT(config)#`**sntp trusted-key** <*1-4294967295*>

   where:

   1-4294967295 is the key number for the trusted time sources.

7. To display information about SNTP, use the **show sntp** command in Privileged EXEC mode.

   `MOT#`**show sntp**

8. Verify that the information was entered correctly by displaying the running configuration.

   `MOT#`**show running-config**

Configuring UDP Broadcast Relay

Network hosts occasionally employ UDP broadcasts to determine address, configuration, and name information. If such a host is on a network segment that does not include a server, UDP broadcasts are normally not forwarded. You can configure an interface to forward certain classes of broadcasts to a helper address. You can have more than one helper address per interface. You can specify a UDP destination port to control which UDP services are forwarded.

1. Use the **ip forward-protocol udp** command in Global Configuration mode to enable forwarding of UDP broadcasts for a specific UDP port and specify the protocols to forward and over which ports, as shown below:

   `MOT(config)#`**ip forward-protocol udp** [<*0-65535*> | <**bootpc**> | <**bootps**> | <**domain**> |<**netbios-dgm**> |<**netbios-ns**> | <**tacacs**> | <**tftp**> | <**time**> | <*cr*>]

   where:

   0-65535 is the UDP port number.

   bootpc is the Bootstrap Protocol (BOOTP) client (68).

   bootps is the Bootstrap Protocol (BOOTP) server (67).

domain is the Domain Name Server (DNS, 53).

netbios-dgm is the NetBios datagram service (138).

netbios-ns is the NetBios name service (137).

tacacs is the TAC Access Control System (49).

tftp is the Trivial File Transfer Protocol (69).

time is the Time (37)

cr is a command return enables the forwarding of UDP broadcasts out the default port.

You can forward the following protocols:

For example:

```
MOT(config)#ip forward-protocol udp 35
```

2. Enter the cable interface from Global Configuration mode.

```
MOT(config)#interface cable <x>/<y>
```

where:

x is the CMTS module slot number.

y is the cable interface number.

3. Use the **ip helper-address** command in Interface Configuration mode to specify a destination IP address for forwarding UDP broadcast packets, including BOOTP, as shown below:

```
MOT(config-if)#ip helper-address {<A.B.C.D>}
```

where:

A.B.C.D is the destination IP address.

4. Verify that the information was entered correctly by displaying the running configuration in Privileged EXEC mode.

```
MOT#show running-config
```

Configuring FTP Access

You can configure the BSR to transfer files between systems on the network using the Internet File Transfer Protocol (FTP). FTP is typically used to transfer upgrade files from an FTP server on the network to the BSR. To configure FTP connections on the BSR, you must specify the FTP username and password that the BSR must use when contacting the FTP server.

Follow these steps to configure FTP Access on the BSR:

1. To specify the FTP user name to be used for the FTP connection, use the **ip ftp username** command in Global Configuration mode, as shown below:

Note: An FTP username can contain up to 31 characters.

```
MOT(config)#ip ftp username <WORD>
```

where:

WORD is the FTP user name that is up to 31 characters.

Use the **no ip ftp username** command to delete the entry.

2. To specify the FTP password to be used for the FTP connection, use the **ip ftp password** command in Global Configuration mode, as shown below.

```
MOT(config)#ip ftp password [0 | 7] <LINE>
```

where:

0 indicates a unencrypted password follows.

7 indicates an encrypted password follows.

LINE is the FTP password, which can be up to 31 characters.

Use the **no ip ftp password** command to delete the entry.

Enabling the RADIUS Client on the BSR

Remote Authentication Dial In User Service (RADIUS) provides additional secure remote network access through authentication, authorization and accounting services. The BSR 64000 uses a RADIUS client to authenticate user login information (passwords) stored on the remote RADIUS server.

The RADIUS client feature is off by default. Once the RADIUS client feature is enabled and configured, a user enters a password in their telnet or console session to access the BSR 64000. The BSR 64000 uses the RADIUS client to authenticate this RADIUS encrypted password with a remote RADIUS server. If the RADIUS server validates the password, the user gains access to the BSR 64000.

Configuring the RADIUS Client for Server Communication

Follow these steps to configure the RADIUS client for server communication:

1. Use the **radius-server host auth-port primary** command in Global Configuration mode to specify a primary RADIUS server for RADIUS client requests, as shown below:

   ```
   MOT(config)#radius-server host [<A.B.C.D> | <Hostname>] auth-port <0-65535> primary
   ```

   where:

   A.B.C.D is the IP address of the remote RADIUS server.

   Hostname is the Hostname of the remote RADIUS server.

   0-65535 is the optionally defined UDP port for the RADIUS authentication server. The default port is 1812.

   primary specifies the server as the primary RADIUS server.

2. Use the **radius-server host** command in Global Configuration mode to specify a secondary or back-up RADIUS server for RADIUS client requests, as shown below:

   ```
   MOT(config)#radius-server host [<A.B.C.D> | <Hostname>] [auth-port <0-65535> | <cr>]
   ```

   where:

   A.B.C.D is the IP address of the remote RADIUS server.

Hostname is the hostname of the remote RADIUS server.

0-65535 is the optionally defined UDP port for the RADIUS authentication server. The default port is 1812.

cr is a command return that configures the RADIUS server host without a UDP port designation.

3. Use the **radius-server key** command in Global Configuration mode to define the shared encryption key that is exchanged between the RADIUS server and BSR RADIUS client, as shown below:

Note: It is recommended that the authentication key text string be more than 22 characters in length.

```
MOT(config)#radius-server key <WORD>
```

where:

WORD is the shared encryption key text.

Caution: Ensure that the RADIUS server authentication key on the BSR is the same as the RADIUS server authentication key on your RADIUS server. If the keys are mismatched, communication does not occur between the BSR and RADIUS server.

Use the following options to change the default RADIUS server settings:

- Use the **radius-server retransmit** command in Global Configuration mode to specify the number of retry attempts to get a response from an active RADIUS server, as shown below:

```
MOT(config)#radius-server retransmit {<0-100>}
```

where:

0-100 is the number of retransmissions. The default is 3 retransmissions.

- Use the **radius-server timeout** command in Global Configuration mode to configure the wait time interval for when there is no response from the server before retransmitting to the RADIUS server, as shown below:

  ```
  MOT(config)#radius-server timeout <0-1000>
  ```

  where:

  > *0-1000* is the wait time interval in seconds. The default value is 5 seconds.

Configuring RADIUS Client Access

Before you configure RADIUS client access parameters, ensure that user password parameters are configured on the BSR 64000 and the RADIUS server. Refer to "Accessing the CLI to Set System Passwords" on page 3-3 for more information.

Note: User password attributes in the RADIUS request sent from the RADIUS client on the BSR to the RADIUS server are encrypted.

Use one or more of the following options to enable and configure the RADIUS Client feature:

1. Use the **telnet authentication radius** command in Global Configuration mode to enable RADIUS client authentication for telnet session access to the BSR, as shown below:

   ```
   MOT(config)#telnet authentication radius telnet [local-password]
   ```

   where:

   > **local-password** allows password authentication by a locally configured password if there is no response from the RADIUS server. If the RADIUS client is not configured with the **local-password** command argument, access to the BSR is denied if there is no response from the RADIUS server.

2. If you are experiencing failed telnet login authentications, use the **telnet authentication radius fail-messege** command in Global Configuration mode to display failed radius client logins and authentications, as shown below:

   ```
   MOT(config)#telnet authentication radius fail-messege [<LINE> | <cr>]
   ```

where:

LINE is the text message for the failed login and authentication.

cr is a command return specifies the default failed loging and authentication message.

3. Use the **console authentication radius username** command in Global Configuration mode to configure a username for RADIUS client authentication for console session access to the BSR, as shown below:

   `MOT(config)#`**console authentication radius username** {<*WORD*>}

   where:

   WORD is the username.

4. Use the **console authentication radius local-password** command in Global Configuration mode to enable RADIUS client authentication for console session access to the BSR, as shown below. This command allows you to configure a user name for RADIUS access, use of a locally set password or both.

   `MOT(config)#`**console authentication radius** [**local-password** | **username** <*WORD*>]

   where:

   local-password allows password authentication by a locally configured password if there is no response from the RADIUS server. If the RADIUS client is not configured with the **local-password** command argument, access to the BSR is denied if there is no response from the RADIUS server.

   username is used for authentication.

   WORD

5. Use the **enable authentication radius** command in Global Configuration mode to enable RADIUS client authentication for Privileged EXEC mode access to the BSR, as shown below:

   `MOT(config)#`**enable authentication radius** [**local-password**]

   where:

local-password allows password authentication by a locally configured password if there is no response from the RADIUS server. If the RADIUS client is not configured with the **local-password** command argument, access to the BSR is denied when there is no response from the RADIUS server.

Viewing RADIUS Client Statistics

Use the **show ip traffic** command in Privileged EXEC mode to display packet statistics for communication between the RADIUS client and RADIUS server, as shown below:

```
MOT#show ip traffic
```

Downloading Software

The following sections show different methods for downloading software on the BSR 64000:

- Before You Download Software
- Downloading Image Files to NVRAM on the SRM
- Downloading Image Files to Flash Memory on the SRM
- Downloading Software to All Modules
- Downloading Software to a Specific Module

Before You Download Software

Follow these steps before downloading software to the BSR 64000:

1. Use the **dir** command in Privileged EXEC mode to ensure that you have enough memory space in Nonvolatile Random Access Memory (NVRAM), which is located on the SRM module, for the new software, as shown below:

   ```
   MOT#dir
   ```

2. If you need to free additional memory space in NVRAM by deleting any unwanted files, use the **delete nvram:** command in Privileged EXEC mode, as shown below:

Caution: Ensure that you do not delete the current start-up configuration. Also ensure that you do not delete any necessary application, or boot image files.

MOT#**delete nvram:**<*file*>

where:

file is an application or boot image file.

For example:

MOT#**delete nvram:image_file.Z**

3. Press the **Enter** key when asked for confirmation.

For example:

MOT#**delete nvram:image_file.Z ? [confirm]**

4. In order to download files to the BSR 64000, you must properly configure your FTP or TFTP server and verify that your local FTP or TFTP server is running and configured properly by doing the following:

   a. Check for the correct file names and ensure that these files are located in the proper directory on the FTP or TFTP server.

   b. Ensure that the proper IP address is configured for your TFTP or FTP server.

5. Use the **ping** command in Privileged EXEC mode to verify the connectivity status of your TFTP or FTP server, as shown below.

MOT#**ping** [<*A.B.C.D*> | <*Hostname*>]

where

ip-address is the IP address of the FTP or TFTP server.

Hostname is the DNS hostname of the FTP or TFTP server.

6. Ensure that the correct FTP username is configured on the BSR 64000 for communication with the FTP server. If the required FTP user name is not displayed in the running configuration or is incorrect, use the **ip ftp username** command in Global Configuration mode as shown below:

Note: If you have a TFTP server, you do not need to set a user name or password on the BSR.

```
MOT(config)#ip ftp username <WORD>
```

where:

WORD is the username configured on the FTP server

7. Ensure that the correct FTP password is configured on the BSR 64000 for communication with the FTP server. If the required FTP password is not displayed in the running configuration or is incorrect, use the **ip ftp password** command in Global Configuration mode as shown below:

```
MOT(config)#ip ftp password <LINE>
```

where:

0 indicates that the following password is unencrypted (clear text).

7 indicates that the following password is encrypted.

LINE is the password configured on the FTP server.

Downloading Image Files to NVRAM on the SRM

Both boot and application image files can be downloaded to NVRAM on the SRM using the FTP or TFTP file transfer process.

Follow these steps to download an image file to NVRAM on the SRM:

Note: The following steps describe the process of transferring the new image files from an FTP server to the SRM. If you are using FTP to transfer the image files, ensure that the FTP username and password are set correctly on the BSR 64000 using the **ip ftp username** and **ip ftp password** commands. If you are using TFTP to transfer the image files, a username and password are not necessary and the **copy tftp: nvram:** command can be substituted for the **copy ftp: nvram:** command.

1. To download an image file to NVRAM, use the **copy ftp: nvram:** command in Privileged EXEC mode and press the **Enter** key, as shown below:

   ```
   MOT#copy ftp: nvram:
   ```

2. Enter the IP address or DNS name of the remote FTP or TFTP server at the **Address or name of remote host** prompt and press the **Enter** key.

   For example:

   ```
   Address or name of remote host[]? 10.10.10.1
   ```

3. Enter the full path from the FTP root directory and the new application or boot image file name after the **Source file name** prompt and press the **Enter** key.

   ```
   Source file name [ ]?
   ```

   For example:

   ```
   Source file name [ ]? /pub/image_file.Z
   ```

4. The **Destination file name** prompt displays with the new file name. Press the **Enter** key to accept the new file name in NVRAM. For example:

   ```
   Destination file name [ image_file.Z ]?
   ```

   The file is successfully copied to NVRAM on the SRM module.

Note: You can optionally rename the image file name stored in NVRAM on the SRM module. If you decide to enter a new file name in NVRAM on the SRM, enter the new file name after the **Destination file name** prompt. For example:

```
Destination file name [ image_file.Z ] ? <new file name>
```

5. Compare each image file size (in bytes) in NVRAM on the SRM to the original size of each image file size on the server. To view the new image files in NVRAM on the SRM, use the **dir** command in Privileged EXEC mode as shown below:

   ```
   MOT#dir
   ```

   The following command output displays:

6. If the image file byte counts in NVRAM on the SRM match the image file byte counts on the server, the image files on the SRM have been copied successfully.

Downloading Image Files to Flash Memory on the SRM

Both boot and application image files can be downloaded to flash memory on the SRM using the FTP or TFTP file transfer process.

Follow these steps to download an image file to flash memory on the SRM:

Note: The following steps describe the process of transferring the new image files from an FTP server to the SRM. If you are using FTP to transfer the image files, ensure that the FTP username and password are set correctly on the BSR 64000 using the **ip ftp username** and **ip ftp password** commands. If you are using TFTP to transfer the image files, a username and password are not necessary and the **copy tftp: nvram:** command can be substituted for the **copy ftp: nvram:** command.

1. To download an image file to flash memory, use the **copy ftp: flash:** command in Privileged EXEC mode and press the **Enter** key, as shown below:

   ```
   MOT#copy ftp: flash:
   ```

2. Enter the IP address or DNS name of the remote FTP or TFTP server at the **Address or name of remote host** prompt and press the **Enter** key.

   For example:

   ```
   Address or name of remote host[]? 10.10.10.1
   ```

3. Enter the full path from the FTP root directory and the new application or boot image file name after the **Source file name** prompt and press the **Enter** key.

   ```
   Source file name [ ]?
   ```

   For example:

   ```
   Source file name [ ]? /pub/image_file.Z
   ```

4. The **Destination file name** prompt displays with the new file name. Press the **Enter** key to accept the new file name in flash memory.

   For example:

   ```
   Destination file name [ image_file.Z ]?
   ```

   The file is successfully copied to flash memory on the SRM module.

Note: You can optionally rename the image file name stored in NVRAM on the SRM module. If you decide to enter a new file name in NVRAM on the SRM, enter the new file name after the **Destination file name** prompt. For example:

```
Destination file name [ image_file.Z ] ? <new file name>
```

5. Compare each image file size (in bytes) in flash memory on the SRM to the original size of each image file size on the server. To view the new image files in flash memory on the SRM, use the **dir flash:** command in Privileged EXEC mode as shown below:

   ```
   MOT#dir flash:
   ```

   If the image file byte counts in flash memory on the SRM match the image file byte counts on the server, the image files on the SRM have been copied successfully.

Downloading Software to All Modules

Use the **download runtime nvram:** command in Privileged EXEC mode to download operating software contained in an application image file or boot image to all modules installed in the BSR 64000 from Nonvolatile Random Access Memory (NVRAM), as shown below:

Note: The designated software loads the next time the BSR 64000 is rebooted.

```
MOT#download runtime nvram:<file>
```

where:

file is the name of the operating image intended for download, such as the application image file or boot image file.

For example:

MOT#**download runtime nvram:image_file.Z**

Downloading Software to a Specific Module

The **download slot** command lets you specify an image stored in either Nonvolatile Random Access Memory (NVRAM) or flash memory, and download it to a resource module installed in a specified chassis slot of the BSR 64000.

Note: The designated software loads the next time the BSR 64000 is rebooted.

The following options are used to download software to a specific module:

- If you want to download the buffer manager FPGA file stored in the application image file (also known as the archive file) to a particular module, use the **download slot bm** command in Privileged EXEC mode as shown below:

  MOT#**download slot** <*NUM*> **bm**

  where:

  NUM is the module slot number of any available module.

- If you want to download the CMTS FPGA file stored in the application image file to a particular module, use the **download slot cmts-fpga** command in Privileged EXEC mode as shown below:

  MOT#**download slot** <*NUM*> **cmts-fpga**

  where:

  NUM is the module slot number of any available module.

- If you want to download the executable file stored in the application image file to a particular module, use the **download slot elf** command in Privileged EXEC mode as shown below:

  MOT#**download slot** <*NUM*> **elf**

  where:

  NUM is the module slot number of any available module.

- If you want to download the buffer manager FPGA file, CMTS FPGA file, and executable file stored in the application image file to a particular module, use the **download slot** command in Privileged EXEC mode as shown below:

  MOT#**download slot** {<*NUM*> <*cr*>}

  where:

  NUM is the module slot number of any available module.

  cr is a command return.

- If you want to download an image file from flash memory to a particular module, use the **download slot flash:** command in Privileged EXEC mode as shown below:

  MOT#**download slot** <*NUM*> **flash:**<*file*>

  where:

  NUM is the module slot number of any available module.

  file is the name of the operating image intended for download, such as the application image file or boot image file.

- If you want to download an image file from NVRAM to a particular module, use the **download slot nvram:** command in Privileged EXEC mode as shown below:

  MOT#**download slot** <*NUM*> **nvram:**<*file*>

  where:

  NUM is the module slot number of any available module.

  file is the name of the operating image intended for download, such as the application image file or boot image file.

Specifying the System Image File Boot Location

Follow these steps to specify the system image file for use when starting the BSR:

1. The **show boot** command can be accessed from all CLI modes except User EXEC mode. Use the **show boot** command to determine the current boot location for the application image. For example:

   ```
   MOT#show boot
   ```

   Boot location currently set to nvram:image_file.Z

2. Use the **boot system** command in Privileged EXEC mode only to indicate which system image file the BSR uses at the system startup.

   ```
   MOT#boot system {flash: | ftp: | nvram: | tftp:} <filename>
   ```

   where:

   flash: Boot from the file stored in flash memory.

   ftp: Boot from the file stored on the File Transfer Protocol (FTP) server.

   nvram: Boot from the file stored in nonvolatile random access memory NVRAM.

   tftp: Boot from the file stored on the Trivial File Transfer Protocol (TFTP) server.

   filename is the file name from which to boot.

   For example:

   ```
   MOT#boot system NVRAM:image_file.Z
   ```

Specifying System Information

The following sections are used to specify system information for the BSR 64000 for management and inventory purposes:

- Configuring SRM and Chassis Alias Information
- Configuring SRM and Chassis Asset ID Information
- Configuring SRM and Chassis Serial Number Information

Configuring SRM and Chassis Alias Information

To configure your alias name for the SRM module, use the **srm alias** command in Privileged EXEC mode, as shown below:

```
MOT#srm alias <string>
```

where:

string is the SRM alias name.

To configure your alias name for the BSR 64000 chassis, use the **chassis alias** command in Privileged EXEC mode, as shown below:

```
MOT#chassis alias <string>
```

where:

string is the BSR 64000 alias name.

Note: Enclose the alias name within quotes if the string contains spaces in the text.

Configuring SRM and Chassis Asset ID Information

To configure your organization's asset ID number that is assigned to the SRM module, use the **srm assetid** command in Privileged EXEC mode, as shown below:

```
MOT#srm assetid <string>
```

where:

string is the SRM asset ID number.

To configure your organization's asset ID number that is assigned to your BSR 64000, use the **chassis assetid** command in Privileged EXEC mode, as shown below:

```
MOT#chassis assetid <string>
```

where:

string is the BSR 64000 asset ID number.

Configuring SRM and Chassis Serial Number Information

To configure the serial number assigned to your SRM module, use the **srm serial-num** command in Privileged EXEC mode, as shown below:

MOT#**srm serial-num** <*string*>

where:

string is the SRM module serial number.

To configure the serial number assigned to your BSR 64000, use the **chassis serial-num** command in Privileged EXEC mode, as shown below:

MOT#**chassis serial-num** <*string*>

where:

string is the BSR 64000 serial number.

Saving and Viewing Your Configuration

Saving the current running configuration to nonvolatile random access memory (NVRAM) is done to prevent your current configuration from being lost the next time the BSR is rebooted. Always save configuration changes.

Follow these steps to save the current running configuration:

1. To copy the current system configuration to the system startup configuration, use the **copy running-config startup-config** command in Privileged EXEC mode as shown below:

   MOT#**copy running-config startup-config**

2. To verify that the changes you made were implemented, use the **show running-config** command in Privileged EXEC mode, as shown below:

   MOT#**show running-config**

The configuration parameters that you have set should appear in the **show running-config** command output.

Reseting BSR Modules

One or all modules on the BSR need to be reset for boot image upgrades, if significant software errors occur on a particular module or accross several modules on the BSR, or in instances where POS modules need to be synchronized.

Use the **reset** command in Privileged EXEC mode to reset one or all modules on the BSR, as shown below:

`MOT#`**reset** [**all** | **slot** {*<0-6, 8-15>*}]

where:

> **all** resets all modules in the BSR chassis.
>
> **slot** indicates that a specific module slot is reset.
>
> *0-6, 8-15* identifies the module slot number to be reset.

For example, if you are resetting one module, the following syntax applies:

```
MOT#reset slot 3
```

For example, if you are resetting all modules on the BSR, the following syntax applies:

```
MOT#reset all
```

Monitoring the System

The following sections provide information about common system management show commands used to examine system processes:

- Displaying System Processing Information
- Displaying System Memory Information
- Displaying the System Version Information
- Displaying System Buffer Information

- Gathering System Information
- Displaying Module Hardware Information

Displaying System Processing Information

In the BSR, *process* and *thread* are used interchangeably and mean an independent thread of execution.

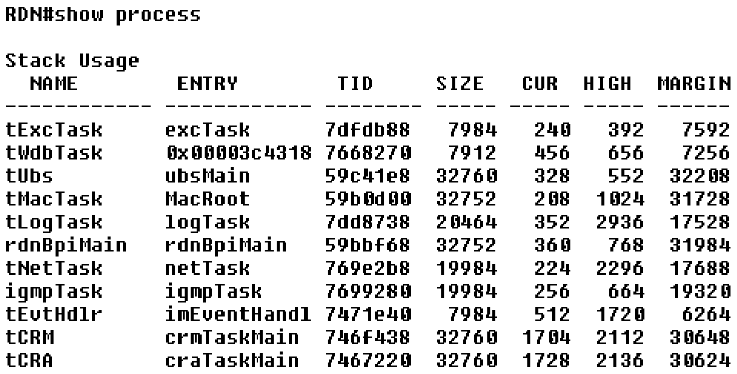
Use the following options to view system processing information:

- To display information about all active processes on the BSR, use the **show process** command in Privileged EXEC mode, as shown below:

  MOT#**show process**

  Figure 3-1 shows a sample of the **show process** output information:

Note: In the BSR, *process* and *task* are interchangeable and mean an independent thread of execution.



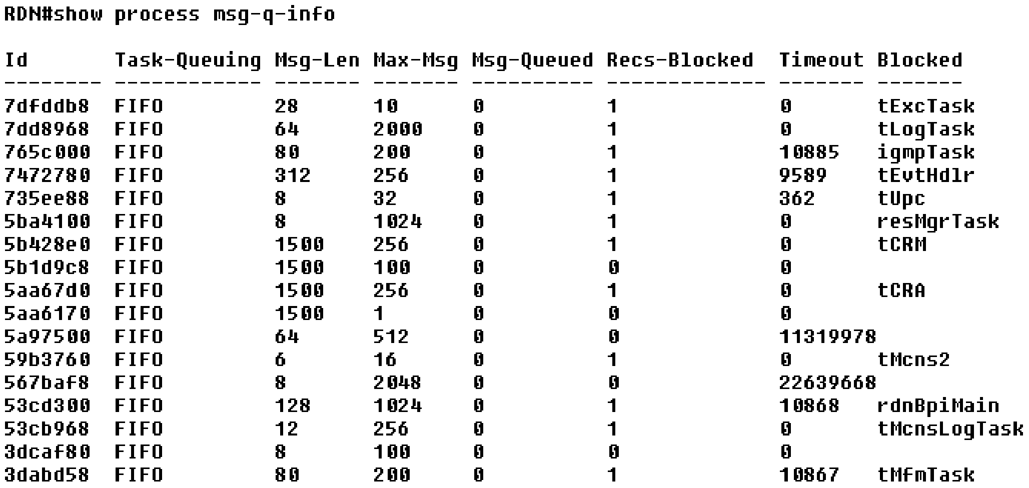
```
RDN#show process

Stack Usage
  NAME         ENTRY          TID      SIZE   CUR   HIGH  MARGIN
------------ ------------ -------- ----- ----- ----- ------
tExcTask     excTask      7dfdb88   7984   240   392   7592
tWdbTask     0x00003c4318 7668270   7912   456   656   7256
tUbs         ubsMain      59c41e8  32760   328   552  32208
tMacTask     MacRoot      59b0d00  32752   208  1024  31728
tLogTask     logTask      7dd8738  20464   352  2936  17528
rdnBpiMain   rdnBpiMain   59bbf68  32752   360   768  31984
tNetTask     netTask      769e2b8  19984   224  2296  17688
igmpTask     igmpTask     7699280  19984   256   664  19320
tEvtHdlr     imEventHandl 7471e40   7984   512  1720   6264
tCRM         crmTaskMain  746f438  32760  1704  2112  30648
tCRA         craTaskMain  7467220  32760  1728  2136  30624
```

Figure 3-1 show process Command Output

- Issue the **show process msg-q-info** command Privileged EXEC mode to display information about current message queues, as shown below:

MOT#**show process msg-q-info**

Figure 3-2 shows the **show process msg-q-info** output information:



```
RDN#show process msg-q-info

Id        Task-Queuing Msg-Len Max-Msg Msg-Queued Recs-Blocked  Timeout Blocked
--------  ------------ ------- ------- ---------- ------------- ------- -------
7dfddb8   FIFO         28      10      0          1             0       tExcTask
7dd8968   FIFO         64      2000    0          1             0       tLogTask
765c000   FIFO         80      200     0          1             10885   igmpTask
7472780   FIFO         312     256     0          1             9589    tEvtHdlr
735ee88   FIFO         8       32      0          1             362     tUpc
5ba4100   FIFO         8       1024    0          1             0       resMgrTask
5b428e0   FIFO         1500    256     0          1             0       tCRM
5b1d9c8   FIFO         1500    100     0          0             0
5aa67d0   FIFO         1500    256     0          1             0       tCRA
5aa6170   FIFO         1500    1       0          0             0
5a97500   FIFO         64      512     0          0             11319978
59b3760   FIFO         6       16      0          1             0       tMcns2
567baf8   FIFO         8       2048    0          0             22639668
53cd300   FIFO         128     1024    0          1             10868   rdnBpiMain
53cb968   FIFO         12      256     0          1             0       tMcnsLogTask
3dcaf80   FIFO         8       100     0          0             0
3dabd58   FIFO         80      200     0          1             10867   tMfmTask
```

Figure 3-2 show process msg-q-info Command Output

- Issue the **show process** command with the **stack** keyword in Privileged EXEC mode to display the size, current usage, and highest usage of each process stack, as shown below:

  MOT#**show process stack** [**procID** | **procName**]

  where:

  > **procID** is the task ID number in decimal or hexidecimal form. 0x is required for hexidecimal form.
  >
  > **procName** is the task name.

- Issue the **show process memory** command keyword in Privileged EXEC mode to display information about memory usage, as shown below:

  MOT#**show process memory**

- Issue the **show process memory slot** command in Privileged EXEC mode to display information about CMTS memory usage, as shown below:

  MOT#**show process memory slot** <*NUM*>

where:

NUM is the DOCSIS module slot number.

- Issue the **show process cpu** command in Privileged EXEC Mode to display information about CPU utilization by each process, as shown below:

  MOT#**show process cpu**

Note: The total utilization is approximate and may not total 100 per cent.

- To display information about CMTS CPU utilization by each process, use the **show process cpu slot** command in Privileged EXEC Mode, as shown below:

  MOT#**show process cpu slot** <*NUM*>

  where:

  NUM is the DOCSIS module slot number.

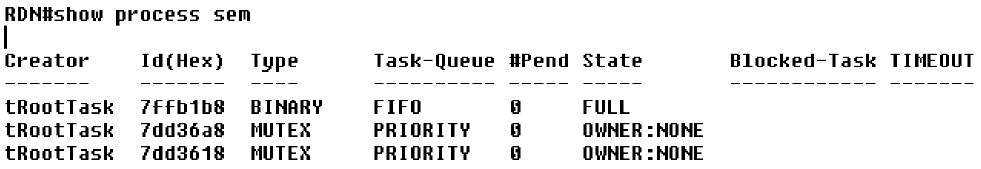
- To restart the CPU utilization measurement process, use the **show process** command with the **cpu restart** options in Privileged EXEC Mode, as shown below:

  MOT#**show process cpu restart**

- Use the **show process sem** command in Privileged EXEC mode display information about the Semaphore ID number on which process is waiting, as shown below:

  MOT#**show process sem**

  Figure 3-3 shows the **show process sem** command output information:



```
RDN#show process sem
|
Creator    Id(Hex)  Type      Task-Queue #Pend State      Blocked-Task TIMEOUT
-------    -------  ----      ---------- ----- -----      ------------ -------
tRootTask  7ffb1b8  BINARY    FIFO        0    FULL
tRootTask  7dd36a8  MUTEX     PRIORITY    0    OWNER:NONE
tRootTask  7dd3618  MUTEX     PRIORITY    0    OWNER:NONE
```

Figure 3-3 show process sem Command Output

Displaying System Memory Information

Follow these options to display BSR system information:

- Issue the **show memory** command in Privileged EXEC mode to show the number of blocks of memory, the hexadecimal address of each block of memory, and the size of each block of memory in bytes, as shown below:

  MOT#**show memory information** [**brief** | **slot** <*NUM*> | <|> | <*cr*>]

  where:

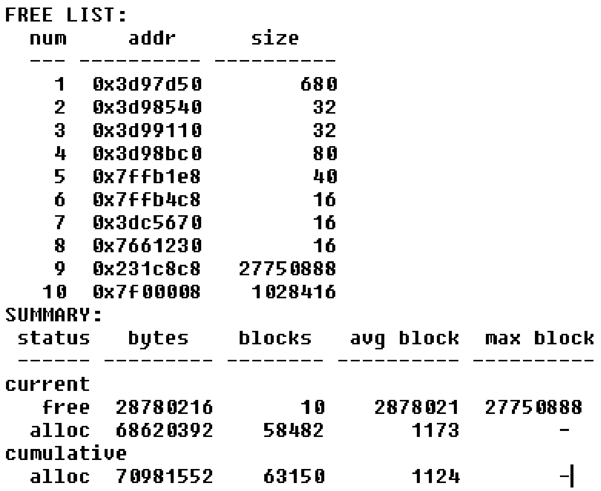
  brief displays only the summary.

  slot displays memory information for the BSR module slot only.

  | indicates that output modifiers can be used.

  cr is a command return that displays all BSR system memory information.

  Figure 3-4 shows a partial **show memory information** command output:



```
FREE LIST:
  num     addr       size
  --- ---------- ----------
    1  0x3d97d50        680
    2  0x3d98540         32
    3  0x3d99110         32
    4  0x3d98bc0         80
    5  0x7ffb1e8         40
    6  0x7ffb4c8         16
    7  0x3dc5670         16
    8  0x7661230         16
    9  0x231c8c8   27750888
   10  0x7f00008    1028416
SUMMARY:
 status   bytes     blocks   avg block  max block
 ------ --------- -------- ---------- ----------
current
   free  28780216       10    2878021   27750888
  alloc  68620392    58482       1173          -
cumulative
  alloc  70981552    63150       1124          -|
```

Figure 3-4 show memory information Command Output

- Issue the **show memory** command in Privileged EXEC mode to display the starting address where memory is dumped in hexadecimal notation, as shown below:

```
MOT#show memory <address> <size>
```

where:

address is the starting memory address expressed in hexadecimal notation to dump memory.

size is the number of bytes to dump.

- Issue the **show memory fastpath** [**brief**] command in Privileged EXEC mode to display the number of bytes used to program the HSIM FastPath, as shown below:

```
MOT#show memory fastpath [brief]
```

where:

brief displays the summary only.

Displaying the System Version Information

Issue the **show version** command in Privileged EXEC mode to display the BSR system software and hardware version information for all modules, as shown below:

```
MOT#show version
```

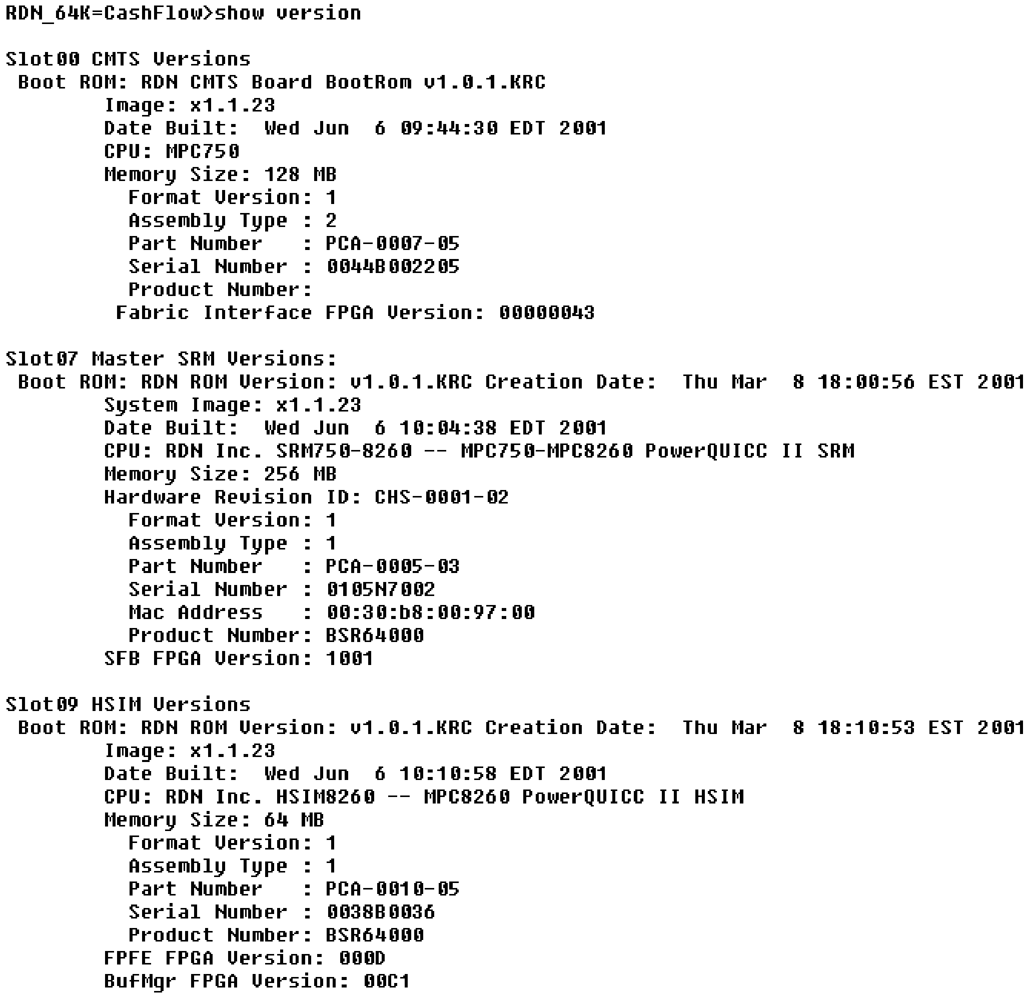
Issue the **show version slot** command in Privileged EXEC mode to display the BSR system software and hardware version information for a particular module, as shown below:

```
MOT#show version slot <NUM>
```

where:

NUM is the slot number for which information is displayed; valid values are from 0 to 15.

Figure 3-5 displays the **show version** command output, which presents the BSR system version information:



```
RDN_64K=CashFlow>show version

Slot00 CMTS Versions
 Boot ROM: RDN CMTS Board BootRom v1.0.1.KRC
        Image: x1.1.23
        Date Built:  Wed Jun  6 09:44:30 EDT 2001
        CPU: MPC750
        Memory Size: 128 MB
          Format Version: 1
          Assembly Type : 2
          Part Number   : PCA-0007-05
          Serial Number : 0044B002205
          Product Number:
         Fabric Interface FPGA Version: 00000043

Slot07 Master SRM Versions:
 Boot ROM: RDN ROM Version: v1.0.1.KRC Creation Date:  Thu Mar  8 18:00:56 EST 2001
        System Image: x1.1.23
        Date Built:  Wed Jun  6 10:04:38 EDT 2001
        CPU: RDN Inc. SRM750-8260 -- MPC750-MPC8260 PowerQUICC II SRM
        Memory Size: 256 MB
        Hardware Revision ID: CHS-0001-02
          Format Version: 1
          Assembly Type : 1
          Part Number   : PCA-0005-03
          Serial Number : 0105N7002
          Mac Address   : 00:30:b8:00:97:00
          Product Number: BSR64000
        SFB FPGA Version: 1001

Slot09 HSIM Versions
 Boot ROM: RDN ROM Version: v1.0.1.KRC Creation Date:  Thu Mar  8 18:10:53 EST 2001
        Image: x1.1.23
        Date Built:  Wed Jun  6 10:10:58 EDT 2001
        CPU: RDN Inc. HSIM8260 -- MPC8260 PowerQUICC II HSIM
        Memory Size: 64 MB
          Format Version: 1
          Assembly Type : 1
          Part Number   : PCA-0010-05
          Serial Number : 0038B0036
          Product Number: BSR64000
        FPFE FPGA Version: 000D
        BufMgr FPGA Version: 00C1
```

Figure 3-5 show version Command Output

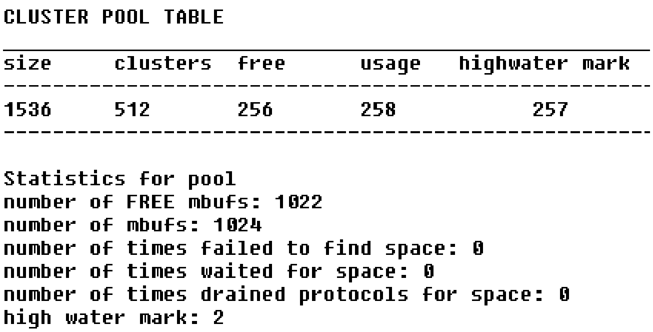
Displaying System Buffer Information

Follow these options to evaluate system buffer information:

- Issue the **show buffer** command in Privileged EXEC mode to display information about the way in which the BSR is buffering data, as shown below:

  MOT#**show buffer**

Figure 3-6 displays a sample of the **show buffer** command output:



```
CLUSTER POOL TABLE
_____________________________________________________
size      clusters  free      usage    highwater mark
-----------------------------------------------------
1536      512       256       258           257
-----------------------------------------------------

Statistics for pool
number of FREE mbufs: 1022
number of mbufs: 1024
number of times failed to find space: 0
number of times waited for space: 0
number of times drained protocols for space: 0
high water mark: 2
```

Figure 3-6 show buffer Command Output

- Issue the **show buffer all** command in Privileged EXEC mode to view all memory buffer pools, as shown below:

  MOT#**show buffer all**

- Issue the **show buffer icp** command in Privileged EXEC mode to view chassis control messages in the ICP pool, as shown below:

  MOT#**show buffer icp**

- Issue the **show buffer** command in Privileged EXEC mode to view the network pool, where network data transfer information for the stack is located, as shown below:

  MOT#**show buffer network**

- Issue the **show buffer pool** command in Privileged EXEC mode to view statistics for how each memory pool is used, as shown below:

  MOT#**show buffer pool**

- Issue the **show buffer system** command in Privileged EXEC mode to view system physical structures, such as the number of sockets, routes, interface addresses, PCB, and multicast addresses in the system pool, as shown below:

  ```
  MOT#show buffer system
  ```

Gathering System Information

The following sections discuss how to gather system information and learn the current status of the BSR:

- Viewing SRM and Chassis Alias Information
- Viewing the SRM and Chassis Asset ID Information
- Viewing the SRM and Chassis Serial Number Information
- Viewing the Chassis Status

Viewing SRM and Chassis Alias Information

Issue the **show srm alias** command in Privileged EXEC mode to show the alias name for the SRM module, as shown below:

```
MOT#show srm alias
```

To show the alias name for the BSR 64000 chassis, use the **show chassis alias** command in Privileged EXEC mode, as shown below:

```
MOT#show chassis alias
```

Viewing the SRM and Chassis Asset ID Information

To view the asset ID number assigned to the SRM, use the **show srm assetid** command in Privileged EXEC mode, as shown below:

```
MOT#show srm assetid
```

To view the asset ID number assigned to your BSR 64000, use the **show chassis assetid** command in Privileged EXEC mode, as shown below:

```
MOT#show chassis assetid
```

Viewing the SRM and Chassis Serial Number Information

To view the serial number assigned to your SRM, use the **show srm serial-num** command in Privileged EXEC mode, as shown below:

```
MOT#show srm serial-num
```

To view the serial number assigned to your BSR 64000, use the **show chassis serial-num** command in Privileged EXEC mode, as shown below:

```
MOT#show chassis serial-num
```

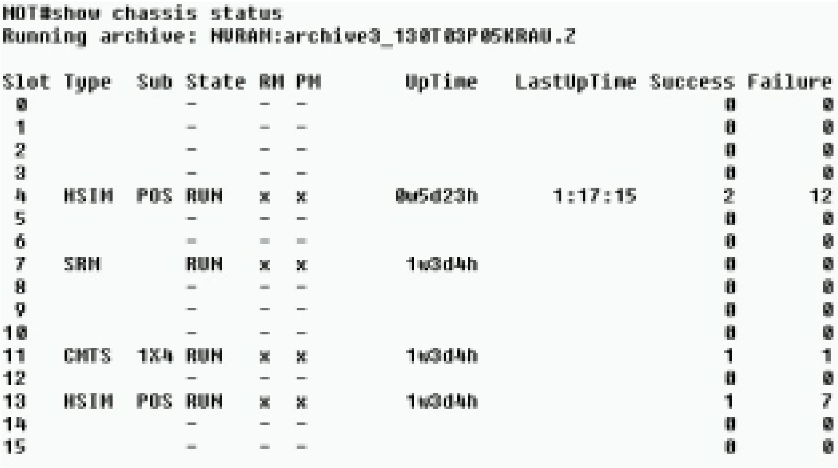
Viewing the Chassis Status

The **show chassis status** command is an important diagnostic tool for learning the operational status of the individual modules and upper and lower fan trays. This command also allows you to determine where modules are populated on the BSR 64000.

To display chassis status information, use the **show chassis status** command in Privileged EXEC mode, as shown below:

```
MOT#show chassis status
```

Figure 3-7 displays the **show chassis status** command output for a fully operational BSR 64000:



```
MOT#show chassis status
Running archive: NVRAM:archive3_130T03P05KRAU.Z

Slot Type  Sub State RM PM       UpTime   LastUpTime Success Failure
 0             -     -  -                                  0       0
 1             -     -  -                                  0       0
 2             -     -  -                                  0       0
 3             -     -  -                                  0       0
 4   HSIM  POS RUN   x  x       0w5d23h     1:17:15        2      12
 5             -     -  -                                  0       0
 6             -     -  -                                  0       0
 7   SRM       RUN   x  x        1w3d4h                    0       0
 8             -     -  -                                  0       0
 9             -     -  -                                  0       0
10             -     -  -                                  0       0
11   CMTS  1X4 RUN   x  x        1w3d4h                    1       1
12             -     -  -                                  0       0
13   HSIM  POS RUN   x  x        1w3d4h                    1       7
14             -     -  -                                  0       0
15             -     -  -                                  0       0
```

Figure 3-7 show chassis status Command Output

Table 3-4 describes the BSR 64000 chassis output fields:

Table 3-4 BSR 64000 Chassis Status Field Descriptions

Field	Description
Slot	Module slot number from 0 to 15
Type	The type of module inserted into the BSR 64000. HSIM indicates either the 8-port Fast Ethernet, POS or Gigabit Ethernet Network Interface modules (NIMs). CMTS indicates the DOCSIS module. SRM indicates the Supervisory Routing Module.
State	Indicates the current operational state of the module. **RUN** indicates that the module is fully operational. **Flash** indicates that the module is updating its FLASH memory. **boot** indicates that the module is currently in a boot state.
RM	Resource module. An **x** indicates that this module is operational. A **-** (dash) indicates that this module is not currently operational.
PM	Physical module (IO module). An **x** indicates that this module is operational. A **-** (dash) indicates that this module is not currently operational.

Table 3-4 BSR 64000 Chassis Status Field Descriptions

Field	Description
UpTime	If the system clock has been set using the **clock set** command, the UpTime field displays the amount of time that the module has been operational. The time is expressed in hh:mm:ss format.
LastUpTime	If the module is down, the last operational time for the module displays.
Success	The module booted successfully.
Failure	The module failed the boot process.
Alarms	Indicates that alarms are configured for the top and bottom fan trays.
Status	An **x** appears in the status field if a fan tray becomes disabled.
Disabled	The disabled column displays no output information at this time.
Priority	Both fan trays have a critical priority assigned to them.

Displaying Module Hardware Information

The **show controllers** command displays detailed hardware and configuration information for each module on installed in the BSR 64000 chassis.

Note: Refer to Displaying Physical SONET Link and Alarm Information for more information about the **show controllers pos** command.

- Use the **show controllers cable** command to display the following CMTS module information:

  MOT#**show controllers cable** {<*x*>/<*y*>} [**upstream** <*port*> | **downstream** <*port*> | **mac** | <*cr*>]

  where:

  x is the CMTS module slot number.

  y is the cable interface number, which is **0**.

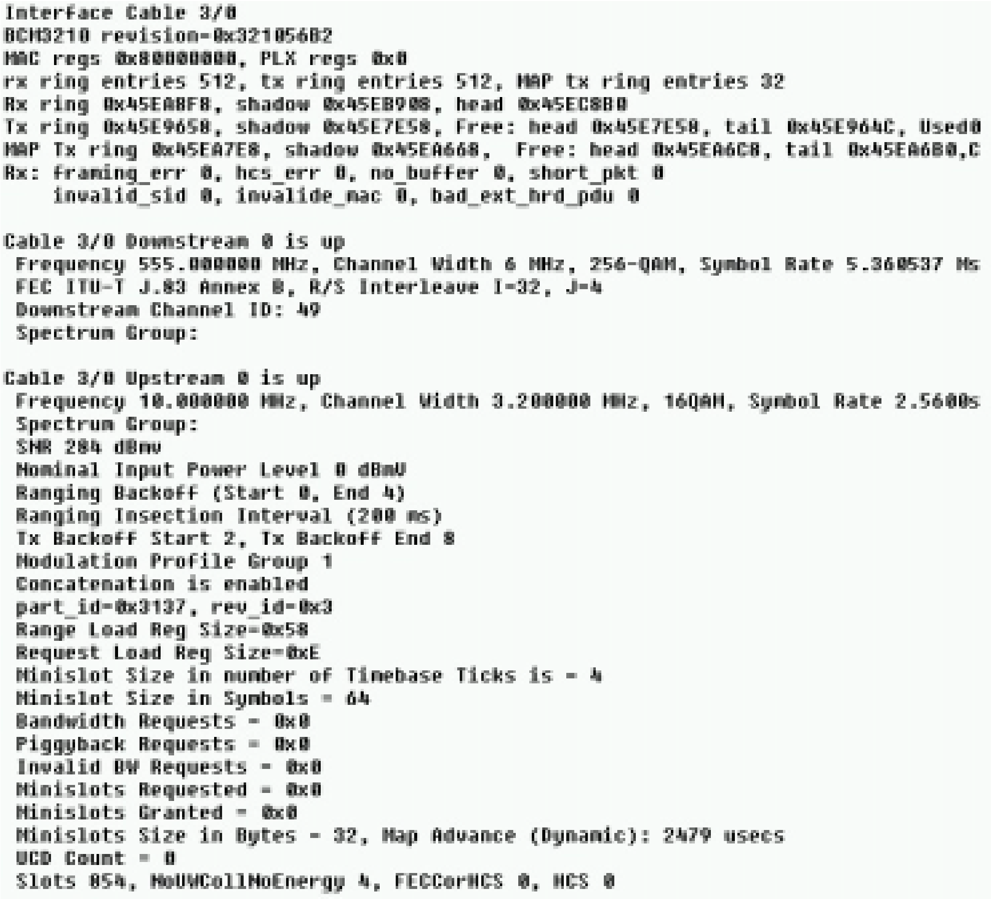
upstream <*port*> displays information for an upstream port including the upstream modulation type, channel width, frequency, and modulation profile information (i.e minislots, interleave, preamble, etc).

downstream <*port*> displays information for a downstream port including downstream modulation type, frequency (label), and symbol rate.

mac displays MAC layer information about the cable interface.

cr a command return displays RF signal information, the type of hardware installed, FEC information for both corrected and uncorrected packets, the spectrum group and the status of the cable interface.

Figure 3-8 displays **show controllers cable** command output for the cable interface, downstream port and upstream ports.



```
Interface Cable 3/0
BCM3210 revision=0x32105682
MAC regs 0x80000000, PLX regs 0x0
rx ring entries 512, tx ring entries 512, MAP tx ring entries 32
Rx ring 0x45EA8F8, shadow 0x45EB908, head 0x45EC880
Tx ring 0x45E9658, shadow 0x45E7E58, Free: head 0x45E7E58, tail 0x45E964C, Used0
MAP Tx ring 0x45EA7E8, shadow 0x45EA668,  Free: head 0x45EA6C8, tail 0x45EA6B0,C
Rx: framing_err 0, hcs_err 0, no_buffer 0, short_pkt 0
    invalid_sid 0, invalide_mac 0, bad_ext_hrd_pdu 0

Cable 3/0 Downstream 0 is up
 Frequency 555.000000 MHz, Channel Width 6 MHz, 256-QAM, Symbol Rate 5.360537 Ms
 FEC ITU-T J.83 Annex B, R/S Interleave I=32, J=4
 Downstream Channel ID: 49
 Spectrum Group:

Cable 3/0 Upstream 0 is up
 Frequency 10.000000 MHz, Channel Width 3.200000 MHz, 16QAM, Symbol Rate 2.5600s
 Spectrum Group:
 SNR 284 dBmv
 Nominal Input Power Level 0 dBmV
 Ranging Backoff (Start 0, End 4)
 Ranging Insection Interval (200 ms)
 Tx Backoff Start 2, Tx Backoff End 8
 Modulation Profile Group 1
 Concatenation is enabled
 part_id=0x3137, rev_id=0x3
 Range Load Reg Size=0x58
 Request Load Reg Size=0xE
 Minislot Size in number of Timebase Ticks is = 4
 Minislot Size in Symbols = 64
 Bandwidth Requests = 0x0
 Piggyback Requests = 0x0
 Invalid BW Requests = 0x0
 Minislots Requested = 0x0
 Minislots Granted = 0x0
 Minislots Size in Bytes = 32, Map Advance (Dynamic): 2479 usecs
 UCD Count = 0
 Slots 854, NoUWCollNoEnergy 4, FECCorHCS 0, HCS 0
```

Figure 3-8 show controllers cable Command Output

- Use the **show controllers ethernet** command to display the following fast Ethernet module information:

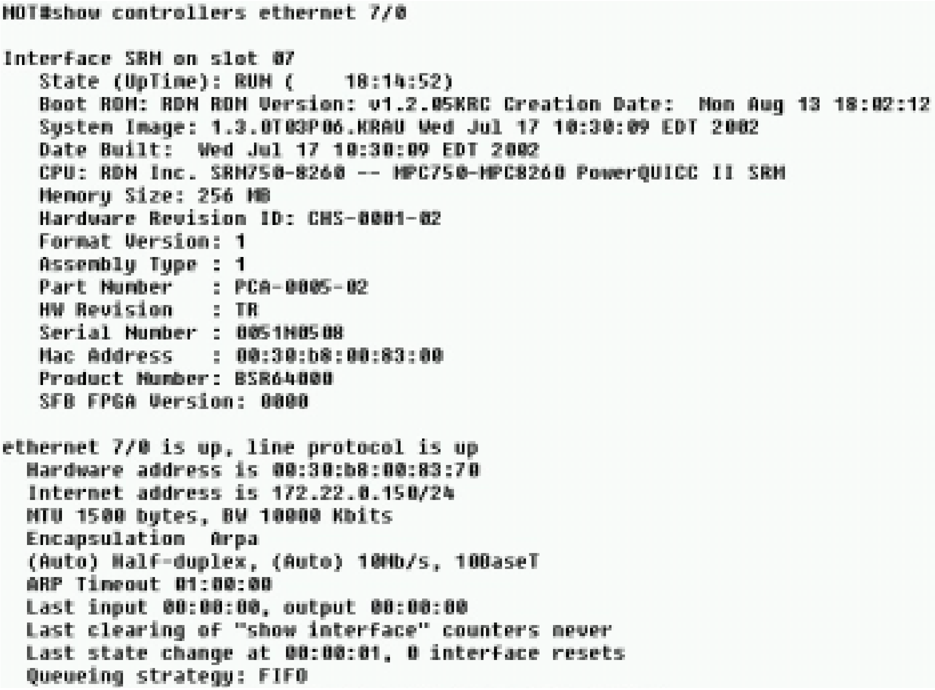
  MOT#**show controllers ethernet** {<*x*>/<*y*>}

  where:

  x is the fast Ethernet module slot number.

y is the Ethernet interface number.

Figure 3-9 displays controller information for the Ethernet interfaces on the SRM module:



```
MOT#show controllers ethernet 7/0

Interface SRM on slot 07
   State (UpTime): RUN (     10:14:52)
   Boot ROM: RDN ROM Version: v1.2.05KRC Creation Date:  Mon Aug 13 18:02:12
   System Image: 1.3.0T03P06.KRAU Wed Jul 17 10:30:09 EDT 2002
   Date Built:  Wed Jul 17 10:30:09 EDT 2002
   CPU: RDN Inc. SRM750-8260 -- MPC750-MPC8260 PowerQUICC II SRM
   Memory Size: 256 MB
   Hardware Revision ID: CHS-0001-02
   Format Version: 1
   Assembly Type : 1
   Part Number    : PCA-0005-02
   HW Revision    : TR
   Serial Number : 0051M0508
   Mac Address    : 00:30:b8:00:83:00
   Product Number: BSR64000
   SFB FPGA Version: 0000

ethernet 7/0 is up, line protocol is up
  Hardware address is 00:30:b8:00:83:70
  Internet address is 172.22.0.150/24
  MTU 1500 bytes, BW 10000 Kbits
  Encapsulation  Arpa
  (Auto) Half-duplex, (Auto) 10Mb/s, 10BaseT
  ARP Timeout 01:00:00
  Last input 00:00:00, output 00:00:00
  Last clearing of "show interface" counters never
  Last state change at 00:00:01, 0 interface resets
  Queueing strategy: FIFO
```

Figure 3-9 show controllers ethernet Command Output

- Use the **show controllers gigaether** command to display the following Gigabit Ethernet module information:

  MOT#**show controllers gigaether** {<*x*>/<*y*>}

  where:

  x is the gigabit Ethernet module slot number.

  y is the gigabit Ethernet interface number.

4

Configuring SNMP

Overview

This chapter describes the commands used to configure SNMP for managing the BSR 64000™ system and monitoring the network using its command line interface. For further information on the CLI commands described in this chapter, refer to the *BSR Command Reference Guide*. Configuring SNMP for the BSR involves the following tasks:

- Configuring SNMP Server Parameters
- Enabling SNMP
- Configuring SNMP Server Identification
- Configuring SNMP Access Levels
- Configuring SNMP Traps
- Monitoring SNMP

Configuring SNMP Server Parameters

To configure SNMP Agent operations, you use a series of **snmp-server** commands. To issue **snmp-server** commands, do the following:

1. Enter Global Configuration mode.

```
MOT#configure
```

2. Type **snmp-server** followed by the available SNMP command and its associated parameters that are listed in Table 4-1:

Table 4-1 snmp-server Commands

Command	Description	Value	Default
snmp-server access	Define SNMP Access Policy information.	Enter the SNMP access group name and information.	Not configured
snmp-server chassis-id	Define chassis ID to uniquely identify this system by writing to the *chassisId* MIB object.	1 to 255 alphabetic-numeric characters.	Not configured
snmp-server community	Set community string and access privilege	Read-Only or Read-Write.	Read-Only
snmp-server community-table	Configures snmpCommunity MIB (RFC 2576)	Octet alpha numeric string that is used as an index into the snmpCommunityMIB table.	Not configured
snmp-server contact	Define system contact information by writing to the *sysContact* MIB object.	1 to 225 alphabetic characters.	Not configured
snmp-server context	Define context information.	Name and OID of context and referenced MIB view.	Not configured
snmp-server convert [password | key]	Convert Authentication or Privacy password or key to a localized key.	Name of the password or key.	Not configured

Table 4-1 snmp-server Commands *(continued)*

Command	Description	Value	Default
snmp-server docs-trap-control	Set SNMP DOCSIS traps.	cmtsBPKMTrap cmtsBpiInitTrap cmtsDCCAckFailTrap cmtsDCCReqFailTrap cmtsDCCRspFailTrap cmtsDynServAckFailTap cmtsDynServReqFailTrap cmtsDynServRspFailTrap cmtsDynamicSATrap cmtsInitRegAckFailTrap cmtsInitRegReqFailTrap cmtsInitRegRspFailTrap	Not configured
snmp-server enable	Enable SNMP traps, informs, or coexistence.	Traps, informs, or coexistence	Not configured
snmp-server engineID	Define SNMP Engine information.	Numeric character string	Not configured
snmp-server group	Define a User Security Model group.	Name of SNMP Group	Not configured
snmp-server host	Define an SNMP host to receive SNMP notification information.	IP address of host machine	Not configured
snmp-server location	Define system location information by writing to the *sysLocation* MIB object.	1 to 225 alphabetic characters.	Not configured

Table 4-1 snmp-server Commands *(continued)*

Command	Description	Value	Default
snmp-server notify {<octet-string>} {<octet-string>} [**inform** | **trap**] [**nonvolitile** | **volitile**] [**active** | **not-in-service** | <*cr*>]	Configure the notification table.	1. Specify the first octet-string, which specifies the notification name. 2. Specify the second octet-string, which specifies the notification tag. 3. Choose whether the notification messages sent to the host is a trap or inform. 4. Choose whether the message is stored in nonvolitile and volitile. 5. Set the RowStatus to active or not-in-service.	RowStatus is active.
snmp-server target-addr {<octet-string>} {<octet-string>} {A.B.C.D} udp-port {<0-65535}	Configuring the SNMP server target address table.	1. Set the first octet-string, which specifies the snmpTargetAddr table name. 2. Configure second octet-string, which specifies the snmpTargetAddrName (index into snmpTargetAddrTable). 3. Set the IP address of the SNMP notification host. 4. Enter the UDP port number.	Not configured

Table 4-1 snmp-server Commands *(continued)*

Command	Description	Value	Default
snmp-server notify-filter {<octet-string>} {<OID>} {<octet-string> [**included** | **excluded**] [**nonvolitile** | **volitile**] [**active** | **not-in-service** | *<cr>*]	Configure snmpNotifyFilter table.	1. Specify the first octet-string, which specifies the snmpNotifyFilter table profile name (index #1). 2. Specify the snmpNotifyFilter subtree (index #2) OID, which defines the family of included and excluded subtrees. 3. Specify the second octet-string, which is the snmpNotifyFilter mask that combines with snmpNotifyFilter subtree to define the family; the default is an empty string. 4. Choose whether subtrees are included or excluded. 5. Choose whether the message is stored in nonvolitile and volitile. 6. Set the RowStatus to active or not-in-service.	Not configured.

Table 4-1 snmp-server Commands *(continued)*

Command	Description	Value	Default
snmp-server notify-filter profile {<OID>} {<octet-string> [**included** | **excluded**] [**nonvolitile** | **volitile**] [**active** | **not-in-service** | *<cr>*]	Configures the snmpNotifyFilter subtree profile.	1. Specify the snmpNotifyFilter subtree (index #2) OID, which defines the family of included and excluded subtrees. 2. Specify the first octet-string, which specifies the snmpNotifyFilter mask table profile name. 3. Choose whether subtrees are included or excluded. 4. Choose whether the message is stored in nonvolitile and volitile. 5. Set the RowStatus to active or not-in-service.	Not configured.
snmp-server packetsize	Define the maximum allowable SNMP packet size. This is the maximum packet size the server can send or receive.	484 to 17940 bytes	1400 bytes
snmp-server port	Define a new SNMP Agent port number.	Port number	161

Table 4-1 snmp-server Commands *(continued)*

Command	Description	Value	Default
snmp-server shutdown	Shuts down the SNMP Agent, thus preventing it from processing incoming SNMP packets, but retains all SNMP configuration data in the event the agent is restarted.	N/A	Disabled
snmp-server sysname	Define system name information by writing to the *sysName* MIB object.	SNMP system name	Not configured
snmp-server target-params	Configure the snmpTarget-params table (rfc2573).	Octet alpha numeric string that is used as an index into the snmpTarget-params table.	Not configured
snmp-server trap	Restrict the rate of SNMP trap messages generated.	0-2147483648/number of seconds	Not configured
snmp-server user	Define a USM user who can access the SNMP Engine.	SNMP User name	Not configured
snmp-server view	Define a particular set of MIB objects that a particular security group can access.	Included/Excluded	Excluded

Enabling SNMP

To configure SNMP, enable the SNMP server using the **snmp-server enable** command, as shown below:

1. Use the **snmp-server enable** command in Global Configuration mode to enable SNMP Server operation with traps, *or* informs, as shown below:

Note: Different SNMP versions can coexist.

```
MOT(config)#snmp-server enable {informs | traps}
```

where:

informs enables SNMP Informs.

traps enables SNMP Traps.

Note: You can also enable the SNMP Agent by issuing any SNMP configuration command.

2. Disable SNMP using the **snmp-server shutdown** command, as shown below:

```
MOT(config)#snmp-server shutdown
```

This command disables the SNMP Agent running on the server.

Configuring SNMPv3

This section describes the procedure to configure the SNMP Agent to receive, process, and respond to SNMPv3 packets. The procedure includes configuring the following:

- Local SNMP name (Engine-ID) (only if not already configured)
- SNMP User
- SNMP Group
- SNMP Access Policy
- SNMP View
- SNMP Context (only if not already configured)

Use the CLI to configure an SNMP User, an SNMP Group to associate the user to an access policy, and an SNMP Access Policy. If they are not already configured, configure the local SNMP Engine-ID of the agent and an appropriate SNMP Context.

1. A default local Engine-ID is configured to the MAC address of the SRM. To determine if a local Engine-ID is configured and to ensure that the agent is running, use the **show snmp engineID** command in Privileged EXEC mode, as shown below:

```
MOT#show snmp engineID
```

Output similar to the following is displayed:

```
Local SNMP engine-ID: 0030b8008300000000000000
```

Note: The local SNMP Engine-ID is configured as b80042005959000000000000. If the agent is not running, you can enable it by entering any SNMP command in Global Configuration mode, such as defining a community string using the **snmp-server community public** command.

2. When the agent is running, and if no local Engine-ID is configured or if you want to change the previously configured value, enter the **snmp-server engineID local** command to specify the desired Engine-ID, as shown below:

```
MOT(config)#snmp-server engineID local <WORD>
```

where:

WORD is the desired engine name.

Note: If you have configured SNMP Users and you change the local SNMP Engine-ID, you must also update the users to use them or delete them and set new ones.

3. The user name is the same user name you specify in a remote network management station (a MIB browser, for example) when you wish to contact the agent via SNMPv3. Issue the **snmp-server user** command in Global Configuration mode to configure an SNMP User with the appropriate security attributes, as shown below:

   `MOT(config)#`**snmp-server user** {<*WORD*>} [**auth** [**sha** | **md5**] [**key** <*string*> | **local** <*string*>} | **password** <*string*> | **string** [**eng-id** <*WORD*> | **priv des56** <*string*> [**eng-id** <*WORD*> |] | **eng-id** <*WORD*> [**public** | <*cr*>]]

Note: If the local engineID changes and existing users are used with the new engineID, you must update the users with the new engineID. This feature guards the SNMP Agent against tampering.

where:

WORD is the name of the user.

auth indicates the user authentication parameter.

sha indicates the HMAC-SHA algorithm for authentication.

md5 indicates the HMAC-MD5 algorithm for authentication.

key <*string*> is the standard key.

local <*string*> is the localized authentication key for user.

password <*string*> is the assigned password; valid size is up to 64 characters.

string sets authentication password string for user.

priv des56 <*string*> sets privacy authentication parameters for users.

eng-id <*WORD*> specify the engine ID octet string associated with this user.

WORD specifies the engine-id with this user; local value of engine ID.

public sets the usmUserPublic SNMP object.

4. Assign an SNMP User to an SNMP Group by first configuring an SNMP Group with the user defined in the Group and then assign the Group to an Access Policy. Configuring an SNMP Access Policy assigns SNMP Groups to particular views into the MIB tree and further identifies them with an SNMP version. Use the **snmp-server group** command in Global Configuration mode to configure a new SNMP Group or a table that maps SNMP Users to SNMP Groups, as shown below. Use the **no snmp-server group** command to remove a specified SNMP Group.

   `MOT(config)#`**snmp-server group** <*WORD*> <*WORD*> **v3**

   where:

   WORD is the desired name of the SNMP Group.

   WORD is the name of the SNMP User (security name) assigned to the group.

5. Assign an SNMP Access Policy to an SNMP Group to set the MIB objects that the SNMP Group can access. This relates to the set of MIB objects that an SNMP User can access.

   `MOT(config)#`**snmp-server access** {<*group-name*>} [**v1** | **v2c** | **v3**] [**notify** <*mib-view-name*> | **prefix** {<*mib-view-name*>} [**notify** | **read** | **write**] | <*cr*>] | **read** {<*mib-view-name*>} [**notify <***notify-view name***>** | **write <***write-view name***>**] <*cr*> | **write** {<*mib-view-name*>} [**notify <***notify-view name***> |** <*cr*>]

   where:

   group-name is the name of the SNMP Group.

   mib-view-name is the name of the MIB view for the type of process.

   notify specifies a notify view for this access group.

   read specifies a read view for this access group.

   write specifies a write view for this access group.

6. Configure an SNMP View specifying the desired set of MIB objects. The view is either read, write, or read and write, depending on the SNMP Access Policy configured. Configure an SNMP View entry using the **snmp-server view** command in Global Configuration mode, as shown below.

   ```
   MOT(config)#snmp-server view <name> <group> {included | excluded}
   ```

   where:

   name is the desired name of the view.

   group is the group name or the object identifier (OID) value.

7. By default, the proper SNMP Context is automatically configured when the SNMP Access Policy is configured, and you need not configure a context. However, if you change parameters for the access policy, you may want to manually configure a context. To verify the proper SNMP Context is set, use the **show snmp context** command, as shown below:

   ```
   MOT#show snmp context
   ```

   Output similar to the following is displayed:

   ```
   context #1:
   context #2: public
   ```

Note: Since no prefix is specified in the configuration of the SNMP Access Policy, a prefix, or context, of blank (that is, "") is associated with this access policy and is automatically configured. This is seen as *context #1* in the output above.

Note: If you specify a prefix of *public*, for example, while configuring the SNMP Access Policy, a prefix *public* is associated with this access policy and a context *public* is automatically added also. This is seen as *context #2* in the output above.

8. If you do not see the proper context, you can set an SNMP Context by issuing the **snmp-server context** command, as shown below:

   ```
   MOT(config)#snmp-server context <name>
   ```

   where:

name is the SNMP Context name.

9. To view the configuration of SNMPv3 entries, use the **show running config** command in Privileged EXEC mode, as shown below:

   MOT#**show running config**

Example

The example below uses the commands described above to specify *sha_user* as the user, *sha* as the authentication algorithm, and *motorola* as the password. It assigns the user to an SNMP Group *auth_g* and specifies that this user and group combination applies for SNMPv3 only. It then configures an SNMP Access Policy for the SNMP Group *auth_g* to have read and write privileges according to the SNMP View *auth_view*. The example then configures the SNMP View *auth_view* and gives access to the *mib-2* MIB group and all its MIB objects.

```
MOT(config)#snmp-server user sha_user auth sha motorola
MOT(config)#snmp-server group auth_g sha_user v3
MOT(config)#snmp-server group auth_g v3 auth read auth_view
write auth_view
MOT(config)#snmp-server view auth_view mib-2 included
MOT#show running config
```

The final command results in a display with SNMPv3 information similar to the following:

```
snmp-server engineid local 123456700000000000000000
snmp-server context
snmp-server user sha_user auth sha local
6efff7e12db360a1b0f97ce84501c6d9aff2d282 eng-id
123456700000000000000000
snmp-server group auth_g sha_user v3
snmp-server access auth_g v3 auth match exact read auth_view
write auth_view
snmp-server community public ro
snmp-server view auth_view mib-2 included
```

Configuring SNMP Server Identification

Configuring the following parameters provides unique network identification for the SNMP Agent:

- Contact person
- System location
- Engine identifier

Configuring System Contact Information

Establish a system contact string using the **snmp-server contact** command in Global Configuration mode, as shown below:

```
MOT(config)#snmp-server contact <text>
```

where:

text is the system contact name.

Configuring System Location Information

Set the system location string using the **snmp-server location** command in Global Configuration mode, as shown below:

```
MOT(config)#snmp-server location <text>
```

where:

text is the location of the system on the network.

Configuring the EngineID

1. Use the **snmp-server engineID** command in Global Configuration mode to configure the Engine-ID for the local or remote SNMP entity, as shown below. An SNMP entity can be an agent or management station.

```
MOT(config)#snmp-server engineID {local <engine-id> | remote
<ip-address> [udp-port <port-num>] <engine-id>}
```

Note: For specifying a local engineID, you need not specify the entire 24-character engineID if it contains trailing zeros. You can specify only the portion of the engineID up to the trailing zeros.

Note: Upon shipment, the agent has a default engineID that is equal to the chassis MAC address.

where:

engine-id is the local or remote SNMP Engine engineID.

ip-address is the remote SNMP Engine IP address.

port-num is the optional UDP port number.

2. To remove an Engine-ID for the local or remote SNMP entity, use the **no snmp-server engineID** command as shown below:

```
MOT(config)#snmp-server engineID {local <engine-id> | remote
<ip-address> [udp-port <port-num>] <engine-id>}
```

where:

engine-id is the local or remote SNMP Server engineID.

ip-address is the remote SNMP Engine IP address.

Note: A local SNMP Engine-ID *must* be configured to use SNMPv3.

Configuring SNMP Access Levels

Access to an SNMP Server by an SNMP client is determined by a specified access level. You can set access levels using the following methods:

- The community name method of access control and View-based Access Control Method (VACM) are used with SNMPv1 and SNMPv2. A community name is a text string used to authenticate messages between a management station and an SNMP client.
- The User-based Security Model (USM) and VACM are used with SNMPv3. USM establishes user names and passwords and provides encryption. VACM determines whether to permit access from a management station to a managed object on the local SNMP client.

Figure 4-1 provides an overview of the SNMP access level configuration process and the SNMP version (v1, v2c, or v3) that supports each **snmp-server** command.

Defining a Community Name

The community name access method, used predominantly with SNMPv1 and SNMPv2, uses an SNMP Community Table that identifies those communities that have read-only, read-write, or administrative permission to the SNMP MIB stored on a particular server. You must define at least one SNMP community string. The community string acts like a password to permit access to the SNMP Agent. Optionally, you can specify one or more of the following characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent.
- A MIB view that defines the subset of all MIB objects accessible to the given community. Refer to *Configuring a MIB View*, later in this chapter.
- Read-write or read-only permission for the MIB objects accessible to the community.

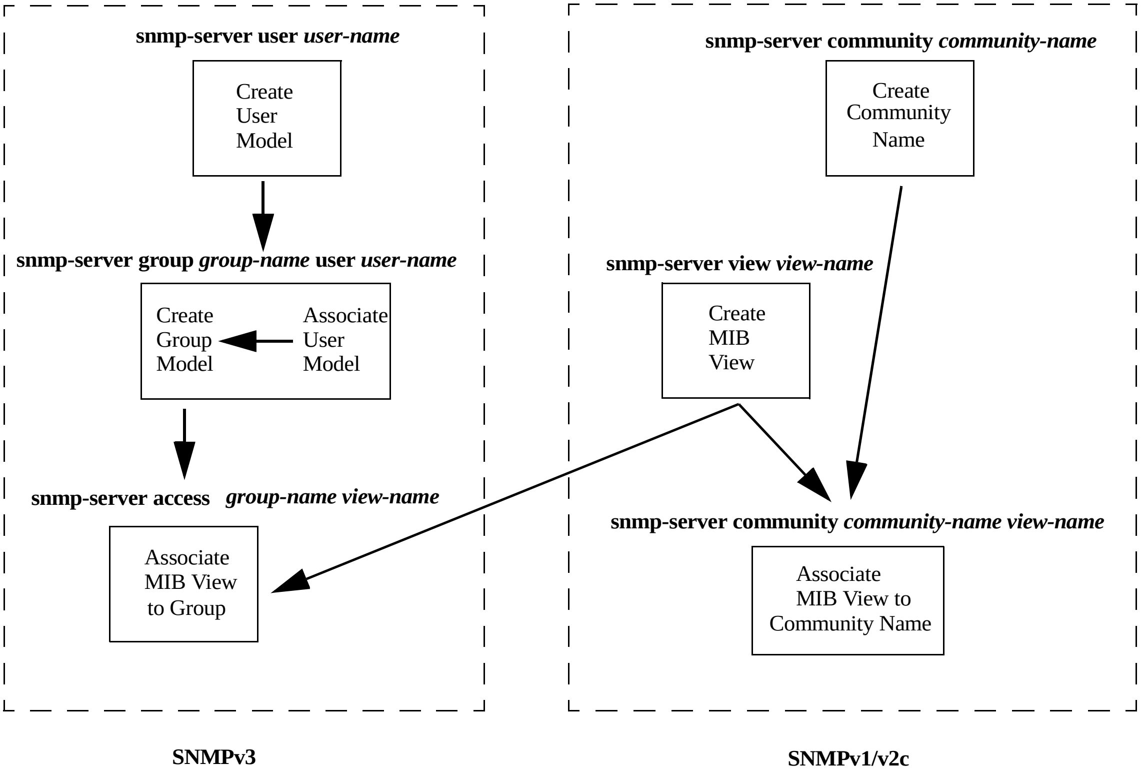

snmp-server user user-name
Create User Model
snmp-server group group-name user user-name
Create Group Model
Associate User Model
snmp-server access group-name view-name
Associate MIB View to Group
snmp-server community community-name
Create Community Name
snmp-server view view-name
Create MIB View
snmp-server community community-name view-name
Associate MIB View to Community Name
SNMPv3
SNMPv1/v2c

Figure 4-1 SNMP Access Level Configuration Process

1. Use the **snmp-server community** command in Global Configuration mode to define a community access string to permit access using SNMPv1 and SNMPv2 to the SNMP Agent as shown below:

   `MOT(config)#`**snmp-server community** <*community-name*> [**ro** | **rw**] [**view** <*view-name*>] [<*number*>]

   where:

   community-name is the name of the SNMP community.

   view-name is the name of the view.

   number is the number of the access list.

2. Use the **no snmp-server community** command to remove the specified community string.

Caution: Using only a community name to establish SNMP access levels is not a completely secure access control method. The community string is included in every packet transmitted between an SNMP client and server but is not encrypted, which makes SNMP Get/Set operations potentially accessible to any packet capture software. Access to SNMP Get/Set operations could provide the following:

- A blueprint of a network topology and configuration
- Control of a device configured for remote SNMP management

Note: If you do not specify a view, the system sets a default to the *dod* MIB group (that is, 1.3.6). If you do not specify an administrative permission (read-only or read-write), the system uses the default of read-only.

Example

```
MOT(config)#snmp-server community public
```

Configuring USM and VACM Security

You can define security levels for each SNMP Server by establishing a USM with defined access levels. Set permissions to a specified set of MIB objects with the VACM. These security methods encrypt transmissions between an SNMP client and server and allow the SNMP Server to authenticate each user requesting access. They also let you specify various protection levels (unsecured, authenticated, and authenticated with encryption) that are common to SNMPv3. USM specifies authentication and encryption functions. VACM specifies how access-control rules are handled. Configuring USM and VACM security for an SNMP Server involves the following tasks:

- Configuring a Group Model
- Configuring a MIB View

- Associating Groups to MIB Views
- Configuring an SNMP Context

1. To configure a new SNMP User, use the **snmp-server user** command in Global Configuration mode, as shown below.

   `MOT(config)#`**snmp-server user** <*username*> [**auth** {**sha** | **md5**} {**password** <*password*> | **key** <key> | **local** <*localized_key*>} [priv des56 {**password** <*password*> | **key** <key> | **local** <*localized_key*>}] [**eng-id** <*engine-id*>]]

   where:

   username is the new SNMP User.

   password is the assigned password; valid size is up to 64 characters.

   localized_key is the localized key.

   engine-id is the engine name.

2. Use the **no snmp-server user** <*user-name*> command to remove a user.

   where:

   username is the SNMP User to be removed.

Configuring a Group Model

1. Use the **snmp-server group** command in Global Configuration mode to configure a new SNMP Group or a table that maps SNMP Users to SNMP Groups, as shown below.

   `MOT(config)#`**snmp-server group** <*group-name*> <*user-name*> {**v1** | **v2c** | **v3**}

   where:

   group-name is the new SNMP Group name.

   username is the SNMP User.

2. Use the **no snmp-server group** command to remove a specified SNMP Group, as shown below:

   `MOT(config)#`**no snmp-server group** <*group-name*>

   where:

group-name is the removed SNMP Group name.

Configuring a MIB View

You can assign MIB views to SNMP Groups or community strings to limit the MIB objects that an SNMP manager can access. You can use a predefined view or create your own view. You create or update an SNMP View entry using the **snmp-server view** command in Global Configuration mode, as shown below. You can enter this command multiple times for the same view record. Later lines take precedence when an object identifier is included in two or more lines.

1. `MOT(config)#`**snmp-server view** <*name*> <*oidsubtree*> {**included** | **excluded**}

   where:

   name is the new MIB view name.

   oidsubtree is the subtree of the MIB view family name.

   included specifies the subtree is included in view.

   excluded specifies the subtree is excluded from view**.**

2. Use the **snmp-server view** <*new-view-name*> to view the available MIB groups when configuring the view. The following example creates a view that includes all objects in the MIB-II subtree:

   snmp-server view <*name*> **mib-2 included**

   where:

   name is the new MIB view name.

3. Use the **no snmp-server view** <*new-view-name*> command to remove the specified SNMP MIB view entry.

Examples

The following example creates a view that includes all objects in the MIB-II system group and all objects in the RiverDelta Networks, Inc. Enterprise MIBs:

```
snmp-server view <new-view-name> system included
snmp-server view <new-view-name> riverdelta included
```

The following example creates a view that includes all objects in the MIB-II group except for the Interfaces group:

```
snmp-server view <new-view-name> mib-2 included
snmp-server view <new-view-name> interfaces excluded
```

Associating Groups to MIB Views

You can associate an SNMP Group to specific SNMP MIB views. This restricts access to the MIB objects defined in the view to the SNMP Group, limiting which MIB objects an SNMP manager can access.

Use the **snmp-server access** command in Global Configuration mode to map SNMP Groups to SNMP MIB views, as shown below.

`MOT(config)#`**snmp-server access** <*group-name*> {**v1** | **v2c** | **v3 noauth** | **v3 auth** | **v3 priv**} [**prefix** <>] [**match exact** | **match prefix**] [**read** <*new-MIB-view-name*>] [**write** <*new-MIB-view-name*>] [**notify** <*new-MIB-view-name*>]

Example

```
MOT(config)#snmp-server access <group-name> v3 auth write
<new-MIB-view-name>
```

Configuring an SNMP Context

An SNMP Context is a collection of managed object resources that an SNMPv2 entity can access. Configuring a context record as part of an access policy further restricts access to MIB views. The object resources identified by a context are either local or remote. An SNMP Context that refers to local object resources is identified as a MIB view. The SNMP entity uses local mechanisms to access the management information identified by the context.

1. To create or update a context record, use the **snmp-server context** command in Global Configuration mode, as shown below.

   `MOT(config)#`**snmp-server context** <c*ontext-name*>

2. Use the **no snmp-server context** command to remove the specified SNMP Context.

Note: You must configure a context in conjunction with configuring an access policy. If a prefix name is configured with the **snmp-server access** command, it should have the same name as the context.

Note: If you do not specify a prefix name with the **snmp-server access** command, you must add a *blank* context such as *snmp-server context.*

Example

The following example shows how to create a context to be used to further restrict access:

```
MOT(config)#snmp-server context mycontext
```

Configuring Packet Size

1. Use the **snmp-server packetsize** command in Global Configuration mode to change the permitted SNMP packet size that the SNMP Server can receive or transmit, as shown below:

   MOT(config)#**snmp-server packetsize** <*value*>

   where:

   > *value* is the permitted SNMP packet size, expressed in bytes; valid range is 484 to 17940; default is 1400.

2. Use the **show snmp packetsize** to view the currently configured packet size.

Configuring SNMP Traps

SNMP traps are generated according to standard and enterprise MIB specifications. Traps are sent to IP hosts configured in a proprietary Trap Host Table that the SNMP Agent maintains. This section provides information for:

- Enabling trap generation
- Configuring trap destinations
- Restricting trap rates

Enabling Traps

1. Use the **snmp-server enable traps** command in Global Configuration mode to enable SNMP traps, as shown below. This command configures the BSR to send SNMP traps. If you do not specify the trap type, all trap types are enabled.

   ```
   MOT(config)#snmp-server enable traps [<trap-type>]
   ```

   where:

   trap-type is the trap type; valid type entries are bgp, ospf, snmp, or vrrp.

2. Use the **no snmp-server enable traps** command to disable SNMP notifications.

Note: For a host to receive a trap, you must configure an **snmp-server host** command for that host and globally enable the trap through the **snmp-server enable traps** command.

Configuring a Trap Destination

Traps are sent to IP hosts configured in a proprietary Trap Host Table that the SNMP Agent maintains. Each entry in the table contains:

- The IP address of the trap destination
- The community or user name to send in the trap message
- The host destination configured to receive specific trap types
- The SNMP format (v1, v2c, or v3) of the trap PDU to use for that destination

Specifying the Destination IP Address

1. Use the **snmp-server host** command in Global Configuration mode to specify a destination machine to receive SNMP trap information, as shown below. This command is disabled by default, specifying that no notifications are sent. If you enter the **snmp-server host** command with no keywords, all trap types are sent to the host.

   `MOT(config)#`**snmp-server host** <*host-address*> [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] <*community-string*> [**udp-port** <*port*>] [<*notification-type*>]

   where:

   host-address is the destination machine to receive SNMP trap information.

   traps indicates enable SNMP traps.

   informs indicates enable SNMP informs.

   version indicates the version of notifications.

   1 is the SNMP version 1 message processing model.

   2c is the SNMP version 2c message processing model.

   3 auth is the SNMP version 3 message processing model with authentication and uses unscrambled packets.

   3 noauth is the SNMP version 3 message processing model with no authentication and uses unscrambled packets.

   3 priv is the SNMP version 3 message processing model that authenticates and scrambles packets.

   community-string is the password; valid entry is from 1 to 32 alphabetic characters.

   port is the UDP port.

   notification-type is the type of traps sent; valid entries are snmp, ospf, vrrp, and bgp.

2. Use the **no snmp-server host** command to remove the specified host.

Note: If the *community-string* is not defined using **snmp-server community** command prior to using the **snmp-server host** command, the default form of the **snmp-server community** command is automatically inserted into the configuration. The password (*community-string*) used for this default configuration is the same as that specified in the **snmp-server host** command.

Specifying Specific Trap Types

Use the **snmp-server host traps** command in Global Configuration mode to configure an SNMP trap host to receive specific trap types as shown below. If no trap type is specified, all traps are sent to this trap host.

```
MOT(config)#snmp-server host <ip-address> traps <community-string>
[udp-port <port>] [snmp | ospf | vrrp | bgp]
```

where:

ip-address is the IP address of the host

community-string is the password; valid entry is from 1 to 32 alphabetic characters.

port is the UDP port number; valid values are 0 to 65535.

Specifying SNMP Trap Versions

Using the **snmp-server host version** command in Global Configuration mode to configure the SNMP trap version for the specified trap type as shown in the example below. If no trap type is specified, all traps are sent to this trap host.

```
MOT(config)#snmp-server host <ip-address> version {1 | 2c | 3 auth | 3 noauth
| 3 priv} <community-string> [udp-port <port>] [snmp | ospf | vrrp | bgp]
```

where:

ip-address is the IP address of the host.

community-string is the password; valid entry is from 1 to 32 alphabetic characters.

port is the UDP port number; valid values are 0 to 65535.

If no **version** keyword is present, the default is version 1. The **no snmp-server enable traps** command entered with no keywords disables traps, but not informs, to the host. To disable informs, use the **no snmp-server enable informs** command.

Restricting Trap Rates

You can restrict the rate of SNMP traps generated to prevent excessive traffic on the network.

1. Use the **snmp-server trap rate-limit** command in Global Configuration mode to specify the number of traps allowed per number of seconds, as shown below. When the number of traps exceeds the limit, trap generation is automatically disabled unless the **auto-restart** parameter is enabled.

   `MOT(config)#`**snmp-server trap rate-limit** [<*number-of-traps 0-2147483648*> <*per-number-of-seconds*> | **auto-restart**]

   where:

   number-of-traps is the number of SNMP traps allowed value; valid entries are from 0 to 2147483648.

   number is the number of seconds during which the number of traps value is allowed.

2. Use the **no snmp-server rate-limit** command to disable rate limitations.

Example

The following example sets a limit of 100 traps per second.

```
MOT(config)#snmp-server trap rate-limit 100 1
```

Monitoring SNMP

To monitor the status of SNMP operations on your network and check current SNMP settings, you use a series of **show snmp** commands. To issue **show snmp** commands, do the following:

1. Enter the Privileged EXEC or Global Configuration mode.
2. Use the **show snmp** command to check the status of SNMP communications and access counter information for SNMP operations. Use the **show snmp** command with the *command-name* option to access specific SNMP information.

Table 4-2 lists the **show snmp** commands.

Table 4-2 show snmp Commands

Command	Description
show snmp	Provides counter information for SNMP operations. It also displays the chassis ID string.
show snmp access	Displays SNMP Access information.
show snmp community	Displays SNMP community information.
show snmp contact	Displays SNMP system contact information.
show snmp context	Displays SNMP v3 context information.
show snmp engineID	Displays local and remote engineIDs.
show snmp group	Display the names of the SNMP Groups, security names, security models, status of the different views, and storage type for each group.
show snmp host	Displays SNMP host notification information.
show snmp location	Displays SNMP system location information.
show snmp packetsize	Displays the currently configured SNMP PDU packet size.
show snmp port number	Displays SNMP Agent port information.
show snmp sysname	Displays SNMP sysname system information.
show snmp users	Displays information for SNMP User names in the SNMP Group user name table.
show snmp view	Displays SNMP View information including subtree, status, storage type, and security.

5

Configuring Interfaces and TCP/IP Features

Overview

This chapter describes how to configure the various interfaces on the BSR and the Transmission Control Protocol/Internet Protocol (TCP/IP) features for the BSR 64000™ system.

This chapter discusses the following topics:

- About TCP/IP Level Features
- Setting IP Interface Addresses
- Configuring the Address Resolution Protocol
- Configuring Broadcast Addressing
- Configuring the MTU
- Configuring Static Routes
- Clearing Route Table Entries
- Configuring the Internet Control Message Protocol
- Configuring Tunnels on an Interface
- Configuring an Unnumbered Interface
- Configuring the Internet Control Message Protocol
- Tracing a Route
- Managing the Router
- Gathering TCP/IP Related Information

About TCP/IP Level Features

IP provides basic packet delivery service for all TCP/IP networks. The connection-oriented TCP exchanges control information with a remote device to verify that the device is ready to receive data before it is sent. However, IP uses other protocols to establish the connection and to supply error detection and recovery such as ICMP.

A datagram is a packet format defined by IP. An IP packet contains the necessary destination address information. A packet-switching network uses the addressing information to switch the packet from one physical network to another, moving it toward its final destination. Each packet travels the network independent of any other packet.

IP performs the following functions:

- Moves data between the Network Access layer and the Host-to-Host Transport layer
- Routes datagrams to remote hosts
- Fragments and reassembles datagrams

A router forwards traffic from one network to another. The router also transmits route information to other routers. This route information is stored in routing tables that enable a router without a direct physical connection to a packet's destination to forward the packet to a router that is closer to its destination. The process continues at each router until the packet reaches a router attached to the same network as the destination host. That router delivers the packet to the specified host on its local network, and the packet reaches its final destination.

Setting IP Interface Addresses

You must configure the interfaces on the BSR in order for the BSR to transmit and receive data and communicate with other network devices.

Table 5-1 describes each BSR interface and corresponding module:

Table 5-1 BSR Interface and Module Descriptions

CLI Interface	BSR Module	Interface Description
ethernet	Supervisory Resource Module (SRM) located in slot 7.	Ethernet interface 0 is a 10 Mbps management interface that does not support the negotiation feature and is associated with its corresponding port on the SRM I/O module. Ethernet interface 1 and 2 are typically used to support an external T1/E1 BITS clock and are associated with their corresponding ports on the SRM I/O module.
	10/100 Ethernet Module	Provides 8 10/100 Mbps Ethernet ports.
cable	DOCSIS 1:4 Resource Module	CMTS that provides 1 downstream channel and 4 upstream channels.
gigaether	Gigabit Ethernet Resource Module	Provides one 1000 Mbps optical Ethernet interface.
pos	OC3/OC12 POS Module	Provides two high speed OC3/OC12 SONET interfaces.
loopback	N/A	Loopback interfaces are used to act as inbound logical interfaces when physical interfaces go down. Up to 16 loopback interfaces can be configured on the BSR. Refer to "Configuring a Loopback Interface" on page 5-8 for more information.
tunnel	N/A	A tunnel interface is a logical interface used to make point-to-point links between two devices. Refer to "Configuring Tunnels on an Interface" on page 5-10 for more information.
unnumbered interface	N/A	Used in point-to-point connections when an IP address is not required. This interface is only available on the POS module. Refer to "Configuring an Unnumbered Interface" on page 5-13 for more information.

Follow these steps to assign an IP address and subnetwork mask to an interface on a module:

1. To identify where the module is in the chassis, use the **show chassis status** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show chassis status
   ```

2. Determine the slot number of the module.

3. Use the **configure** command in Privileged EXEC mode to enter Global Configuration mode, as shown below:

   ```
   MOT#configure
   ```

4. To enter an interface, use the **interface** command in Global Configuration mode, as shown in the following example:

   ```
   MOT(config)#interface {pos | ethernet | gigaether | cable} <x>/<y>
   ```

   where:

   pos is the Packet over SONET interface.

   ethernet is any 10 or 10/100 Ethernet interface.

   gigaether is the Gigabit Ethernet interface

   cable is any DOCSIS interface.

   x is the desired module slot on the BSR.

   y is the interface number on the module.

Note: There is only one cable interface per module. The cable interface is always **0**.

5. Use the **ip address** command to set a primary IP address and subnetwork mask for an interface, as shown below:

   ```
   MOT(config-if)#ip address <A.B.C.D> <net-mask>
   ```

   where:

   A.B.C.D is the IP address of the interface.

net-mask is the network mask of the IP network, on which the interface is associated.

For example:

```
MOT(config-if)#ip address 10.10.10.135 255.255.255.0
```

6. To optionally configure a secondary IP address for an interface use the **ip address secondary** command, in Interface Configuration mode, as shown below:

Note: A secondary IP address can be used in some implementations as the loopback interface. There are special options for configuring a secondary IP address on the cable interface. Refer to "Subneting DHCP Clients on the Cable Interface" on page 6-40 for more information.

```
MOT(config-if)#ip address <A.B.C.D> <net-mask> secondary
```

where:

A.B.C.D is the IP address of the BSR interface designated for the loopback interface.

net-mask is the subnetwork mask of the IP network, on which the interface is associated.

secondary optionally designates the IP address as a secondary ip address.To specify additional secondary IP addresses include the keyword **secondary** after the IP address and subnet mask.

For example:

In the sample below, 198.108.1.127 is the primary address and 172.45.7.17 is a secondary address for Ethernet 0/0.

```
interface ethernet 0/0
ip address 198.108.1.127 255.255.255.0
ip address 172.45.7.17 255.255.255.0 secondary
```

7. To verify that the information was entered correctly, use the **show running-config** command in Interface Configuration mode as shown below:

```
MOT(config-if)#show running-config
```

The running configuration displays. If the information is *not* correct, repeat this procedure.

Removing an IP Address

You can disable IP processing on a particular interface by removing its IP address with the **no ip address** command. If the router detects another host using one of its IP addresses, it will print an error message on the console. The software supports multiple IP addresses per interface. A specific IP address can be removed from an interface or all IP addresses associated with the interface can be removed.

- Use the **no ip address** command in Interface Configuration mode to remove a specific IP address from the interface, as shown below:

  `MOT(config-if)#`**no ip address** <*A.B.C.D*> <*net-mask*> [<*cr*> | <**secondary**>]

- Use the **no ip address** command in Interface Configuration mode to remove all addresses from the interface, as shown below:

  `MOT(config-if)#`**no ip address**

Configuring Auto-Negotiation on the 10/100 Ethernet Module

The Ethernet interface on the 10/100 Ethernet module can be configured for the Ethernet port speed (10 or 100), duplex mode (full or half), and to enable/disable auto-negotiation:

speed {**10**|**100**|**auto**}

duplex {**half**|**full**|**auto**}

The 10/100 Mbps Ethernet interface on the 10/100 Ethernet module is set to auto-negotiate the speed and duplex mode by default.

Follow these steps to manually set the speed and negotiation parameters for the Ethernet port on the 10/100 Ethernet module:

1. To enter the Ethernet interface on the 10/100 Ethernet module, use the **interface ethernet** command in Global Configuration mode, as shown in the following example:

   `MOT(config)#`**interface ethernet** <*slot*>/<*interface*>

where:

slot is the 10/100 Ethernet module slot.

interface is the Ethernet interface number.

2. To manually set the duplex mode for full-duplex so that the Ethernet interface can send and receive signals at the same time, use the **duplex full** command in Interface Configuration mode, as shown in the following example:

   ```
   MOT(config-if)#duplex full
   ```

   - or -

   To manually set the duplex mode for half-duplex so that the Ethernet interface can either send or receive signals, but cannot do both at the same time, use the **duplex half** command in Interface Configuration mode, as shown in the following example:

   ```
   MOT(config-if)#duplex half
   ```

   To return to the default, which is to auto-negotiate the duplex mode, use the **duplex auto** command in Interface Configuration mode, as shown in the following example:

   ```
   MOT(config-if)#duplex auto
   ```

3. To manually set the speed of the Ethernet interface to 100 Mbps, use the **speed 100** command in Interface Configuration mode, as shown in the following example:

   ```
   MOT(config-if)#speed 100
   ```

   - or -

   To manually set the speed of the Ethernet interface to 10 Mbps, use the **speed 10** command in Interface Configuration mode, as shown in the following example:

   ```
   MOT(config-if)#speed 10
   ```

   To return to the default, which is to auto-negotiate the speed of the Ethernet interface, use the **speed auto** command in Interface Configuration mode, as shown in the following example:

   ```
   MOT(config-if)#speed auto
   ```

4. To verify the speed and duplex mode for the Ethernet interface, use the **show interface ethernet** command in any mode, as shown in the following example:

   ```
   MOT(config-if)#show interface ethernet <slot>/<interface>
   ```

   where:

slot is the 10/100 Ethernet module slot.

interface is the Ethernet interface number.

5. To verify that the information was entered correctly, use the **show running-config** command in Privileged EXEC mode as shown below:

   ```
   MOT#show running-config
   ```

   The running configuration displays. If the information is *not* correct, repeat this procedure.

Configuring a Loopback Interface

Logical interfaces called a loopback interfaces can be used to act as inbound logical interfaces when physical interfaces go down. These logical interfaces are always active and they allow the routing process associated with physical interfaces to stay active. IP Packets routed to loopback interfaces are rerouted to the appropriate BSR routing process. IP packets not destined to loopback interfaces are dropped by the loopback interfaces.

Loopback interfaces are used for the following reasons:

- Collect accurate service-related information through an SNMP manager about active or down interfaces on the BSR.
- Indirectly access an outbound physical interface that cannot be directly accessed.
- When the designated router election process occurs in OSPF, the designated router choice can be forced by assigning a higher IP address for the loopback address.

Up to 16 loopback interfaces can be configured on the BSR. Follow these steps to define a loopback address:

1. Use the **interface loopback** command in Global Interface mode, to define a loopback interface, as shown below:

   ```
   MOT(config)#interface loopback <n>
   ```

   where:

   n is the number of the loopback interface from 1 to 16

2. Use the **ip address** command in Interface Configuration mode to define an IP address for the loopback interface, as shown below:

`MOT(config-if)#`**ip address** <*A.B.C.D*> <*net-mask*>

where:

A.B.C.D is the IP address of the BSR interface designated for the loopback interface.

net-mask is the subnetwork mask of the IP network, on which the interface is associated.

3. To optionally configure a secondary IP address for the loopback interface use the **ip address secondary** command, in Interface Configuration mode, as shown below:

`MOT(config-if)#`**ip address** <*A.B.C.D*> <*net-mask*> **secondary**

where:

A.B.C.D is the secondary IP address of the BSR interface.

net-mask is the subnetwork mask of the IP network, on which the interface is associated.

secondary optionally designates the IP address as a secondary ip address.To specify additional secondary IP addresses include the keyword **secondary** after the IP address and subnet mask.

4. To verify the loopback interface configuration, use the **show interface loopback** command in Interface configuration mode, as shown below:

`MOT(config-if)#`**show interface loopback** <*n*>

where:

n is the number of the loopback interface from 1 to 16

Configuring Tunnels on an Interface

A tunnel interface is a logical interface that is used to encapsulate various packet types and send them over a created a point-to-point link between two devices at remote points over an IP internetwork. Multi-protocol packets are encapsulated using either IP, GRE, or DVMRP tunnel encapsulation to traverse the link.

Tunneling is used for the following reasons:

- Allows multiprotocol LANs to connect over a single-protocol backbone.
- Solves problems for routed networks with restricted hop counts.
- Connects disjointed subnetworks.
- Permits virtual private networks (VPNs) across the internet.

Up to 255 tunnel interfaces can be configured on the BSR. A separate tunnel for each link must be configured, since it is a point-to-point link.

When configuring tunnels on an interface, you must specify the tunnel source and tunnel destination. You can optionally enable an ID key for a tunnel interface. You can use Tunnel ID keys as a form of weak security to prevent misconfiguration or injection of packets from a foreign source. Set the key to the same value on the tunnel endpoints. The tunnel ID key is available with generic router encapsulation (GRE) only.

Note: When using GRE, the ID key is carried in each packet. We recommend that you do not rely on this key for security purposes.

Up to 255 tunnel interfaces can be configured on the BSR. Follow these steps to define a tunnel address:

1. Use the **interface tunnel** command in Global Interface mode, to define a tunnel interface, as shown below:

   ```
   MOT(config)#interface tunnel <n>
   ```

   where:

   n is the number of the tunnel interface from 0 to 255

2. Use the **ip address** command in Interface Configuration mode to define an IP address for the tunnel interface, as shown below:

   ```
   MOT(config-if)#ip address <A.B.C.D> <net-mask>
   ```

   where:

   A.B.C.D is the IP address of the tunnel interface.

   net-mask is the subnetwork mask of the tunnel interface.

3. To optionally configure a secondary IP address for the tunnel interface use the **ip address secondary** command, in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip address <A.B.C.D> <net-mask> secondary
   ```

   where:

   A.B.C.D is the IP address of the tunnel interface.

   net-mask is the subnetwork mask of the tunnel interface.

   secondary optionally designates the IP address as a secondary ip address.To specify additional secondary IP addresses include the keyword **secondary** after the IP address and subnet mask.

4. To specify the tunnel source, use the **tunnel source** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#tunnel source <address> <type> <number>
   ```

   where:

   address is the IP address of the tunnel interface source.

   type is the interface type.

   number is the tunnel interface number.

5. To specify the tunnel destination, use the **tunnel destination** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#tunnel destination {<address> | <hostname>}
   ```

   where:

   address is the IP address of the tunnel interface destination.

   hostname is the DNS name of the destination.

6. To delete a tunnel source, use the **no tunnel source** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**no tunnel source**

7. To delete a tunnel destination, use the **no tunnel destination** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**no tunnel destination**

8. To set the encapsulation mode when sending packets over a tunnel, use the **tunnel mode** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**tunnel mode** {**ipip** | **gre** | **dvmrp**}

   where:

   ipip indicates IP in IP encapsulation; the default is IP in IP.

   gre indicates GRE.

   dvmrp indicates Distance Vector Multicast Routing Protocol (DVMRP).

9. To disable the encapsulation mode for a tunnel interface, use the **no tunnel mode** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**no tunnel mode** {**ipip** | **gre** | **dvmrp**}

   where:

   ipip indicates IP in IP encapsulation.

   gre indicates generic routing encapsulation (GRE).

   dvmrp indicates distance vector multicast routing protocol (DVMRP).

10. To specify a security key for GRE tunneling, use the **tunnel key** command in Interface-tunnel Configuration mode as shown below:

    MOT(config-if)#**tunnel key** <*number*>

    where:

    number is the key number; valid entries are 0 to 4294967295.

11. To delete a GRE tunnel security key for GRE tunneling, use the **no tunnel key** command in Interface-tunnel Configuration mode as shown below:

    MOT(config-if)#**no tunnel key**

12. To verify the tunnel interface, use the **show interface tunnel** command in Interface configuration mode, as shown below:

    ```
    MOT(config-if)#show interface tunnel <n>
    ```

    where:

    n is the tunnel number; valid entries are 0 to 255.

Configuring an Unnumbered Interface

An unnumbered interface is used in point-to-point connections when an IP address is not required. This enables IP processing on an interface without assigning an explicit IP address to the interface. You supply the interface location, which is the type and number of another interface on which the router has an assigned IP address, and this interface cannot be another unnumbered interface.

Note: An unnumbered interface can only be configured on the POS module.

Follow these steps to set an unnumbered interface on the POS module:

1. Before configuring the unnumbered interface, a loopback interface must be configured. Refer to “Configuring a Loopback Interface” on page 5-8 for more information.
2. Use the **end** command to go back to Global Configuration mode.
3. Use the **interface pos** command in Global Configuration mode to enter the POS interface, as shown below:

    ```
    MOT(config)#interface pos <slot>/<interface>
    ```

    where:

    slot is the POS interface slot number.

    interface is the POS interface number.

4. To enable an interface for data processing without an explicit IP address, use the **ip unnumbered** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**ip unnumbered** {**loopback** <*n*> | **pos** <*slot*>/<*interface*> | **serial** <*slot*>/<*interface*> | **ethernet** <*slot*>/<*interface*> **gigaether** | <*slot*>/<*interface*>}

   where:

   loopback is the loopback interface on the POS module.

   n is the loopback interface from 1 to 16.

   pos is POS interface the on the POS module.

   serial is the Serial interface on the POS module.

   ethernet is the loopback interface on the 10/100 Ethernet module or Ethernet management interface or serial interfaces on the SRM.

   gigaether is the loopback interface on the Gigabit Ethernet module.

   slot identifies the module slot number.

   interface identifies the BSR interface number.

Configuring the Address Resolution Protocol

Since no relationship exists between an Ethernet (MAC) address and an Internet address, a router uses the Address Resolution Protocol (ARP) to send a packet across the network to a host with a known Internet address. A host that uses ARP maintains a cache of Internet-to-Ethernet address mappings. To keep the cache from growing too large, dated entries are removed. Before transmitting a packet, the host checks its cache for the Internet-to-Ethernet address mapping. If the mapping is not found, the host sends an ARP request.

To add a permanent ARP entry for an interface on the BSR, use the steps in this section:

1. To add an entry to the ARP cache, use the **arp** command in Global Configuration mode, as shown in the example below:

   MOT(config)#**arp** <*A.B.C.D*> <*mac-address*> <*type*> [**alias**]

   where:

A.B.C.D is the IP address of the ARP entry, specified in dotted-decimal notation.

mac-address is a 48-bit hardware address of the ARP entry.

type is the encapsulation type.

alias specifies that the software respond to ARP as if it owns the specified address, if proxy arp is enabled.

2. To set the ARP cache timeout for a specific interface, use the **arp timeout** command in Interface Configuration mode, as shown below:

   ```
   MOT(config)#arp timeout <number>
   ```

   where:

   number is ARP cache timeout value, expressed in minutes; valid entries are 1 to 6000; default is 60.

   Use the **no arp timeout** command to restore the default.

Reverse ARP

Reverse ARP, defined in RFC 903, works like ARP, except that the RARP request packet requests an Internet address instead of a hardware address. The BSR acts as an RARP server. To enable RARP, use the **ip rarp-server** command in Interface Configuration mode, as shown below. Set the IP address to one of the interface addresses.

```
MOT(config-if)#ip rarp-server <A.B.C.D>
```

where:

A.B.C.D is the source protocol IP address in replies.

Address Resolution Using Proxy ARP

The router uses proxy ARP, as defined in RFC 1027, to help hosts with no knowledge of routing determine the hardware addresses of hosts on the same or other networks or subnets. Under proxy ARP, if the router receives an ARP request for a host that is not on the same network as the ARP request sender, and if the router has the best route to that host, the router sends an ARP reply packet giving its own local data link address. In addition, if a host on the local network is incapable of responding to an ARP request, the router responds on its behalf when Proxy ARP is enabled and host IP-to-MAC address mapping is stored in the router with a static **arp** command invoked with the **alias** option. The host that sent the ARP request then sends its packets to the router, which forwards them to the intended host.

1. Proxy ARP is not enabled by default. To enable Proxy ARP, use the **ip proxy-arp** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip proxy-arp
   ```

2. To disable Proxy ARP, use the **no ip proxy-arp** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#no ip proxy-arp
   ```

3. To verify ARP status, use the **show running-config** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show running-config
   ```

Configuring Broadcast Addressing

A broadcast is a data packet destined for all hosts on a particular physical network. Network hosts recognize broadcasts by special addresses. The BSR system supports the following broadcast packet types:

- Limited Broadcast - A packet is sent to a specific network or series of networks.
- Flooded Broadcast - A packet is sent to every network.
- Directed Broadcast - A packet is sent to a specific destination address where only the host portion of the IP address is either all ones or all zeros.

To avoid broadcast storms, use a single broadcast address scheme on a network and set the address to be used as the broadcast address. The BSR can accept and interpret all possible forms of broadcast addresses.

Defining Broadcast Address

You can use several IP commands to perform broadcast tasks.

1. To define a broadcast address for an interface, use the **ip broadcast-address** command in Interface Configuration mode. You specify an IP address to set the broadcast address, as shown in the example below:

   ```
   MOT(config-if)#ip broadcast-address <A.B.C.D>
   ```

   where:

   A.B.C.D is the interface IP address.

   Use the **no ip broadcast-address** command to restore the default IP broadcast address for an interface.

2. To enable broadcasting of all directed broadcasts to all addresses in the host portion of an IP address, use the **ip directed-broadcast** command in Interface Configuration mode, as shown in the example below:

   ```
   MOT(config-if)#ip directed-broadcast
   ```

   Use the **no ip directed-broadcast** command to disable broadcasting to all addresses.

Configuring the MTU

Fragmentation occurs when an IP datagram is too large for a network maximum transmission unit (MTU) size, and the large datagram is divided into several smaller pieces for transmission. Lower layer protocols may also set the MTU. If the MTU that is set in lower layers differs from the MTU that is set at the IP layer, the BSR uses the lower value.

To set the MTU for packets on an interface, use the **ip mtu** command in Interface Configuration mode. On some interfaces, such as cable, you cannot set the MTU.

```
MOT(config-if)#ip mtu <size>
```

where:

size is the MTU size, expressed in bytes; valid entries are 68 to the maximum MTU of the physical interface; default is 1496.

Use the **no ip mtu** command to restore the default MTU size.

Note: The MTU depends on the type of physical interface.

Configuring Static Routes

You can arrange for a router to receive and send traffic by a specific static route, and you can set a default route to reduce the routing table size. If a path to a destination network cannot be located by a router, the BSR forwards the traffic to the default router, if one is defined. Static routes cause packets moving between a source and a destination to take a specific path. Static routes are important when the software cannot build a route to a particular destination and for specifying a gateway to which all unroutable packets are sent. To configure a static route, use the following command in Global Configuration mode:

1. To set a specific route through a network, use the **ip route** command in Global Configuration mode, as shown below.

   ```
   MOT(config)#ip route <A.B.C.D> <mask> {<forward-ip-address> | null
   <num:0,0> | pos <x>/<y> | tunnel <n>} [<distance>] [tag <1-4294967295>]
   [range <n>]
   ```

   where:

   A.B.C.D is the static route destination IP address.

   mask is the static route destination IP address mask.

   forward-ip-address is the Forwarding router’s IP address.

   null is null interface and port; valid entries are 0 and 0.

   pos specifies the POS interface.

x is the POS module slot number.

y is the POS interface number.

tunnel <*n*> is a tunnel interface number from 1 to 255.

distance is the administrative distance; default is 1.

tag specifies the match to control route-map redistribution.

1-4294967295 is the match value.

range <*n*> indicates an established static route; valid entries are 1 to 65536.

2. To set a default route, use the address 0.0.0.0 with the **ip route** command in global Configuration mode, as shown in the example below:

   ```
   MOT(config)#ip route 0.0.0.0 0.0.0.0 198.56.0.2
   ```

   Use the **no ip route** command to remove a static route from the routing table.

Table 5-2 describes the dynamic routing protocols and their default distances.

Table 5-2 Route Sources and Administrative Distances

Route Source	Default Distance
Enhanced IGRP external route	170
Enhanced IGRP summary route	5
External BGP	20
IGRP	100
IGRP external route	170
Internal BGP	200
Internal Enhanced IGRP	90
OSPF	110
RIP	120
Static route	1
Unknown	255

Clearing Route Table Entries

To clear dynamic entries from the routing table, use the **clear ip route** command in Privileged EXEC mode, as shown below. You must specify the IP address of the routes and the mask of the IP address. Use the asterisk (*) to clear all dynamic routes.

`MOT(config-if)#`**clear ip route** {***** | <*A.B.C.D*> | <*A.B.C.D*> <*mask*>}

where:

> * is the asterisk character that clears all routes in the routing table.
>
> *A.B.C.D* is the IP address of the route.
>
> *mask* is the subnet mask of the IP address.

Configuring the Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) allows a router or destination host to report errors in data traffic processing to the original packet source. ICMP messages occur when errors take place in processing an unfragmented data packet or in the first fragment of a fragmented data packet. ICMP message delivery is not guaranteed. The Router Discovery Protocol, enabled via ICMP, informs hosts of the existence of routers by tracing router discovery packets.

Follow these sections to configure ICMP on the BSR:

- About IRDP
- Enabling IRDP
- Enabling ICMP

About IRDP

The router software provides router discovery, by which the router can dynamically learn about routes to other networks using the ICMP Router Discovery Protocol (IRDP) for detecting routers. IRDP uses router advertisement and router solicitation messages to discover addresses of routers on directly attached subnets.

With IRDP, each router periodically multicasts or broadcasts router advertisement messages from each of its interfaces. Hosts discover the addresses of routers on the directly attached subnet by listening for these messages. Hosts can use router solicitation messages to request immediate advertisements, rather than wait for unsolicited messages.

IRDP offers several advantages over other methods of discovering addresses of neighboring routers. Primarily, it does not require hosts to recognize routing protocols, nor does it require manual configuration by an administrator.

Router advertisement messages allow hosts to discover the existence of neighboring routers, but not which router is best to reach a particular destination. If a host uses a poor first-hop router to reach a particular destination, it receives a redirect message identifying a better choice.

Enabling IRDP

Use the following procedure to configure IRDP.

1. To enable IRDP, use the **ip irdp** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**ip irdp**

2. To configure a proxy IP address to advertise messages from an interface, use the **ip irdp address** command in Interface Configuration mode, as shown below

   MOT(config-if)#**ip irdp address** <*A.B.C.D*> [**address** <*A.B.C.D*> | **holdtime** <*number1*> | **maxadvertinterval** <*number2*> | **minadvertinterval** <*number3*> | **multicast** | **preference** <*number4*>]

   where:

   > **address** is the IP addresses to proxy-advertise, preference value.
   >
   > *A.B.C.D* is the IP address of the interface on which the messages are advertised.
   >
   > **holdtime** is the amount of time that advertisements are valid, expressed in seconds.
   >
   > *number1* is the holdtime, expressed in seconds; valid entries are 1 to 9000; default is 1800.

maxadvertinterval is the maximum time between advertisements.

number2 is the maximum interval between advertisements, expressed in seconds; valid entries are 4 to 1800; default is 600.

minadvertinterval is the minimal time between advertisements, expressed in seconds.

number3 is the minimum interval between advertisements, expressed in seconds; valid entries are 3 to 1800; default is 450.

multicast indicates advertisements are sent with multicast.

preference indicates preference value for this interface.

number4 indicates preference; valid entries are -2147483648 to 2147483647; default is 0.

Use the **no ip irdp** command to disable the function.

3. To change the duration of IRDP advertisement ages, use the **ip irdp holdtime** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip irdp holdtime <number>
   ```

   where:

   number is the IRDP advertisement ages value, expressed in seconds; valid entries are 1 to 9000; default is 1800.

4. To change the maximum time between IRDP advertisements, use the **ip irdp maxadvertinterval** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip irdp maxadvertinterval <number>
   ```

   where:

   number is the maximum time between IRDP advertisements, expressed in seconds; valid entries are 4 to 1800; default is 600.

5. To change the minimum time between IRDP advertisements, use the **ip irdp minadvertinterval** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip irdp minadvertinterval <number>
   ```

   where:

number is the minimum time between IRDP advertisements, expressed in seconds; valid entries are 3 to 1800; default is 450.

6. To send IRDP advertisements with Multicast packets, use the **ip irdp multicast** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip irdp multicast
   ```

   The default is IRDP broadcast.

7. To set the IRDP routing preference level, use the **ip irdp preference** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip irdp preference <number>
   ```

   where:

   number is the IRDP routing preference level; valid entries are -2147483648 to 2147483647; default is 0.

Enabling ICMP

Once IRDP is enabled, follow the steps in this section to enable ICMP.

1. Use the **ip mask-reply** command to enable ICMP netmask reply.

   ```
   MOT(config-if)#ip mask-reply
   ```

   Use the **no ip mask-reply** command to disable the function.

2. Use the **ip redirects** command to enable the sending of IP redirect messages.

   ```
   MOT(config-if)#ip redirects
   ```

   Use the **no ip redirects** command to disable the sending of redirect messages.

3. Use the **ip unreachables** command to enable the generation of ICMP unreachable messages, as shown in the example below:

   ```
   MOT(config-if)#ip unreachables
   ```

   Use the **no ip unreachables** command to disable the function.

4. You can send ICMP echo request packets to a specified address. You can set an optional packet count for a destination. To do this, use the **ping** command from Privileged EXEC mode, as shown below:

MOT#**ping** [<*hostname*> | <*A.B.C.D*>] [**size** <*number1*>] [<*number2*>] [**timeout** <*number3*>] [**source** <*A.B.C.D*>] [**tos** <*number4*>] [**ttl** <*number5*>] [**df**]

where:

hostname is the DNS host name.

A.B.C.D is an IP address.

number1 is the packet size value, expressed in bytes; valid entries are 40 to 65515.

number2 is the packet number or request messages sent, between 1 to 65535.

timeout is the duration.

number3 is the timeout value, expressed in seconds; valid entries are 1 to 1024.

source is the IP address of the source.

tos specifies the type of service.

number4 is a value between 0 to 255.

ttl is the time to live.

number5 is the TTL value; valid entries are 0 to 255.

df sets the *don't fragment* flag in the IP header.

In the following example, a request packet is sent to address 192.35.42.1, with a size of 55, a packet count of 10, and a timeout value of 10 seconds.

MOT#**ping 192.35.42.1 size 55 10 timeout 10**

Tracing a Route

A route path includes all IP level devices, such as routers and servers, that packets travel through over the network on a hop-by-hop bases to get to their intended destination.

To to identify the route path from the route source to the route destination, use the **traceroute** command in Privileged EXEC mode, as shown in the following example:

MOT#**traceroute** [<*A.B.C.D*> | <*hostname*>]

where:

A.B.C.D is the source IP address.

hostname is the Domain Name Server (DNS) hostname.

Managing the Router

Follow these sections to manage routing operations on the BSR:

- Enabling IP Source Routing
- IP Accounting
- Clearing Interface Counters
- Clearing Interface Counters
- Clearing IP Routes
- Clearing the ARP Cache
- Clearing DNS Entries
- Clearing IP Traffic

Enabling IP Source Routing

The BSR examines IP header options on every packet. It supports the IP header options Strict Source Route, Loose Source Route, Record Route, and Time Stamp, which are defined in RFC 791. If the software finds a packet with one of these options enabled, it performs the appropriate action. If it finds a packet with an invalid option, it sends an ICMP Parameter Problem message to the source of the packet and discards the packet.

IP provides a provision that allows the source IP host to specify a route through the IP network. This provision is known as source routing. Source routing is specified as an option in the IP header. If source routing is specified, the software forwards the packet according to the specified source route. This feature is employed when you want to force a packet to take a certain route through the network. The default is to disable source routing.

You can enable IP source-route header options if they have been disabled by using the following command in Global Configuration mode:

```
MOT(config)#ip source-route
```

IP Accounting

IP accounting provides basic IP accounting functions. By enabling IP accounting, users can see the number of bytes and packets switched through the software on a source and destination IP address basis. Only transit IP traffic is measured and only on an outbound basis; traffic generated by the software or terminating in the software is not included in the accounting statistics. To maintain accurate accounting totals, the software maintains two accounting databases: an active and a checkpoint database.

IP accounting also provides information identifying IP traffic that fails IP access lists. Identifying IP source addresses that violate IP access lists alerts you to possible attempts to breach security. The data also indicates that you should verify IP access list configurations. By default, IP accounting displays the number of packets that have passed access lists and were routed.

1. To enable IP accounting, use the following command for each interface in Interface Configuration mode:

   ```
   MOT(config-if)#ip accounting
   ```

2. To configure other IP accounting functions, use one or more of the following commands in Global Configuration mode. Set the maximum number of accounting entries to be created.

   MOT(config)#**ip accounting-threshold** <*num*>

   where:

   > *num* is the maximum number of accounting entries; valid entries are from 0 to 10000.

3. To filter accounting information for hosts, use the **ip accounting-list command** in Global Configuration mode, as shown below:

   MOT(config)#**ip accounting-list** <*prefix*> <*mask*>

   where:

   prefix is the host IP address.

   mask is the wildcard mask.

4. To control the number of transit records stored in the IP accounting database, use the **ip accounting-transits** command, as shown below:

   MOT(config)#**ip accounting-transits** <*num:0.10000*>

   where:

   num is the maximum number of transit entries

5. To select hosts for which IP accounting information is kept, use the **ip accounting-list** command, as shown below:

   MOT(config)#**ip accounting-list** <*prefix*> <*wildcard-mask*>

   where:

   prefix is the IP address of host

   wildcard-mask is the wildcard mask bits

6. To display the active accounting database, use the EXEC command **show ip accounting**, as shown below:

   MOT#**show ip accounting** [**access-violations** | **checkpoint** | **output-packets**]

   where:

   access-violations is access violations in accounting database.

   checkpoint is the checkpoint IP accounting database.

   output-packets is the output packets in accounting database.

7. To display the checkpoint database, use the **show ip accounting checkpoint** EXEC command.

   MOT#**show ip accounting checkpoint**

8. To clear the active database and create the checkpoint database, use the **clear ip accounting** command in Privileged Exec mode, as shown below:

   MOT#**clear ip accounting**

9. To clear the checkpoint database, use the **clear ip accounting checkpoint** command in Privileged Exec mode, as shown below:

   MOT#**clear ip accounting checkpoint**

Clearing Interface Counters

To clear a specific or all interface counters, use the **clear counters** command in Privileged EXEC mode, as shown below.

`MOT#`**clear counters** [**ethernet** <*slot*>/<*interface*> **cable** <*slot*>/<*interface*> | **loopback** <*loop number*> | **pos** <*slot*>/<*interface*> | **gigaether** <*slot*>/<*interface*> | **tunnel** <*tunnel number*>]

where:

cable clears the cable interface counters.

ethernet clears the Ethernet interface counters.

loopback clears the loopback interface counters.

loop number is the loopback interface number from 1 to 16.

tunnel clears the tunnel interface counters.

tunnel number is the tunnel interface number from 0 to 255.

pos clears the Packet over SONET (POS) interface counters.

gigaether clears the Gigabit Ethernet interface counters.

interface identifies the port number.

slot identifies the module slot number.

Clearing IP Routes

To clear one or more IP routes from the IP routing table, use the **clear ip route** command in Privileged EXEC mode, as shown below:

`MOT#`**clear ip route** {***** | <*A.B.C.D*> | <*A.B.C.D*> <*mask*>}

where:

***** is the asterisk character is entered to delete all routes.

A.B.C.D is the network or subnetwork address.

mask is the associated IP address of the removed routes.

Clearing the ARP Cache

To clear all dynamic entries from the ARP cache, use the **clear arp-cache** command in Privileged EXEC mode, as shown below:

MOT#**clear arp-cache**

Clearing IP Traffic

To reset the IP traffic statistics counters to zero, use the **clear ip traffic** command in Privileged EXEC mode, as shown below:

MOT#**clear ip traffic**

Clearing DNS Entries

Use the **clear host** command in Privileged EXEC mode to delete DNS host entries from the host-name-and-address cache, as shown below:

MOT#**clear host** {<*hostname*> | *****}

where:

hostname deletes a specific DNS host entry.

* deletes all DNS host entries.

Gathering TCP/IP Related Information

You can monitor IP using the **show ip** commands discussed in this section.

1. Use the **show ip arp** command to display ARP table information.

   MOT#**show ip arp** [<*A.B.C.D*>] [<*hostname*>] [<*mac-address*>] [ethernet <*slot*> {/} <*port*>]

   where:

   A.B.C.D is the IP address.

   hostname is the host name.

mac-address is the MAC address.

slot is the Ethernet interface slot number.

port is the Ethernet interface port number.

2. Use the **show ip interface** command to display the current state of all IP interfaces or a specific interface, as shown below. The default is all interface types and all interfaces.

   ```
   MOT#show ip interface ethernet 7/0 [brief]
   ```

   where:

   brief displays summary information.

3. Use the **show ip route** command to display the routing table status. You can specify an optional IP mask that filters specific routes. You can enter this command from any mode.

   ```
   MOT#show ip route [<hostname> | bgp | connected | ospf | rip | static |
   <A.B.C.D> [mask]]
   ```

4. Use the **show ip route static** command to display the status of static routes in the routing table. You can specify an optional IP mask that filters specific routes.

   ```
   MOT#show ip route static [<hostname> | bgp | connected | ospf | rip | static |
   <A.B.C.D> [mask]]
   ```

5. Use the **show ip traffic** command from Privileged EXEC mode to display statistics about IP traffic, which includes DHCP lease query statistics, as shown below:

   ```
   MOT#show ip traffic
   ```

6

Configuring the CMTS

Overview

The following sections contain the tasks used to configure and manage your cable network:

- Initial Cable Interface Configuration Tasks
- Configuring a Downstream Channel
- Configuring an Upstream Channel
- Bundling Cable Interfaces into a Single IP Subnet
- Subneting DHCP Clients on the Cable Interface
- Creating a Modulation Profile
- Setting Network Parameters for Cable Modems
- Configuring Baseline Privacy
- Setting QoS Parameters
- Implementing Spectrum Management
- Using Flap Lists
- Managing Multicast Maps
- Pinging a Cable Modem at the MAC Layer
- Resetting the Cable Modem
- Clearing Cable Interface Counters
- Gathering DOCSIS Network Information

Initial Cable Interface Configuration Tasks

Ensure that an IP address and subnetwork mask is configured for the cable interface before performing the following tasks. Refer to Chapter 5 for more information.

This section discusses the initial configuration tasks that must be performed to make the cable interface on the DOCSIS module operational.

You must perform the following basic tasks to configure the cable interface:

- Setting the IP DHCP Relay Functions
- Configuring the Cable Helper and IP Helper Addresses

- Enabling Host Authorization for All CMs
- Creating a Static Host Authorization Entry for a Specific CM
- Enabling Host Authorization for an IP Range of CPEs
- Using DHCP Lease Query Function to Secure Cable Network
- Setting ARP Parameters

Setting the IP DHCP Relay Functions

The IP DHCP relay function is used to forward DHCP messages between clients and servers. The IP DHCP relay agent function gathers broadcast DHCP discovery packets from a Multimedia Terminal Adapter (MTA) device, cable modem (CM), or Customer Premises Equipment (CPE), and forwards the packets to their corresponding DHCP server. The DHCP relay function enables an MTA, CM, or CPE to obtain an IP address from a DHCP server through the DHCP relay agent, which is the router (SRM) between the cable interface and the DHCP server.

The **ip dhcp relay information** command enables the BSR's DHCP relay agent to insert the Spectrum Group Name that DHCP client belongs to and/or inserts the MAC address of the DHCP client and the DOCSIS Device Class Identifier into outbound requests to the DHCP server. The DOCSIS Device Class Indenter is only supported for 1.1 CMs. Support for DHCP Option 82, sub-option 2 (Agent Remote ID) and sub-option 4 (DOCSIS Device Class Identifier) is enabled by the **ip dhcp relay information option** command. Support for DHCP Option 82, sub-option 85 (Spectrum Group Name) is enabled by the **ip dhcp relay information spectrum-group-name.** The **no ip dhcp relay agent information** command disables the insertion of DHCP Option 82, sub-options.

The following steps outline the IP DHCP relay process:

1. An MTA device, CM, or CPE sends broadcast DHCP discover packets to the DHCP relay agent containing a request for an IP address.
2. The DHCP relay agent inserts the MTA, CM, or CPE option into the DHCP discover packets. This option contains either the spectrum group name and associated MAC address or a MAC address.
3. The DHCP relay agent inserts any configured options into the DHCP discover packets. This can be a spectrum group name, a MAC address and a DOCSIS Device Class Identifier or both the spectrum group name and the MAC address and DOCSIS Device Class Identifier.

4. The DHCP server assigns an IP address to each MTA, CM, or CPE that requested an IP address by placing the IP address in the (Your IP Address) yiaddr field in the DHCP packet header. The yiaddr is the IP address to be used by the MTA, CM, or CPE.

5. The DHCP relay agent removes the MTA, CM, or CPE option and forwards the DHCP server reply, containing the IP address to the MTA, CM, or CPE.

Follow these steps to configure the DHCP relay option on the BSR:

1. Use the **show running-config** command in Privileged EXEC mode determine the DHCP relay function is enabled for the desired cable interface, as shown below:

```
MOT#show running-config
```

2. If you need to change or enable the DHCP relay function for a cable interface, enter the desired cable interface from Global Configuration mode.

3. Use the **ip dhcp relay information option** command in Interface Configuration mode to enable the DHCP relay agent to insert a MAC address (Agent Remote ID) only into a client's DHCP packet, as shown below:

Note: The **ip dhcp relay information option** command must be entered to enable the DHCP relay information option function. If the DHCP relay information option function is not enabled, CMs cannot register and go on-line.

```
MOT(config-if)#ip dhcp information option
```

4. Optionally use the **ip dhcp relay information spectrum-group-name** command in Interface Configuration mode to enable the DHCP relay agent to insert the spectrum group name (Circuit ID) into all of the DHCP packets, as shown below:

Note: If a DHCP client on a particular subnet is using an upstream frequency that is not configured as a member of a spectrum group, the spectrum group name is not inserted by the DHCP relay agent into the DHCP discover packet.

```
MOT(config-if)#ip dhcp information spectrum-group-name
```

Configuring the Cable Helper and IP Helper Addresses

The cable helper IP address function disassembles a CM DHCP broadcast packet, and reassembles it into a unicast packet so that the packet can traverse the router and communicate with the DHCP server.

The cable helper address function is used in conjunction with the DHCP relay function. If the **ip dhcp relay information option** command is not set, all requests are sent to the IP address defined by the **ip helper-address** command. When **ip dhcp relay information option** is enabled, the BSR can distinguish between requests from CMs, secondary hosts and secondary MTAs, and forwards the DHCP requests to the cable helper IP address specifically defined for the requesting device.

Note: Multiple cable-helper addresses can be configured for CMs, hosts, and MTAs. If you want both CM and host DHCP requests to be sent to the same DHCP server, configure the same cable helper IP address for hosts and CMs.

Follow the steps in this section to configure the cable helper and IP helper address:

1. Issue the **cable helper-address cable-modem** command in Interface Configuration mode to configure the helper IP address for the cable interface to forward only DHCP broadcasts, as shown below:

   ```
   MOT(config-if)#cable helper-address <A.B.C.D> cable-modem
   ```

   where:

   A.B.C.D is the IP address of the destination DHCP server.

2. Optionally use the **cable helper-address host** command in Interface Configuration mode to configure a secondary helper IP address for the CPE to forward only UDP broadcasts, as shown below:

   ```
   MOT(config-if)#cable helper-address <A.B.C.D> host
   ```

   where:

   A.B.C.D is the IP address of the destination DHCP server.

3. Optionally use the **cable helper-address mta** command in Interface Configuration mode to configure a secondary helper IP address for the Multimedia Terminal Adapter (MTA) device to forward only UDP broadcasts, as shown below:

```
MOT(config-if)#cable helper-address <A.B.C.D> mta
```

where:

A.B.C.D is the destination DHCP server IP address.

4. The IP helper address necessary for the BSR to forward packets to the DHCP server. Issue the the **ip helper-address** command in Interface Configuration mode to forward default UDP broadcasts including IP configuration requests to the DHCP server, as shown below:

Note: The IP helper address must be entered for the DHCP Lease Query function to work regardless of whether the relay agent option is used.

```
MOT(config-if)#ip helper-address <A.B.C.D>
```

where:

A.B.C.D is the destination DHCP server IP address.

5. Use the **show ip dhcp stats** command in Interface Configuration mode to display information about DHCP upstream and downstream port statistics, as shown below:

```
MOT(config-if)#show ip dhcp stats [<0-15> | <cr>]
```

where:

0-15 is the module slot number.

cr is a command return.

Enabling Host Authorization for All CMs

The host authorization feature is used for security purposes on the cable network. When enabled, host authorization denies access to any hacker who tries to take or “spoof” an IP address from any legitimate user on the same cable network. A hacker takes the IP address from this user to steal their data service. The hacker accomplishes this by changing the IP address on their PC to the IP address that the DHCP server assigned to a legitimate user’s CPE.

Note: The host authorization feature is turned off by default.

Follow these steps to enable the host authorization feature:

1. Enter the cable interface on which host authorization is enabled.
2. Use the **host authorization on** command in Interface Configuration mode to enforce the bind of the CM and CPE MAC addresses to the IP address assigned to them (statically or through DHCP), as shown below:

   ```
   MOT(config-if)#host authorization on
   ```

3. To view all entries in the ARP authorization table, use the **show host authorization** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#show host authorization
   ```

Disabling Host Authorization for All CMs

Use the **no host authorization on** command to disable host authorization on the cable interface.

Creating a Static Host Authorization Entry for a Specific CM

Cable operators can create static entries to deny hackers from stealing service from users. Through static entries, cable operators can manually bind the CPE MAC (hardware) and IP address to a particular cable modem. This command may be used in circumstances when DHCP is not used to assign the CPE IP addresses.

Follow these steps to configure and verify a static host authorization:

1. Enter Privileged EXEC mode.

2. Issue the **host authorization cpe** command in Privileged EXEC mode to create a static entry for a specific CM and CPE in Privileged EXEC mode, as shown below:

   MOT#**host authorization** <*mac*> **cpe** <*mac*> <*prefix*>

   where:

   > *mac* is the MAC address of the cable modem.
   >
   > *mac* is the MAC address of the customer premises equipment (CPE).
   >
   > *prefix* is the IP address of the CPE.

3. Use the **show host authorization cpe** command in Privileged EXEC mode to display the static entries and DHCP lease query information for CPEs only, as shown below:

   MOT#**show host authorization cpe**

Deleting a Static Host Authorization Entry for a Specific CM

Use the **no host authorization on cpe** command to delete a host authorization entry in Privileged EXEC mode, as shown below:

MOT#**no host authorization** <*mac*> **cpe** <*mac*> <*prefix*>

where:

> *mac* is the MAC address of the cable modem.
>
> *mac* is the MAC address of the customer premises equipment (CPE).
>
> *prefix* is the IP address of the CPE.

Enabling Host Authorization for an IP Range of CPEs

Instead of adding individual static CPEs on a specific cable interface using the **cable host authorization cpe** command, CPEs can be added automatically to the network by specifying a start and end range of IP addresses. This function allows you to specify a partial subnet by allowing CPEs on different cable interfaces to automatically connect to a network.

Use the **cable host authorization range** command in Global Configuration mode to define a range of CPE IP addresses that are allowed to be added to the host authorization table (static IP table), as shown below. Any CPE IP address within the specified start and end IP address range is added to the host authorization table when a CPE joins the network.

Note: Up to 32 CPE IP address ranges can be defined for the BSR.

```
MOT(config)#cable host authorization range {<prefix> <prefix>}
```

where:

prefix is the start of the IP address range.

prefix is the end of the IP address range.

For example:

```
MOT(config-if)#cable host authorization range 150.42.19.100 150.42.19.109
```

Removing Host Authorization for an IP Range of CPEs

Use the **no cable host authorization range** command to remove the start and end IP range of CPE addresses so that new CPEs trying to join the network (having an IP address within the specified range) cannot be added to the host authorization table, as shown below:

```
MOT(config)#no cable host authorization range {<prefix> <prefix>}
```

A CPE with an IP address that is within the start and end range defined by the **host authorization range** command that is currently connected remains in the host authorization table until it is individually removed by the **no host authorization** command, or if the BSR is reset.

Using DHCP Lease Query Function to Secure Cable Network

The DHCP lease query feature provides additional security on the cable network by preventing hackers from stealing service from customers. Hackers steal service from other subscribers by spoofing their connection information contained in ARP broadcasts. Preventing hackers from spoofing the cable network also prevents undesirable ARP broadcasts from disrupting service on the cable network.

The DHCP Lease Query feature is used in conjunction with the host authorization feature on the BSR to query the location of a hacker’s Cable Modem (CM) and its connected Customer Premises Equipment (CPE) when a packet either arrives from or is destined to a subscriber’s CM and its CPE, and has no location information in the DHCP Lease table.

If the DHCP Lease Query attempt fails, packets associated with the CM and its CPE are discarded. The BSR sends DHCPLEASEQUERY messages to the specified DHCP server and accepts DHCPACTIVE, DHCPKNOWN and DHCPUNKNOWN replies from the DHCP server.

The following steps demonstrate how the BSR uses the DHCP lease query feature:

1. Cable Subscriber requests and gets an IP address from DHCP server.
2. Cable Subscriber starts to pass traffic through the cable interface.
3. The BSR inspects the cable network traffic to ensure source IP addresses are valid by doing the following:
   - Verify DHCP server acknowledgement messages to learn if IP packets are forwarded only once for an IP address.
   - Query the DHCP server to verify if an IP address was legally assigned by verifying DHCP lease information table. If it is confirmed that static IP address was assigned by a hacker for a CM, packets are not forwarded beyond the cable interface.
   - Disallow ARP broadcasts

- Query the DHCP server to verify that one IP address to MAC address binding appears for a CM. If there is more than one IP address to MAC address combination, one IP was assigned by DHCP and the other IP address is statically (manually) set by a hacker. In this instance, only packets sent from the legal source learned through DHCP are forwarded.

The DHCP lease query feature can also determine:

- If the BSR is replaced or inadvertently rebooted.
- When a CM re-registers and acquires a new lease.
- When a CM or CPE maintains its lease because it has not expired.
- When the cable interface learns about the DHCP lease through a DHCPLEASEQUERY exchange.
- When the CM or CPE can continue passing data.

Follow these steps to enable the DHCP lease query feature:

1. Use the **interface cable** command in Global Configuration mode to enter the desired cable interface, as shown below:

   ```
   MOT(config-if)#interface cable {<x>/<y>}
   ```

   where:

   x is the slot number of the cable module.

   y is the cable interface number, which is **0**.

2. Use the **dhcpleasequery authorization on** command in Interface Configuration mode to enable DHCP lease query messages to be exchanged between the cable interface and DHCP server, as shown below:

Note: The IP helper address must be entered for the DHCP Lease Query function to work. Refer to “Configuring the Cable Helper and IP Helper Addresses” on page 6-4 for more information on setting the IP helper address on the cable interface.

```
MOT(config-if)#dhcpleasequery authorization on
```

3. Use the **show ip traffic** command to monitor DHCP lease query statistics, which include the number of active, known, unknown, and unimplemented DHCP packet transmitions.

Setting ARP Parameters

The Address Resolution Protocol (ARP) is used to build a correlation between the cable network and the connected cable modems (CMs) and customer premises equipment (CPE) by translating the CM and CPE's MAC address to a logical IP address. The collected information is dynamically stored in a table called the ARP cache.

Follow these steps to set ARP parameters on a cable interface.

1. Enter the cable interface on which ARP is enabled.
2. Use the **arp** command in Interface configuration mode to specify the type of ARP packet that is used on the BSR 64000, as shown below:

   ```
   MOT(config-if)#arp [arpa | snap]
   ```

   where:

   arpa is entered for the standard ARP protocol.

   snap is entered for the IEEE 802.3 usage of ARP.

The ARP timeout feature is used to prevent unnecessary flooding of traffic over the cable network. ARP resolution requests are terminated after a defined interval when attempts to resolve addressing information, for a device entry in the ARP cache table.

3. The ARP cache table expiration value is disabled by default. Use the **arp timeout** command in Interface Configuration mode to set the ARP cache table expiration value, as shown below:

   ```
   MOT(config-if)#arp timeout <1-6000>
   ```

   where:

   1-6000 is the expiration value in minutes.

If you want to return to the default ARP timeout condition, use the **no arp timeout command** in Interface Configuration mode, as shown below:

```
MOT(config-if)#no arp timeout
```

Configuring a Downstream Channel

A downstream channel is configured to control the data flow from the cable interface to the user CM. This section is divided into two parts. The downstream parameters that must be configured for the minimal operation of the downstream port are discussed in the Initial Downstream Configuration Tasks section. The downstream parameters that are configured to manage the downstream channel operation are discussed in the Managing the Downstream Channel section.

Table 6-1 contains the downstream parameters.

Table 6-1 Downstream Parameters

Parameter	Identification	Default	Value
Frequency Range	Radio frequency carrier center frequency	555,000,000 Hz	93,000,000 to 855,000,000 Hz
Insertion Interval	Time available for CM initial channel request	2,000 milliseconds	100 to 2,000 milliseconds
Interleave Depth	Depth to provide protection from noise	32	8, 12, 16, 32, 64, 128,
Modulation Rate	Data traffic speed	256 QAM	64 QAM (6 bits) or 256 QAM (8 bits) per symbol
MPEG Framing Format	Annex A (European) standard Annex B (North American) standard	B	A or B
Power Level	Downstream transmit power level	55 dBmV	45 to 65 decibels per millivolt (dBmV)
SYNC Interval	Interval between SYNC message transmissions	200	1 to 200 milliseconds

Initial Downstream Configuration Tasks

Use the tasks in this section to perform the following basic downstream configuration tasks:

- Configuring the Downstream Frequency and Modulation Rate
- Enforcing the Downstream Rate Limit
- Enabling the Downstream Port

Configuring the Downstream Frequency and Modulation Rate

Follow the steps in this section to configure the downstream center frequency and modulation rate:

1. To enter the fixed center frequency for the downstream channel, use the **cable downstream frequency** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#cable downstream 0 frequency {<91000000 - 857000000>}
```

Note: The digital carrier frequency cannot be the same as the video carrier frequency.

where:

91000000 - *857000000* is the downstream frequency in Hertz.

To disable the downstream center frequency setting, use the **no cable downstream frequency** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable downstream 0 frequency {91000000 - 857000000}
```

2. Use the **cable downstream modulation** command in Interface Configuration mode to set the downstream modulation rate, as shown below:

```
MOT(config-if)#cable downstream 0 modulation [64 | 256]
```

where:

64 is 64 Quadrature Amplitude Modulation (QAM).

256 is 256 (QAM).

To restore the default, use the **no cable downstream modulation** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable downstream 0 modulation [64 | 256]
```

Enforcing the Downstream Rate Limit

Use the **cable downstream rate-limit** command to enable the downstream data transmission rate-limit to CMs on the HFC network. Once the downstream data transmission rate-limit function is enabled, data sent from the cable interface to the CMs is rate-limited according to each CM configuration. For example, a CM may drop packets when the data from the network exceed the permitted bandwidth of the CM. Follow the steps in this section to enable the downstream rate-limit for CMs:

1. Edit the CM configuration file to set the downstream data rate limit.
2. To enable the rate-limiting function, use the **cable downstream rate-limit** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#cable downstream rate-limit
   ```

   To disable the downstream rate-limiting function, use the **no cable downstream rate-limit** command.
3. To verify that downstream rate-limiting is enabled or disabled on the cable interface, use the **show running-config** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show running-config
   ```
4. Use the **show cable qos svc-flow statistics** command in Privileged EXEC mode to determine the number of dropped packets due to downstream rate-limiting for a particular service flow, as shown below:

   ```
   MOT#show cable qos svc-flow statistics {<x>/<y>} [<1-4292967295> | <cr>]
   ```

   where:

   x is the slot number of the cable module.

   y is the cable interface number, which is **0**.

1-4292967295 is the Service Flow Identifier (SFID).

cr is a command return, which displays QoS service flow statistics for all SFIDs.

Enabling the Downstream Port

The downstream port is in an administrative shut-down state by default and must be enabled to function.

Follow these steps to enable the downstream port:

1. To enable the downstream port, use the **no cable downstream shutdown** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#no cable downstream shutdown
   ```

2. To verify that the downstream port is enabled, use the **show interface cable** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#show interface cable <x>/<y>
   ```

   where:

   x is the slot number of the cable module.

   y is the cable interface number, which is **0**.

Managing the Downstream Channel

The tasks in this section involve some parameters that you may choose to change. If a parameter default is satisfactory, you can ignore its associated task.

The tasks in this section are used to manage the operation of the downstream channel:

- Configuring the Downstream Interleave Depth
- Setting the Downstream Power Level
- Resetting a Downstream Port
- Reserving Downstream Bandwidth
- Unreserving Downstream Bandwidth
- Testing RF Carrier Modulation

Configuring the Downstream Interleave Depth

The cable operator can protect the downstream path from excess noise or decrease latency on the downstream path by setting the interleave depth. A higher interleave depth provides more protection from noise on the HFC network, but increases downstream latency. A lower interleave depth decreases downstream latency, but provides less protection from noise on the HFC network.

1. Review Table 6-2 to determine the appropriate interleave-depth.

Table 6-2 Interleave Depth Criteria

Depth	# of Taps	Increments
8	8	16
16	16	8
32	32	4
64	64	2
128	128	1

Note: The Euro DOCSIS standard requires an interleave depth of 12, 12 Taps, and 17 increments.

2. To set the downstream port interleave depth, use the **cable downstream interleave-depth** command in Interface Configuration mode, as shown below:

Note: A higher interleave depth provides more protection from bursts of noise on the HFC network; however, it increases downstream latency.

```
MOT(config-if)#cable downstream <0-0> interleave-depth [8 | 16 | 32 | 64 | 128]
```

where:

0-0 is the downstream port number.

To restore the default, use the **no cable downstream interleave-depth** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable downstream 0 interleave-depth [8 | 16 | 32 | 64 | 128]
```

Setting the Downstream Power Level

Follow these options to adjust the downstream power level:

- The default downstream power level is 55 decibels per millivolt (dBmV). If you need to adjust the downstream power level, issue the **cable downstream power-level** command, as shown below:

  ```
  MOT(config-if)#cable downstream 0 power-level <450-630>
  ```

  where:

  450-630 is the downstream power level expressed in one tenth of a dB.

- To return to the default power-level setting, use the **no cable downstream power-level** command in Interface Configuration mode, as shown below:

  ```
  MOT(config-if)#no cable downstream 0 power-level <450-630>
  ```

  where:

  450-630 is the downstream power level expressed in one tenth of a dB.

Resetting a Downstream Port

Follow these steps to optionally reset a downstream port:

1. To optionally disable a downstream port if it must be reset, use the **cable downstream shutdown** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#cable downstream <0-0> shutdown
   ```

   where:

   0-0 is the downstream port number.

2. To enable the downstream port again, use the **no cable downstream shutdown** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**no cable downstream shutdown**

3. To verify that the downstream port is activated, use the **show interface cable** command in Privileged EXEC mode, as shown below:

   MOT#**show interfaces cable 0**

Reserving Downstream Bandwidth

To reserve a specified amount of downstream bandwidth for CMs, use the **cable downstream reserve-bandwidth** command, in Interface Configuration mode, as shown below:

MOT(config-if)#**cable downstream** <*0-0*> **reserve-bandwidth** <*1-38469736*>

where:

port is the downstream port number.

1-38469736 is the amount of downstream bandwidth in bits per second.

Unreserving Downstream Bandwidth

To unreserve a specified amount of downstream bandwidth for CMs, use the **cable downstream unreserve-bandwidth** command, in Cable Interface mode, as shown below:

MOT(config-if)#**cable downstream** <*0-0*> **unreserve-bandwidth** <*1-38469736*>

where:

0-0 is the downstream port number.

1-38469736 is the amount of downstream bandwidth in bits per second.

Testing RF Carrier Modulation

The downstream carrier-only function is disabled by default and is used for testing purposes only to control downstream output. To optionally enable this test function, use the in Interface Configuration mode, as shown below.

```
MOT(config-if)#cable downstream <0-0> carrier-only
```

where:

0-0 is the downstream port number.

To disable the downstream carrier-only function, use the **no cable downstream carrier-only** command.

Configuring an Upstream Channel

An upstream channel is configured to control the data flow from a CM to the cable interface. This section is divided into two parts. The initial upstream parameters that must be configured for the minimal operation of the upstream port are discussed in the Initial Upstream Configuration Tasks section. The upstream parameters that are configured to manage the upstream channel operation and performance are discussed in the Managing the Upstream Channel section.

Note: The cable interface does not operate until a fixed upstream frequency is set.

Table 6-3 describes the upstream parameters:

Table 6-3 Upstream Parameters

Parameter	Identification	Default	Value
Channel Width	Radio frequency channel width	1,600,000 Hz (1280 ksps)	200000, 400000, 800000, 1600000, 3200000 Hz (160, 320, 640, 1280, 2560 ksps)
Data Backoff	Initial ranging backoff fixed start and end values	Start 2 End 8	Start 0 to 15 End 0 to 15 Automatic
FEC	Forward Error Correction	On	On or Off
Frequency	Center frequency for CM use	None	5,000,000 to 42,000,000 Hz
Minislot Size	Port minislot size in number of time ticks	4 ticks (64 symbols)	2, 4, 8, 16, 32, 64 ticks (32, 64, 128, 256, 512, 1024, 2048 symbols)
Modulation Profile	Physical layer profile characteristics	5	1 to 16
Power Level	Input power level	0	-16 dBmV to +26 dBmV
Range Backoff	Initial ranging backoff start and end values	Start 0 End 4	Start 0 to 15 End 0 to 15 Automatic

Initial Upstream Configuration Tasks

Follow these tasks for the initial configuration of the upstream channel:

- Setting the Upstream Frequency
- Setting the Upstream Power Level
- Applying an Upstream Modulation Profile
- Enforcing the Upstream CM Rate Limit
- Enabling an Upstream Port

Setting the Upstream Frequency

The cable interface does not operate until a fixed upstream frequency is set. The RF upstream frequency must comply with the expected CM output frequency.

Note: Make sure that the upstream frequency selected does not interfere with the frequencies used for any other upstream applications running in the cable plant.

1. To set the upstream frequency for an upstream port, use the **cable upstream frequency** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#cable upstream <NUM> frequency
<5000000-42000000>
```

   where:

   NUM is the upstream port number.

   5000000-42000000 is the upstream frequency value in Hertz (Hz) for DOCSIS.

Note: Upstream frequency ranges are different depending on your regional implementation of DOCSIS or Euro-DOCSIS. The frequency ranges that appear in the CLI help are related to your implementation of DOCSIS. The upstream frequency value for Euro-DOCSIS is 5000000 to 65000000 Hz.

Setting the Upstream Power Level

The cable interface controls CM output power levels to meet the desired upstream port input power level. Input power level adjustments to an upstream port compensate for cable interface signal degradation between the optical receiver and the upstream RF port. You can configure the upstream input power level in either *absolute* or *relative* mode.

- If the upstream input power level is set to the *absolute* mode, the input power level does not change when the upstream channel width is changed. Defining the input power level in *absolute* mode could possibly cause upstream return lasers to clip on a completely populated upstream channel.
- If the upstream input power level is set in *relative* mode, the input power level changes when the upstream channel width is changed. For example, if the input power level is +11 dBmV for a DOCSIS 3.2 MHz upstream channel bandwidth setting in *relative* mode and is changed to 1.6 MHz, the default receive power is +8 dBmV. The default power levels for the 3.2 MHz and 1.6 MHz channels are equal relative to their respective channel bandwidth settings.

Caution: If the power level is not explicitly set on the upstream interfaces, they default to 0 dBmV in absolute mode with a 3.2 MHz, 2560 kilosymbols per second rate. Ensure that the correct power level is set on each upstream channel.

Table 6-4 describes how the upstream channel bandwidth corresponds to the input power-level range and default power-level range for a specific upstream channel.

Table 6-4 Upstream Input Power Level Range Parameters

Upstream Channel Bandwidth	Default Power-level Range	Power-level Range
200 KHz	-1 dBmV	-16 to +14 dBmV
400 KHz	+2 dBmV	-13 to +17 dBmV
800 KHz	+5 dBmV	-10 to +20 dBmV
1.6 MHz	+8 dBmV	-7 to +23 dBmV
3.2 MHz	+11 dBmV	-4 to +26 dBmV

Setting the Upstream Power Level in Relative Mode

Issue the **cable upstream power-level default** command in Interface Configuration mode to set the upstream input power level in *relative* mode, as shown below:

`MOT(config-if)#`**cable upstream** <*NUM*> **power-level default** <*offset*>

where:

NUM is the upstream port number.

offset is the number expressed in dB above or below the default input power level.

Example 1:

Issue the **cable upstream power-level default** command in Interface Configuration mode to set the input power level for a 3.2 MHz channel in *relative* mode from +11 dBmV to +5 dBmV, as shown below:

`MOT(config-if)#`**cable upstream 0 power-level default -60**

The default input power level is reduced by 6 dBmV. The power level is now +5 dBmV.

Example 2:

Issue the **cable upstream power-level default** command in Interface Configuration mode to set the input power level for a 3.2 MHz channel in *relative* mode from +11 dBmV to 0 dBmV, as shown below:

`MOT(config-if)#`**cable upstream 0 power-level default -110**

The default input power level is reduced by 11 dBmV.

Setting the Upstream Power Level in Absolute Mode

Issue the **cable upstream power-level** command in Interface Configuration mode to set the upstream input power level in *absolute* mode, as shown below:

Caution: Use caution when increasing the input power level in *absolute* mode. The CMs on the HFC network increase their transmit power level by 3 dB for every incremental upstream channel bandwidth change, causing an increase in the total power on the upstream channel. This may violate the upstream return laser design parameters

```
MOT(config-if)#cable upstream <NUM> power-level <power>
```

where:

NUM is the upstream port number.

power is the input power level, expressed in dB.

Example

Issue the **cable upstream power-level** command in Interface Configuration mode to set the upstream input power level to +5 dBmV in *absolute* mode, which keeps the input power level at +5 dBmV regardless of the upstream channel bandwidth setting, as shown below:

```
MOT(config-if)#cable upstream 0 power-level 50
```

Applying an Upstream Modulation Profile

Modulation profile 1 and 2 are the default upstream modulation profiles containing upstream burst parameters. Modulation profile numbers 1 or 2 can be applied to an upstream port, or other upstream modulation profiles that currently exist for the cable interface. Follow the steps in this section to apply a modulation profile to an upstream port:

1. To view all existing modulation profiles, use the **show cable modulation-profile** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show cable modulation-profile
   ```

2. To view a specific modulation profile, use the **show cable modulation-profile** command in Privilege EXEC mode, as shown below:

```
MOT#show cable modulation-profile [<1-16> | <cr>]
```

where:

1-16 is a profile group number.

cr is a command return, which displays all configured modulation profiles.

3. To apply an upstream modulation profile to an upstream port, use the **cable upstream modulation-profile** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#cable upstream <NUM> modulation-profile <1-16>
```

where:

NUM is the upstream port number.

1-16 is the modulation profile number.

To restore the default, use the **no cable upstream modulation profile** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable upstream <NUM> modulation-profile <1-16>
```

where:

NUM is the upstream port number.

1-16 is the modulation profile number.

Enforcing the Upstream CM Rate Limit

Use the **cable upstream rate-limit** command to enable the upstream data transmission rate-limit. This limits the traffic rate for data sent from the CMs to the cable interface.

Follow these steps to enable the upstream rate-limit for CMs.

1. Edit the CM configuration file to set the upstream data rate limit.
2. To enable the rate-limiting function, use the **cable upstream rate-limit** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#cable upstream <NUM> rate-limit
```

where:

NUM is the upstream port number.

To disable the upstream rate-limiting function, use the **no cable upstream rate-limit** command.

3. To verify that upstream rate-limiting is enabled or disabled on the cable interface, use the **show running-config** command in Privileged EXEC mode, as shown below:

```
MOT#show running-config
```

4. To determine the number of packets dropped due to upstream rate-limiting for a particular service flow, use the **show cable qos svc-flow statistics** command in Privileged EXEC mode, as shown below:

```
MOT#show cable qos svc-flow statistics {<x>/<y>} [<1-4292967295> | <cr>]
```

where:

x is the slot number of the cable module.

y is the cable interface number, which is **0**.

1-4292967295 is the Service Flow Identifier (SFID).

cr is a command return, which displays QoS service flow statistics for all SFIDs.

Enabling an Upstream Port

The upstream ports are in an administrative shut down state by default.

Follow these steps to enable the upstream ports:

1. To determine if an upstream port is activated or deactivated, use the **show interface cable** command in Privileged EXEC mode, as shown below:

```
MOT#show interface cable
```

2. The upstream ports are in a shutdown state by default. To enable the upstream ports, use the **no cable upstream shutdown** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable upstream <NUM> shutdown
```

where:

NUM is the upstream port number.

Managing the Upstream Channel

The following upstream tasks in this sections are used to manage or improve the performance of the upstream channel:

- Configuring Upstream CM Registration Parameters
- Adjusting for Physical Delay between Cable Interface and CMs
- Activating Upstream Forward Error Correction
- Activating the Scrambler on the Upstream Channel
- Enabling Pre-equalization
- Forcing the Fragmentation of Large Upstream Packets
- Disabling an Upstream Port
- Configuring the Upstream Channel Descriptor
- Limiting the Number of Voice Calls on an Upstream Channel

Configuring Upstream CM Registration Parameters

Configuring upstream CM registration parameters includes the following options:

- Setting the Upstream Minislot Size
- Setting the Upstream Channel Width
- Setting the Upstream Range-backoff
- Forcing a Range-response
- Forcing a Range Power Override
- Setting the Upstream Data-backoff
- Configuring the Invited Ranging Interval
- Configuring the Map Interval

Follow these options to configure upstream CM registration parameters:

- The minislot size is the number of time ticks. To set the upstream minislot size, use the **cable upstream minislot-size** command in Interface Configuration mode, as shown below:

  `MOT(config-if)#`**cable upstream** *<NUM>* **minislot-size** [**2** | **4** | **8** | **16** | **32** | **64** | **128**]

where:

NUM is the upstream port number.

To reset the upstream minislot size, use the **no cable upstream minislot-size** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable upstream <NUM> minislot-size [2 | 4 | 8 | 16 | 32
| 64 | 128]
```

where:

NUM is the upstream port number.

- Issue the **cable upstream channel-width** command in Interface Configuration mode to set the upstream channel width in Hertz (Hz), as shown below:

```
MOT(config-if)#cable upstream <NUM> channel-width [200000 |
400000 | 800000 | 1600000 | 3200000]
```

where:

NUM is the upstream port number.

To reset the default, use the **no cable upstream channel-width** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable upstream <NUM> channel-width [200000 |
400000 | 800000 | 1600000 | 3200000]
```

where:

NUM is the upstream port number.

- Use the **cable upstream range-backoff** command in Interface Configuration mode to set the start and end upstream range-backoff values for a CM or re-establish a CM if a power outage occurs, as shown below. If you choose automatic, the system sets the upstream data-backoff start and end values:

```
MOT(config-if)#cable upstream <NUM> range-backoff [automatic |
<0-15> | <0-15>]
```

where:

NUM is the upstream port number.

0-15 is the ranging backoff start value.

0-15 is the ranging backoff end value.

Use the **no cable upstream range-backoff** command in Interface Configuration mode to return the range-backoff function to the default, as shown below:

MOT(config-if)#**no cable upstream** <*NUM*> **range-backoff** [**automatic** | <*start*> | <*end*>]

where:

automatic prompts the ranging backoff to start and end automatically.

NUM is the upstream port number.

0-15 is the ranging backoff start value.

0-15 is the ranging backoff end value.

- To enable CM power adjustment, use the **cable upstream range-power-override** command in Interface Configuration mode, as shown below:

MOT(config-if)#**cable upstream** <*NUM*> **range-power-override**

where:

NUM is the upstream port number.

To disable the power adjustment, use the **no cable upstream range-power-override** command in Interface Configuration mode, as shown below:

MOT(config-if)#**no cable upstream** <*NUM*> **range-power-override**

where:

NUM is the upstream port number.

- To set the upstream data-backoff start and end values, use the **cable upstream data-backoff** command in Interface Configuration mode, as shown below. If you choose automatic, the system selects the upstream data-backoff start and end values.

MOT(config-if)#**cable upstream** <*NUM*> **data-backoff** [**automatic** | <*0-15*> | <*0-15*>]

where:

NUM is the upstream port number.

0-15 is the start value.

0-15 is the end value.

To restore the upstream data-backoff default, use the **no cable upstream data-backoff** command in Interface Configuration mode, as shown below:

`MOT(config-if)#`**no cable upstream** <*NUM*> **data-backoff** [**automatic** | <*0-15*> | <*0-15*>]

where:

NUM is the upstream port number.

0-15 is the start value.

0-15 is the end value.

- The default number of retries allowed by the cable interface for inviting ranging requests transmitted by the CM is 10,000. If you need to adjust the number of inviting ranging request retries, issue the **cable upstream invited-range-interval** command in Interface Configuration mode, as shown below:

  `MOT(config-if)#`**cable upstream** <*NUM*> **invited-range-interval** <*0-30000*>

  where:

  NUM is the upstream port number.

  0-30000 is the number of invited range interval requests.

- To determine the time interval in microseconds for bandwidth maps messages (MAP) to be used by the CM to allocate upstream time slots, use the **cable upstream map-interval** command in Interface Configuration mode, as shown in the following command example:

  `MOT(config-if)#`**cable upstream** <*NUM*> **map-interval** <*2000-16000*>

  where:

  NUM is the upstream port number.

  2000-16000 is the time interval in microseconds.

Adjusting for Physical Delay between Cable Interface and CMs

The physical delay function is used to adjust the upstream propagation delay threshold between the cable interface and cable modems (CMs). The cable interface adjusts the physical delay function automatically by default.

You can use the following options to adjust the physical delay function:

- A single fixed time can be set for physical delay.
- Physical delay parameters can be configured so that they are adjusted automatically by the BSR when you specify the automatic option with a specified minimum and maximum microsecond range.
- If you do not want to specify a range for the automatic option, select the automatic option only.

Issue the **cable upstream physical-delay automatic** command in Interface Configuration mode to set the *automatic* physical delay value for an upstream channel, as shown below:

```
MOT(config-if)#cable upstream <NUM> physical-delay automatic
[<200-1600> | <200-1600> | <cr>]
```

where:

NUM is the upstream port number.

automatic indicates adjust the physical delay automatically.

200-1600 is the minimum upstream physical delay from microseconds.

200-1600 is the maximum upstream physical delay from 200 to 1600 microseconds.

cr is a command return used after the **automatic** parameter to specify that the physical delay is adjusted automatically.

-or-

Issue the **cable upstream physical-delay** command in Interface Configuration mode to set the *fixed* value for an upstream channel, as shown below:

```
MOT(config-if)#cable upstream <NUM> physical-delay {<200-1600>}
```

where:

NUM is the upstream port number.

automatic indicates adjust the physical delay automatically.

200-1600 is the fixed upstream physical delay value from 200 to 1600 microseconds.

Activating Upstream Forward Error Correction

The cable interface uses Forward Error Correction (FEC) to correct any corrupt upstream data. FEC is enabled by default and should not be disabled. When FEC is enabled, all CMs on the network also activate FEC. If you must re-enable upstream FEC, use the **cable upstream fec** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#cable upstream <NUM> fec
```

where:

NUM is the upstream port number.

To disable upstream FEC, use the **no cable upstream fec** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable upstream <NUM> fec
```

where:

NUM is the upstream port number.

Activating the Scrambler on the Upstream Channel

The scrambler on the upstream channel enables CMs on the HFC network to use built-in scrambler circuitry for upstream data transmissions. The scrambler circuitry improves BSR upstream receiver. The upstream scrambler is enabled by default and should not be disabled.

Note: Scrambler must be enabled for normal operation. Disable only for prototype modems that do not support scrambler.

If you must re-enable the upstream scrambler, use the **cable upstream scrambler** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#cable upstream <NUM> scrambler
```

where:

NUM is the upstream port number.

To disable the upstream scrambler, use the **no cable upstream scrambler** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable upstream <NUM> scrambler
```

Enabling Pre-equalization

Use the **cable upstream pre-equalization** command to enable the pre-equalization adjustment function on the upstream port, which includes sending pre-equalization coefficients in a ranging response to a CM to compensate for impairment over the transmission line in Interface Configuration mode, as shown below:

Note: Not all CMs support the pre-equalization adjustment. If you enable the pre-equalization adjustment for an upstream port and the CM does not support this adjustment, the cable interface may not receive valid upstream data from the CM.

```
MOT(config-if)#cable upstream <NUM> pre-equalization
```

where:

NUM is the upstream port number.

If you need to disable the pre-equalization adjustment, use the **no cable upstream pre-equalization** command.

Forcing the Fragmentation of Large Upstream Packets

The **cable upstream force-frag** command is used as a traffic shaping tool. When a CM sends a request to the cable interface for a large data grant that exceeds the configured minislot threshold, the cable interface grants the CM the configured minislot threshold, which forces the CM to make another data grant request for the remaining data, thereby causing the data packets to be fragmented by the CM.

Use the **cable upstream force-frag** command to force CMs to fragment large upstream packets in Interface Configuration mode, as shown below:

```
MOT(config-if)#cable upstream <NUM> force-frag <0-255>
```

where:

NUM is the upstream port number.

0-255 is the threshold number of minislots without fragmentation for large data grants.

Disabling an Upstream Port

Follow these steps to administratively shut down an upstream port:

1. To determine if an upstream port is activated or deactivated, use the **show interface cable** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show interfaces cable 0
   ```

2. To disable an upstream port, use the **cable upstream shutdown** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#cable upstream <NUM> shutdown
   ```

   where:

   NUM is the upstream port number.

Configuring the Upstream Channel Descriptor

Follow the steps in this section to configure the upstream channel descriptor (UCD).

1. To disable the upstream channel, use the **cable upstream shutdown** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**cable upstream** <*NUM*> **shutdown**

   where:

   NUM is the upstream port number.

2. The default UCD message transmission interval is 1000 milliseconds. If you want to adjust this parameter, use the **cable ucd-interval** command in Interface Configuration mode to set the UCD message transmission interval, as shown below:

   MOT(config-if)#**cable ucd-interval** <*0-2000*>

   where:

   0-2000 is the maximum UCD message transmission interval in milliseconds.

3. To display the configured ucd interval value, use the **show cable ucd-interval** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**show cable ucd-interval**

4. To re-enable the upstream channel, use the **no cable upstream shutdown** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**no cable upstream** <*NUM*> **shutdown**

   where:

   NUM is the upstream port number.

Limiting the Number of Voice Calls on an Upstream Channel

The Maximum Assigned Bandwidth (MAB) feature is used on the cable interface to regulate the number of Voice-over-IP (VOIP) calls that are available on a particular upstream channel for Unsolicited Grant Service (UGS) and Unsolicited Grant Service with Activity Detection UGS-AD constant bit rate (CBR) data flows. A definitive limit on the number of voice calls ensures that bandwidth resources are not overused on an upstream channel.

- The maximum number of calls is set to 32 by default. Use the **cable upstream max-calls** command in Interface Configuration mode to configure the maximum number of voice calls for an upstream channel, as shown below:

  MOT(config-if)#**cable upstream** <*NUM*> **max-calls** <*0-255*>

  where:

  NUM is the upstream port number.

  0-255 is the number of voice calls permitted on the upstream channel.

- Use the **no cable upstream max-calls** command in Interface Configuration mode to return the maximum number of voice calls to the default value, which is zero.

  MOT(config-if)#**no cable upstream** <*NUM*> **max-calls** <*0-255*>

  where:

  NUM is the upstream port number.

  0-255 is the number of voice calls permitted on the upstream channel.

Bundling Cable Interfaces into a Single IP Subnet

Cable bundling allows you to group multiple cable interfaces into a single IP subnet. Cable bundling simplifies network management and conserves IP addresses.

Each BSR 64000 DOCSIS module provides one cable interface. A cable bundle comprises two or more cable interfaces: one cable interface is configured as the master, while the remaining interfaces are configured as slaves to the master. If one DOCSIS module is configured as the master, the other DOCSIS modules can become slaves. The master cable interface is assigned a single IP address and the slaves share the same IP address with the master. Therefore, the bundling feature eliminates the need for an IP subnet for each cable interface.

The cable bundle feature provides the following benefits:

- You can add new DOCSIS modules without having to assign an IP address to each new DOCSIS module or to cable modems on each interface.
- You can move a cable modem that has a static IP address to any cable interface on the same bundle without assigning the cable modem a new IP address.

- You can bundle all cable interfaces into a single bundle to share a single IP subnet.

The cable bundling feature requires that the following conditions are observed:

- Cable interface bundling is only supported on cable interfaces.
- One cable interface must be configured as the master interface for the bundle. The other cable interfaces are configured as slave interfaces.
- An IP address is only assigned to master cable interface.
- DHCP relay is only enabled on the master cable interface.
- ARP authorization is enabled on both the master cable interface and the slave cable interfaces.
- Cable upstream and downstream parameters remain unique for each individual cable interface.
- Cable interface bundles are configured using CLI commands only.

Creating a Cable Bundle

Follow these steps to configure a cable bundle on the BSR 64000:

1. To enter the cable interface that you want to designate as the master cable interface, use the **interface cable** command in Global Configuration mode as shown below:

   ```
   MOT(config)#interface cable <x>/<y>
   ```

   where:

   x is the slot number of the cable module.

   y is the cable interface number, which is **0**.

   For example:

   ```
   MOT(config)#interface cable 4/0
   ```

2. To make sure that DHCP relay is enabled on the master cable interface, use the **show running-config** command in Privileged EXEC mode. Check the command output to see if the IP helper address or cable helper address is assigned to the master cable interface.

3. If the IP helper address or cable helper address is not assigned to the master cable interface, use the **ip helper-address** or **cable helper-address** command in Interface configuration mode to enable DHCP relay. The ip helper-address or cable helper address specifies the DHCP server.

4. To determine if an IP address is assigned to the master cable interface, use the **show interface cable** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#show interface cable <x>/<y>
   ```

   where:

   x is the slot number of the cable module.

   y is the cable interface number, which is **0**.

5. If the master cable interface does not have IP address, use the **ip address** command in Interface Configuration mode to specify the master cable interface IP address.

6. To assign the cable interface as the master cable interface and assign the bundle a number, use the **cable bundle master** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#cable bundle <0-255> [master | <cr>]
   ```

   where:

   0-255 is the number of the cable bundle identifier.

   cr is a command return.

   For example:

   ```
   MOT(config-if)#cable bundle 1 master
   ```

7. To exit the master cable interface, use the **end** command in Interface Configuration mode.

8. To make sure that the slave cable interface does not have an IP address assigned to it, use the **show interface cable** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#show interface cable <x>/<y>
   ```

   where:

   x is the slot number of the cable module.

y is the cable interface number, which is **0**.

9. To assign another cable interface as the slave interface, use the **interface cable** command in Global Configuration mode as shown below:

   `MOT(config)#`**interface cable** <*x*>/<*y*>

   where:

   x is the slot number of the cable module.

   y is the cable interface number, which is **0**.

   For example:

   `MOT(config)#`**interface cable 5/0**

10. To assign this cable interface as the slave cable interface and assign the bundle the same number as the master cable interface, use the **cable bundle** command in Interface Configuration mode, as shown below:

    `MOT(config-if)#`**cable bundle** <*0-255*>

    where:

    0-255 is the number of the cable bundle identifier.

    For example:

    `MOT(config-if)#`**cable bundle 1**

11. To verify your cable bundle configuration for each cable interface, use the **show running-config** command in Privileged EXEC mode.

The examples in this section show that the DOCSIS module master cable interface is in slot 4 and the DOCSIS module slave cable interface is in slot 5. Both modules are in the same IP subnet that is described as cable bundle 1.

Here is the entire example configuration discussed in this section:

```
MOT(config)#interface cable 4/0
MOT(config-if)#cable bundle 1 master
MOT(config-if)#end
MOT(config)#interface cable 5/0
MOT(config-if)#cable bundle 1
MOT(config-if)#end
```

Adding a Static Arp Entry to a Cable Bundle Interface

To optionally add a static arp entry to the cable bundle interface, use the **arp cable bundle cable** command in Global Configuration mode as shown below:

MOT(config)#**arp** <*A.B.C.D*> <*H.H.H*> [**arpa** | **snap**] **cablebundle cable** <*x*>/<*y*>

where:

A.B.C.D is the IP address of the ARP entry.

H.H.H is the 48-bit MAC address of the ARP entry.

The following example shows how to add a static arp entry that has an IP address of 10.1.1.10 and MAC address of 0000.0000.0010 MAC in arpa format on cable bundle interface cable 3/0:

MOT(config)#**arp 10.1.1.10 0000.0000.0010 arpa cablebundle cable 3/0**

Subneting DHCP Clients on the Cable Interface

The DHCP-relay agent, which is the router (SRM) between the cable interface and DHCP server, monitors DHCP CPE host requests for the presence of the Vendor Class Identifier (VCI) also known as “DHCP Relay Option 60”. Vendors define a VCI to optionally identify the DHCP client vendor type and configuration information. For example, if a VCI identifies a DHCP client as a Multimedia Terminal Adapter (MTA) device, the DHCP server can put all MTA devices that are on a cable interface into the same subnet.

The **host** and **mta** VCI options are used to configure a CPE host or MTA gateway IP address (giaddr) for the cable interface. During the DHCP process, the DHCP relay agent requests an IP address in a particular subnet by inserting the cable interface giaddr into the DHCP requests from CMs, hosts, and MTAs. The primary IP address is always inserted in CM DHCP requests. The **ip dhcp relay information option** command must be enabled to allow the BSR to determine what type of device originated the DHCP request if one or more secondary giaddr IP addresses are defined for a secondary CM host or MTA device. The primary IP address for the cable interface is inserted into DHCP requests by default.

Follow these steps to enable a VCI on the cable interface:

1. Enter the desired cable interface.

2. Use the **ip address secondary** command to define the Gateway IP address (giaddr) for CPE host DHCP requests or MTA DHCP requests that creates individual subnets for host CPEs, and MTAs, as shown below:

Note: The primary IP address for the cable interface is used for CM DHCP requests.

```
MOT(conf-if)# ip address {<A.B.C.D> <A.B.C.D>} secondary {host | mta |
<cr>}
```

where:

A.B.C.D is the IP address.

A.B.C.D is the subnetwork IP address mask.

secondary specifies that the secondary IP address is a secondary IP address.

host optionally defines this secondary IP address as the giaddr to be inserted into CPE host DHCP requests.

mta optionally defines this secondary IP address as the giaddr to be inserted into MTA DHCP requests.

<*cr*> is a command return that defines a secondary IP address for CM DHCP requests.

Creating a Modulation Profile

Modulation profile 1 and 2 are the default upstream modulation profiles containing upstream burst parameters. Modulation profile numbers 1 or 2 can be modified, or new upstream modulation profiles can be created for the cable interface. Follow the steps in this section to modify or configure a new modulation interface profile to set upstream burst parameters.

1. To view all existing modulation profiles, use the **show cable modulation-profile** command in Privileged EXEC mode, as shown below:

```
MOT#show cable modulation-profile
```

2. To view a specific modulation profile, use the **show cable modulation-profile** command in Privilege EXEC mode, as shown below:

   ```
   MOT#show cable modulation-profile {<1-16>}
   ```

   where:

   1-16 is the profile group number.

3. To create a modulation profile that contains the initial ranging burst parameters, use the **cable modulation-profile** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#cable modulation-profile {<1-16>} [initial <0-10> | long
   <0-10> | request <0-10> | short <0-10> | station <0-10>] {<16-23>} {<0-255>}
   16qam [scrambler | no scrambler] qpsk [scrambler <0x0-0x7fff> | no
   scrambler <0x0-0x7fff>] [diff | no-diff] <64-256> [fixed | shortened]
   ```

   Table 6-5 provides a description of each parameter listed in the **cable modulation-profile** command syntax above and what value that needs to be entered for each parameter:

Table 6-5 Modulation Profile Parameters and Descriptions

Parameter	Description
1-16	Profile group number.
initial <0-10>	Interval Usage Code (IUC) for intial ranging burst.
long <0-10>	IUC for long grant burst.
request <0-10>	IUC for request burst.
short <0-10>	IUC for short grant burst.
station <0-10>	IUC for station ranging burst.
16-253	FEC code word length.
0-255	Maximum burst length in minislots. 0 means no limit.
16qam [**scrambler** <0x0-0x7fff> | **no scrambler**]	Modulation type. If enabling the scramber option identify the scrambler seed in hexidecimal format.
qpsk [**scrambler** | **no scrambler**]	Modulation type. Enable or disable the scramber option identify the scrambler seed in hexidecimal format.
diff	Enable differential encoding.
no-diff	Disable differential encoding.

Table 6-5 Modulation Profile Parameters and Descriptions

Parameter	Description
64-256	Preamble length in bits.
fixed	Handles the FEC for last code word.
shortened	

4. To remove a modulation profile parameter, use the **no cable modulation-profile** in Privileged EXEC mode, as shown below:

   ```
   MOT#no cable modulation-profile {<1-16>}
   ```

   where:

   > *1-16* is the profile group number.

5. To view the modulation profile groups and to verify your changes, use the **show cable modulation-profile** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show cable modulation-profile {<1-16>}
   ```

   where:

   > *1-16* is a profile group number.

Setting Network Parameters for Cable Modems

Follow these tasks to configure parameters for cable modems (CMs) on your network:

- Enabling the CM Aging Timer
- Setting the Insertion Interval for CMs
- Setting the Synchronization Interval
- Setting CM Authentication Parameters
- Denying Access to a Cable Modem
- Setting the Maximum Number of Hosts

Enabling the CM Aging Timer

The cable modem (CM) aging timer feature is disabled by default. The CM aging timer feature is used to automatically remove off-line CMs from the network after a configured time period.

Use the **cable modem-aging-timer** command in Global Configuration mode to set and enable the CM aging timer, as shown below:

```
MOT(config)#cable modem-aging-timer {<10-30240>}
```

where:

10-30240 is the CM Aging Timer number in minutes (10 minutes to 21 days).

Use the **cable modem-aging-timer off** command to disable the aging timer.

Setting the Insertion Interval for CMs

The insertion interval is the fixed time period available for CM initial channel request. The default insertion interval is 20.

1. To set the insertion interval for CM initial channel request, use the **cable insert-interval** command in Interface Configuration mode, as shown below:

Note: Ensure that the upstream port is down before setting the insertion interval.

```
MOT(config-if)#cable insert-interval {<0-200>}
```

where:

0-200 is the insertion interval.

2. To return the default insertion interval, use the **no cable insertion-interval** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable insert-interval {<0-200>}
```

where:

0-200 is the insertion interval.

3. To view the insertion interval, use the **show cable insert-interval** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#show cable insert-interval
```

Setting the Synchronization Interval

The synchronization message interval is the interval between successive synchronization message transmissions from the cable interface to the CMs.

Follow the steps in this section to set and verify the synchronization interval:

1. To set the synchronization message interval value, use the **cable sync-interval** command in Interface Configuration mode, as shown below:

Note: Ensure that the interface is down before setting the synchronization message interval.

```
MOT(config-if)#cable sync-interval {0-200}
```

where:

0-200 is the maximum synchronization interval set in milliseconds (msecs).

2. To verify the synchronization message interval setting, use the **show running-config** command in Privileged EXEC mode. Until an interval is set, the `no sync interval` entry appears in the display.

   To reset the default synchronization message interval, use the **no cable sync-interval** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable sync-interval {1-200}
```

3. To display the synchronization message interval, use the **show cable sync-interval** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#show cable sync-interval
```

To reset the default synchronization message interval, use the **no cable sync-interval** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#no cable sync-interval {1-200}
```

Setting CM Authentication Parameters

The default authentication parameters are enabled, but have a null value by default. Set authentication parameters on the cable interface and the CMs to ensure security on the HFC network.

1. Use one the following two options to configure CM authentication parameters so that all CMs return a known text string to register with the cable interface for network access:

Caution: Ensure that the authentication string or hexadecimal key in the CM configuration file matches the authentication string or hexadecimal key configured on the cable interface. CMs cannot register with the cable interface if the authentication parameters do not match.

- If you want to activate CM authentication so that all CMs return an unencrypted text string to register with the cable interface for network access, use the **cable shared-secret 0** command in Global Configuration mode, as shown below:

  ```
  MOT(config)#cable shared-secret 0 <"authentication string">
  ```

  where:

  0 specifies that an unencrypted authentication text string follows.

  “authentication string” is an alphanumeric text string specified in double quotation marks.

Note: The "authentication string" on the BSR can be up to 253 characters.

- If you want to activate CM authentication on the cable interface so that all CMs return a encrypted key to register with the cable interface for network access, use the **cable shared-secret 7** command in Global Configuration mode, as shown below:

  MOT(config)#**cable shared-secret 7** <*key*>

  where:

  7 indicates a hidden hexadecimal key.

  key is the shared secret key, expressed in hexadecimal notation, for example, 0x434F5453.

2. To determine if CM authentication is activated or deactivated, use the **show running-config** command in Privileged EXEC mode. If CM authentication is active, CM authentication information does not appear in the display.

MOT#**show running-config**

Restoring Previously Defined Authentication Parameters

Use the **no cable shared-secret** command in Interface Configuration mode to restore the previously defined authentication parameters, as shown below:

MOT(config)#**no cable shared-secret** [<**“***authentication string***”**> | <*key*>]

where:

“authentication string” is an alphanumeric text string. You must specify this using double quotation marks.

key is the shared secret key, expressed in hexadecimal notation, for example, 0x434F5453.

Denying Access to a Cable Modem

Use the **cable modem deny** command in Privileged EXEC mode to remove a specified cable modem from the network and deny it future entry, as shown below:

MOT#**cable modem deny** <*mac*>

where:

mac is the MAC address of the cable modem.

Use the **no cable modem deny** command to remove the restriction from the specified cable modem, as shown below:

MOT#**no cable modem deny** <*mac*>

where:

mac is the MAC address of the cable modem.

Setting the Maximum Number of Hosts

Use the **cable modem max-hosts** command to set the number of CPE hosts that can connect to a CM on the HFC subnetwork.

1. To specify the maximum number of CPE hosts that can attach to a CM, use the **cable modem max-hosts** command in Privileged EXEC Mode, as shown below:

MOT#**cable modem** [<*mac*> | <*prefix*>] **max-hosts** <*0-32*>

where:

mac is the CM MAC address.

prefix is the CM IP address.

0-32 is the maximum number of hosts.

2. To verify the maximum number of hosts setting, use the **show cable modem hosts** command in Privileged EXEC mode, as shown below:

```
MOT#show cable modem {<mac> | <prefix>} hosts
```

where:

mac is the CM MAC address.

prefix is the CM IP address.

The screen displays the current number of hosts connected to the CM, the maximum number of hosts allowed for the CM, and the host CPE IP addresses behind the CM.

Configuring Baseline Privacy

This section contains the tasks to configure Baseline Privacy (BPI). To encrypt upstream and downstream data on an HFC network, you must configure BPI. BPI is activated by default and, in most cases, the BPI parameter default values are satisfactory. The optional tasks described in this section involve some parameters you may choose to change. If a parameter default is satisfactory, you can ignore its associated task.

You can set the Traffic Encryption Key (TEK) and Authorization Key (AK) for BPI. The encryption is based on 40-bit or 56-bit Data Encryption Standard (DES) algorithms.

You can set the TEK to expire based on a grace-time value or a lifetime value. A grace-time key assigns a temporary key to a CM to access the network. A lifetime key assigns a more permanent key to a CM. Each CM that has an assigned lifetime key requests a new lifetime key from the cable interface before the current key expires.

Note: The configuration and activation of BPI depend on the cable operator physical plant.

Configuring BPI involves the following tasks:

- Setting TEK Privacy
- Setting Authorization Key Values
- Displaying BPI Configuration Settings

Table 6-6 describes the BPI parameters.

Table 6-6 BPI Parameters

Parameter	Identification	Default	Value
AK grace-time	Temporary AK assigned to the CM	600 seconds	300 to 1,800 seconds
AK lifetime	More permanent AK assigned to the CM after grace-time AK expires	604,800 seconds	1 to 6,048,000 seconds
TEK grace-time	Temporary traffic key assigned to CM	600 seconds	300 to 1,800 seconds
TEK lifetime	More permanent TEK assigned to CM after grace-time TEK expires	43,200 seconds	1,800 to 6,048,000 seconds

Setting TEK Privacy

The TEK is assigned to a CM when its Key Encryption Key is established during the CM registration process. The TEK encrypts data traffic between the CM and the cable interface. The cable interface assigns a temporary grace-time TEK to a CM so the CM can access the network. When the grace-time TEK expires, the CM must renew its grace-time TEK with a lifetime TEK. The cable interface assigns a more permanent lifetime TEK to a CM when the grace-time TEK expires.

1. To enter Interface Configuration mode from Global Configuration mode to configure the cable interface, use the **interface cable** command, as shown below:

   ```
   MOT(config)#interface cable <x>/<y>
   ```

   where:

   x is the slot number of the cable module.

   y is the cable interface number.

2. Use the **cable privacy tek life-time** command in Interface Configuration mode to configure the global TEK lifetime value, as shown below:

   ```
   MOT(config-if)#cable privacy tek life-time {<30-604800>}
   ```

   where:

   30-604800 is the TEK life-time value in seconds (the maximum value is 7 days).

   To restore the default, use the **no cable privacy tek life-time** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#no cable privacy tek life-time {<30-604800>}
   ```

   To set an individual CM grace-time TEK value for Baseline Privacy, use the **cable privacy cm-tek grace-time** command in Interface Configuration mode, as shown below. Use the **no cable privacy cm-tek grace-time** command to reset the default.

   ```
   MOT(config-if)#cable privacy cm-tek grace-time {<1-8192>
   <300-302399>}
   ```

   where:

   1-8192 is the service-id (SID).

   300-302399 is the grace-time value, expressed in seconds (5 minutes to 3.5 days).

3. If a new grace-time value is set for one or more CMs, use the **cable privacy cm-tek reset** command in Interface Configuration mode to reset the grace-time CM TEK value, as shown below:

   ```
   MOT(config-if)#cable privacy cm-tek reset
   ```

4. The default lifetime TEK is 43,200 seconds. Use the **cable privacy cm-tek life-time** command to set a lifetime TEK for an individual CM in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#cable privacy cm-tek life-time {<0-16383>}
   [<1800-604800> | <cr>]
   ```

   where:

   0-16383 is the CM’s primary SID.

   1800-604800 is the TEK lifetime value, expressed in seconds.

   cr is a command return enables the default TEK lifetime value.

5. Use the **show cable privacy tek** command in Interface Configuration mode to display current TEK lifetime and grace-time information, as shown below:

   ```
   MOT#show cable privacy tek
   ```

Setting Authorization Key Values

An Authorization Key (AK) has a limited lifetime and must be periodically refreshed. A CM refreshes its AK by re-issuing an Authorization Request to the cable interface. Follow the steps in this section to configure the AK values.

1. Use the **cable privacy cm-auth life-time** command in Interface Configuration mode to set the AK lifetime value, as shown below:

   ```
   MOT(config-if)#cable privacy cm-auth life-time {<mac>}
   [<300-6048000> | <cr>]
   ```

   where:

   mac is the CM MAC address.

   300-6048000 is the AK lifetime value, expressed in seconds.

   cr is a command return that enables the default AK lifetime value.

2. Use the **cable privacy cm-auth grace-time** command in Interface Configuration mode to set an individual grace-time AK value, as shown below:

   ```
   MOT(config-if)#cable privacy cm-auth grace-time <mac>
   <300-3024000>
   ```

   where:

   mac is the CM MAC address.

   300-3024000 is the individual grace-time AK, expressed in seconds (5 minutes to 35 days).

3. Use the **cable privacy cm-auth reset** command in Interface Configuration mode to reset the individual CM life-time or grace-time value once a new AK life-time or grace-time value is configured, as shown below:

   ```
   MOT(config-if)#cable privacy cm-auth reset <mac>
   ```

   where:

   mac is the CM MAC address.

4. Use the **show cable privacy cm-auth** command in Interface Configuration mode to display AK information for an individual CM using its MAC address, as shown below:

   ```
   MOT(config)#show cable privacy cm-auth <mac>
   ```

   where:

   mac is the CM MAC address.

5. Use the **show cable privacy auth** command to view Authorization Key (AK) grace-time and life-time values in all modes except User EXEC mode, as shown below:

   ```
   MOT(config-if)#show cable privacy auth
   ```

Setting QoS Parameters

This section describes how to configure Quality of Service (QoS) using service flows. Use the commands in this section to create, change, or delete service flows with Dynamic Service Addition (DSA), Dynamic Service Change (DSC), and Dynamic Service Deletion (DSD) MAC management messages. Configuring QoS involves the following tasks:

- Initiating a DSA
- Initiating a DSC
- Initiating a DSD
- Viewing QoS Information

Initiating a DSA

Use a DSA message to create a new service flow. The cable interface initiates a DSA for a specified CM.

Note: Before you begin, ensure that the correct DSA definition is entered in the CM configuration file.

1. To create a new service flow, use the **cable modem qos dsa** command in Privileged EXEC mode, as shown below:

   MOT#**cable modem** {<*mac*> | <*prefix*>} **qos dsa** {<*prefix*> <*string*>}

   where:

   mac is the CM MAC address.

   prefix is the CM IP address.

   prefix is the TFTP server IP address.

   string is the CM configuration file name.

2. Use the **show cable qos svc-flow dynamic-stat** command in Privileged EXEC mode to display statistics for dynamic service additions, deletions, and changes for both upstream and downstream service flows, as shown below:

   `MOT#`**show cable qos svc-flow dynamic-stat**

Initiating a DSC

Use a DSC messages to change an existing service flow. The cable interface initiates a DSC for a CM.

Note: Before you begin, ensure that the correct DSA definition is entered in the CM configuration file. Also ensure that the DSC definition applies to each SFID. The CM configuration file must contain the correct SFID for the service flow you change.

1. to display the SFID of all the service flows used by a specific CM, Use the **show cable modem svc-flow-id** command in Privileged EXEC mode , as shown below:

   `MOT#`**show cable modem** {<*mac*> | <*prefix*>} **svc-flow-id**

   where:

   mac is the CM MAC address.

   prefix is the CM IP address.

2. Use the **cable modem qos dsc** command in Privileged EXEC mode to change an existing service flow, as shown below:

   `MOT#`**cable modem** {<*mac*> | <*prefix*>} **qos dsc** {<*prefix*> <*string*>}

   where:

   mac is the CM MAC address.

   prefix is the CM IP address.

   prefix is the TFTP server IP address.

   string is the CM configuration file name.

3. To display service flow statistics, use the **show cable qos svc-flow statistics** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show cable qos svc-flow statistics {<x>/<y>} {<1-4292967295> | <cr>}
   ```

   where:

   x is the slot number of the cable module.

   y is the cable interface number, which is **0**.

   1-4292967295 is the Service Flow Identifier (SFID) number.

   cr is a command return, which displays QoS service flow statistics for all SFIDs.

4. To display statistics for both upstream and downstream DSC messages, use the **show cable qos svc-flow dynamic stat** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show cable qos svc-flow dynamic stat
   ```

Initiating a DSD

Use DSD messages to delete an existing service flow. Follow these steps to delete a specific service flow:

Note: Before you begin, ensure that the correct CM SFID is selected.

1. To display the SFID of all the service flows used by a specific CM, use the **show cable modem svc-flow-id** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show cable modem [<mac> | <prefix>] svc-flow-id
   ```

   where:

   mac is the CM MAC address.

   prefix is the CM IP address.

2. To initiate the DSD of a specific SFID, use the **cable modem qos dsd** command in Privileged EXEC mode, as shown below:

   `MOT#`**cable modem qos dsd** {<*x*>/<*y*> <*1-262143*>}

   where:

   x is the slot number of the cable module.

   y is the cable interface number; which is **0**.

   1-262143 is the Service Flow Identifier (SFID).

3. To display the deleted service flow log, use the **show cable qos svc-flow log** command in Privileged EXEC mode, as shown below:

   `MOT#`**show cable qos svc-flow log**

4. To display statistics for both upstream and downstream DSD messages, use the **show cable qos svc-flow dynamic stat** command in Privileged EXEC mode, as shown below:

   `MOT#`**show cable qos svc-flow dynamic stat**

Viewing QoS Information

Use the following sections to obtain QoS information:

- Displaying the Packet Classifier
- Displaying SFID and QoS Information
- Displaying Service Flow Statistics
- Displaying Payload Header Suppression Entries

Displaying the Packet Classifier

A service flow classifier matches a packet to a service flow using a service flow reference. The service flow reference associates a packet classifier encoding with a service flow encoding to establish a SFID. Classifiers have the following features:

- Classifiers are loosely ordered by priority.
- Several classifiers can refer to the same service flow.
- More than one classifier may have the same priority.

- The cable interface uses a downstream classifier to assign packets to downstream service flows.
- The CM uses an upstream classifier to assign packets to upstream service flows.

To display the packet classifiers of a service flow configured on the cable interface, use the **show cable qos svc-flow classifier** command in Privileged EXEC mode, as shown below:

```
MOT#show cable qos svc-flow classifier {<x>/<y>} [<1-4292967295> | <cr>]
[<1-65535> | <cr>]
```

where:

x is the slot number of the cable module.

y is the cable interface number, which is **0**.

1-4292967295 is the Service Flow Identifier (SFID).

cr is a command return, which displays QoS service flow statistics for all SFIDs.

classifier-id is the classifier identifier..

Note: If the Classifier ID is not given, all the classifiers with the given SFID are listed.

Displaying SFID and QoS Information

Use the **show cable qos svc-flow summary** command in Privileged EXEC mode to display service flow identifier (SFID) and QoS parameter information, as shown below:

`MOT#`**show cable qos svc-flow summary** {<*x*>/<*y*>} [<*1-4292967295*> | <*cr*>]

where:

x is the slot number of the cable module.

y is the cable interface number, which is **0**.

1-4292967295 is the Service Flow Identifier (SFID).

cr is a command return, which displays QoS service flow statistics for all SFIDs.

Displaying Service Flow Statistics

Use the **show cable qos svc-flow statistics** command in Privileged EXEC mode to display service flow statistics, as shown below:

`MOT#`**show cable qos svc-flow statistics** {<*x*>/<*y*>} [<*1-4292967295*> | <*cr*>]

where:

x is the slot number of the cable module.

y is the cable interface number, which is **0**.

1-4292967295 is the Service Flow Identifier (SFID).

cr is a command return, which displays QoS service flow statistics for all SFIDs.

Displaying Payload Header Suppression Entries

Use the **show cable qos svc-flow phs** command in Privileged EXEC mode to display the Payload Header Suppression (PHS) entries for a service flow, as shown below:

MOT#**show cable qos svc-flow phs** {<*x*>/<*y*>} [<*1-4292967295*> | <cr>] [*1-65535*> | <cr>]

where:

x is the slot number of the cable module.

y is the cable interface number, which is **0**.

1-4292967295 is the Service Flow Identifier (SFID).

cr is a command return, which displays QoS service flow statistics for all SFIDs.

1-65535 is the classifier identifier.

Note: If the Classifier ID is not given, all the classifiers with the given SFID are listed.

Implementing Spectrum Management

The spectrum management system monitors the upstream signal integrity, and collects upstream spectrum information. When signal integrity degrades due to noise, the spectrum management system automatically configures the upstream channel parameters to maintain low packet loss by changing the upstream frequency, modulation profile, channel-width and/or power level to ensure that upstream performance stays at acceptable levels.

The following tasks are used to implement frequency agility on the BSR:

- Configuring a Spectrum Group
- Applying a Spectrum Group to an Upstream Port
- Evaluating Spectrum Management Performance

Configuring a Spectrum Group

Each spectrum group contains spectrum data, a spectrum map, and channel assignment:

- The spectrum data is where the collected spectrum noise information is kept. It contains the starting frequency, resolution, number of data points, time of the last measurement, and a pointer to an array where the noise level is kept.
- The spectrum map describes the way the upstream spectrum is used for a particular band. It contains the start and stop frequency, and the current status.
- The channel assignment defines the frequency allocation of the upstream channel.

Note: The term "upstream channel" is synonymous to the term "upstream port".

Each spectrum group also contains the following management information:

- Spectrum signal quality information is collected through the spectrum monitoring process. This information contains the periodic error rate that is computed and compared with the Forward Error Correction (FEC) error threshold to determine if spectrum hopping is necessary, and the periodic combination of in-band power and spectrum data collection to compute signal-to-noise ratio (SNR) for the upstream channel.
- The spectrum schedule contains information on the availability of a certain frequency band. The band can be made available statically, or available only at certain time period(s). The actual available spectrum is the super set of all the bands that are available at the time.
- Spectrum hopping rules determine the action taken when the spectrum manager decides to change the parameters of an upstream channel to combat noise. Operators can improve upstream channel conditions to combat ingress noise by specifying hopping rules for upstream frequency changes, upstream frequency band changes, modulation profile changes, channel-width reduction (until channel conditions improve), and power adjustments (if it is necessary).

Defining a spectrum group contains the following tasks outlined in this section:

- Creating a Spectrum Group
- Scheduling the Availability of a Spectrum Group Band
- Scheduling the Removal of a Spectrum Group Band
- Configuring Spectrum Data Collection
- Configuring Spectrum Hopping Rules
- Configuring the Spectrum Hopping Error Threshold
- Configuring the Spectrum Hopping Flap Threshold
- Enabling and Disabling Spectrum Roll-back
- Configuring the Guard Band
- Reviewing the Spectrum Group that You Created

Creating a Spectrum Group

Follow these steps to create a cable spectrum group:

1. To create a cable spectrum group and enter the new mode in which to configure your cable spectrum group, use the **cable spectrum** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#cable spectrum <WORD>
   ```

   where:

   WORD is the spectrum group name.

Note: No spaces are allowed for the spectrum group name.

The Cable Spectrum Group mode displays. From this new prompt, all of the cable spectrum parameters are configured. For example, if you defined your group name as *spectrum1* the prompt would display as shown below:

```
MOT(config-spcgrp:spectrum1)#
```

- If you need to delete a spectrum group, use the **no cable spectrum** command, in Global Configuration or Cable Spectrum Group mode as shown below:

  ```
  MOT(config)#no cable spectrum <WORD>
  ```

  ```
  MOT(config-spcgrp:<WORD>)#no cable spectrum <WORD>
  ```

- If you need to change to another spectrum group or wish to create a new spectrum group, use the **cable spectrum** command in Cable Spectrum Group mode as shown below:

  ```
  MOT(config-spcgrp:spectrum1)#cable spectrum <WORD>
  ```

  where:

  WORD is the name of a new spectrum group or another spectrum group.

  For example:

  ```
  MOT(config-spcgrp:spectrum1)#cable spectrum2
  ```

  ```
  MOT(config-spcgrp:spectrum2)#
  ```

2. Use the **band** command in Cable Spectrum Group mode to define the start and end frequency band for the spectrum group, as shown below:

   ```
   MOT(config-spcgrp:<WORD>)#band {<start frequency> <end frequency>}
   ```

   where:

   WORD is the spectrum group name that you defined.

   start frequency is the start upstream frequency from 5000000 to 42000000 Hertz.

   end frequency is the end upstream frequency from 5000000 to 42000000 Hertz.

For example, if you defined your spectrum1 group to have a start frequency of 8 MHz and an end frequency of 12 MHz your command syntax would look as shown below:

```
MOT(config-spcgrp:spectrum1)#band 8000000 12000000
```

- If you need to add another start and end frequency band to the spectrum group, repeat this step.
- If you need to delete a start and end frequency band from a spectrum group, use the **no band** command as shown below:

  ```
  MOT(config-spcgrp:<WORD>)#no band {<start frequency> <end frequency>}
  ```

Scheduling the Availability of a Spectrum Group Band

The **time band** command is used to schedule when a spectrum group band is available. The spectrum group band can be made available on either a daily or weekly schedule.

Follow these steps to schedule the availability of a spectrum group band:

1. Use the **cable spectrum** command in Global Configuration mode to enter Cable Spectrum Group mode, as shown below:

Note: If a new availability time for a band is entered for a spectrum group, the existing availability time for a band must be deleted first.

```
MOT(config)#cable spectrum <WORD>
```

where:

WORD is the spectrum group name.

The Cable Spectrum Group mode displays for the spectrum group.

2. If you want to schedule the time for when the spectrum group band becomes available on a daily basis, use the **time band** command in Cable Spectrum Group mode, as shown below:

MOT(config-spcgrp:<*WORD*>)#**time** <*hh:mm:ss*> **band** {<*start frequency*> <*end frequency*>}

where:

WORD is the spectrum group name.

hh:mm:ss is the time of the day, which includes the hour, minute, and second when the band becomes available.

start frequency is the start upstream frequency from 5000000 to 42000000 Hertz.

end frequency is the end upstream frequency from 5000000 to 42000000 Hertz.

For example:

The following example defines the 25 MHz to 35 MHz upstream frequency band as being available daily at 4:00 PM for spectrum group *spectrum1*:

```
MOT(config-spcgrp:spectrum1)#time 16:00:00 band 25000000 35000000
```

3. If you want to schedule the time for when the spectrum group band becomes available on a weekly basis, use the **time band** command in Cable Spectrum Group mode, as shown below:

   ```
   MOT(config-spcgrp:<WORD>)#time {<day> <hh:mm:ss>} band {<start frequency> <end frequency>}
   ```

   where:

   WORD is the spectrum group name.

   day is the three letter abbreviation for day of the week.

   hh:mm:ss is the time during the day when the band becomes available.

   start frequency is the start upstream frequency from 5000000 to 42000000 Hertz.

   end frequency is the end upstream frequency from 5000000 to 42000000 Hertz.

   The following example defines the 21 MHz to 29 MHz upstream frequency band as being available every Thursday morning at 10:00 AM for spectrum group *spectrum1*:

   ```
   MOT(config-spcgrp:spectrum1)#time Thu 10:00:00 band 21000000 29000000
   ```

Deleting an Existing Availability Time for a Band

If you need to delete the existing availability time for a band, use the **no time band** command in Cable Spectrum Group mode, as shown below:

```
MOT(config-spcgrp:<WORD>)#no time {<day> | <hh:mm:ss>} band {<start frequency> <end frequency>}
```

Note: When deleting the time for a band, ensure that the exact day, hh:mm:ss, and start and end upstream frequencies are used.

where:

WORD is the spectrum group name.

day is the three letter abbreviation for day of the week

hh:mm:ss is the time during the day when the band is removed.

start frequency is the start upstream frequency from 5000000 to 42000000 Hertz.

end frequency is the end upstream frequency from 5000000 to 42000000 Hertz.

Scheduling the Removal of a Spectrum Group Band

Follow these steps to schedule the removal of a spectrum group band:

1. Use the **cable spectrum** command in Global Configuration mode to enter Cable Spectrum Group mode, as shown below:

Note: If a new removal time for a band is entered for a spectrum group, the existing removal time for a band must be deleted first.

```
MOT(config)#cable spectrum <WORD>
```

where:

WORD is the spectrum group name.

The Cable Spectrum Group mode displays for the spectrum group.

2. If you want to schedule the time when the spectrum group band is removed on a daily basis, use the **time delete band** command in Cable Spectrum Group mode, as shown below:

```
MOT(config-spcgrp:<WORD>)#time <hh:mm:ss> delete band {<start frequency> <end frequency>}
```

where:

WORD is the spectrum group name.

hh:mm:ss is the time during the day when the band is removed.

start frequency is the start upstream frequency from 5000000 to 42000000 Hertz.

end frequency is the end upstream frequency from 5000000 to 42000000 Hertz.

The following example determines that the band from 25 MHz to 35 MHz, belonging to spectrum group *spectrum1*, is removed every day at 20:00 PM:

```
MOT(config-spcgrp:spectrum1)#time 20:00:00 delete band 25000000 35000000
```

The following example determines that the band from 21 MHz to 29 MHz, belonging to spectrum group *spectrum1*, is removed every Thursday morning at 11:00 AM:

`MOT(config-spcgrp:spectrum1)#`**time Thu 11:00:00 delete band 21000000 29000000**

3. If you want to schedule the time when the spectrum group band is removed on a weekly basis, use the **time delete band** command in Cable Spectrum Group mode, as shown below:

   `MOT(config-spcgrp:`<*WORD*>`)#`**time** {<*day*> <*hh:mm:ss*>} **delete band** {<*start frequency*> <*end frequency*>}

   where:

   WORD is the spectrum group name.

   day is the three letter abbreviation for day of the week

   hh:mm:ss is the time during the day when the band is removed.

   start frequency is the start upstream frequency from 5000000 to 42000000 Hertz.

   end frequency is the end upstream frequency from 5000000 to 42000000 Hertz.

   For example, the following syntax is used to express that the band from 21 MHz to 29 MHz, belonging to spectrum group *spectrum1*, is removed every Thursday morning at 11:00 AM:

   `MOT(config-spcgrp:spectrum1)#`**time Thursday 11:00:00 delete band 21000000 29000000**

Deleting an Existing Removal Time for a Band

If you need to delete the existing removal time for a band, use the **no time delete band** command in Cable Spectrum Group mode, as shown below:

`MOT(config-spcgrp:`*<WORD>*`)#`**no time** {*<day>* *<hh:mm:ss>*} **delete band** {*<start frequency>* *<end frequency>*}

Note: Ensure that the exact parameters for the removal of a time band are entered in order for the change to occur.

where:

WORD is the spectrum group name.

day is the three letter abbreviation for day of the week.

hh:mm:ss is the time during the day when the band is removed.

start frequency is the start upstream frequency from 5000000 to 42000000 Hertz.

end frequency is the end upstream frequency from 5000000 to 42000000 Hertz.

Configuring Spectrum Data Collection

The spectrum data collection feature can be used to take a ’snapshot’ or view of all or a portion of the upstream spectrum between 5 and 42 MHz to help determine the best places on the upstream spectrum to configure the hop action frequency and band values. The spectrum data collection feature is also used to analyze and troubleshoot the upstream channel when a user does not have access to a spectrum analyzer to view noise level and signal information. The spectrum data collection feature is disabled by default and does not need to be enabled in order for spectrum management to work.

The spectrum data collection task is split between the spectrum manager and the spectrum agent. The spectrum manager schedules the data collection with the spectrum agent, and provides data storage for the collected data, while the spectrum agent performs the actual data collection and sends the collected data to the spectrum manager.

Note: Spectrum data is not collected for an upstream channel until the correct collection interval is configured for the spectrum group, and the spectrum group is applied to the upstream port.

Follow these options to change the default spectrum data collection parameters used by the spectrum manager:

- The default resolution is 200000 Hertz (Hz). Use the **collect resolution** command in Cable Spectrum Group mode to change the frequency resolution rate that the spectrum manager performs, as shown below:

  ```
  MOT(config-spcgrp:<WORD>)#collect resolution <200000-4000000>
  ```

  where:

  WORD is the spectrum group name.

  200000-4000000 is the resolution in Hz.

- The default collection interval is 0, which indicates that no collection interval is defined. Use the **collect interval** command in Cable Spectrum Group mode to configure the interval rate at which data collection is performed by the spectrum manager while it scans the entire spectrum map from 5 MHz to 42 MHz, as shown below:

  ```
  MOT(config-spcgrp:<WORD>)#collect interval <60-65535>
  ```

  where:

  WORD is the spectrum group name.

  60-65535 is the time interval expressed in seconds.

For example, the following syntax shows that the spectrum manager's data collection interval rate for scanning the spectrum map occurs once every hour:

```
MOT(config-spcgrp:spectrum1)#collect interval 3600
```

- Since the spectrum collection feature uses resources on the upstream channel that may affect throughput for CMs associated with the upstream port, the spectrum data collection feature should be turned off when it is no longer in use to conserve network resources.
  - Use the **no collect interval** command to disable the collection interval.
  - Use the **no collect resolution** command to disable the frequency resolution.

Configuring Spectrum Hopping Rules

Rules for spectrum hopping must be defined before the spectrum hopping function is used. Spectrum hopping rules are searched before spectrum hopping occurs on an upstream port when the spectrum group is triggered. Spectrum hopping rules are used by the spectrum manager to find the best way to defeat noise problems on an upstream port.

The following spectrum hopping rules apply:

- No actions are taken if spectrum hopping rules are not defined. The rules include the preferred frequency, modulation profile, channel-width parameters, and power adjustment.
- Multiple hopping rules with same type of action are allowed.
- Each hopping rule can be assigned with different priorities.
- Hopping rules are applied by priority and hopping rules with same priority are applied in the order in which they are entered.
- If each individual rule fails to apply, the spectrum manager attempts to apply different combinations of the individual rules.

Follow these steps to configure spectrum hopping rules:

1. The default hop period is 300 seconds. Use the **hop period** command in Cable Spectrum Group mode to prevent excessive frequency hops on an upstream port, as shown below:

   `MOT(config-spcgrp:`<*WORD*>`)#`**hop period** <*num:30-3600*>

   where:

   WORD is the spectrum group name.

   num:30-3600 is the rate at which the frequency hop takes place, expressed in seconds.

2. Use the **hop action frequency** command in Cable Spectrum Group mode to determine the frequency hop for discrete center frequencies during the frequency hop action, as shown below:

   `MOT(config-spcgrp:`<*WORD*>`)#`**hop action frequency** <*center frequency*> [**priority** <*n*>]

   where:

   WORD is the spectrum group name.

   center frequency is the upstream frequency from 5000000 to 42000000 Hz

   1-255 is the priority number of the upstream frequency hop action. When no priority is assigned, the default priority is 128. The lower number takes precedence.

   For example, the following syntax determines that 28 MHz replaces the existing upstream frequency when a hop action is triggered and defines the priority level of the hop:

   `MOT(config-spcgrp:`spectrum1`)#`**hop action frequency 28000000 priority 30**

3. Use the **hop action modulation-profile** command in Cable Spectrum Group mode to change the modulation profile setting for a hop action, as shown below:

Note: Refer to the Creating a Modulation Profile section for more information on configuring modulation profiles.

`MOT(config-spcgrp:`<*WORD*>`)#`**hop action modulation-profile** {<*1-16*>} [**priority** {<*1-255*>}]

where:

WORD is the spectrum group name.

1-16 is the modulation profile number. The default modulation profiles are 1 and 2.

1-255 is the priority number of the upstream modulation profile hop action. When no priority is assigned, the default priority is 128.

For example, the following syntax determines that modulation profile 2 replaces the existing modulation profile when the hop action is triggered and defines the priority level of the hop:

`MOT(config-spcgrp:`spectrum1`)#`**hop action modulation-profile 2 priority 50**

4. Use the **hop action channel-width** command in Cable Spectrum Group mode to change the upstream channel-width setting in Hertz (Hz) for a hop action, as shown below:

Note: Refer to the Configuring Upstream CM Registration Parameters section for more information on setting the upstream channel width.

```
MOT(config-spcgrp:<WORD>)#hop action channel-width [1600000 |
200000 | 3200000 | 400000 | 800000] [priority <1-255>]
```

where:

WORD is the spectrum group name.

1-255 is the priority number of the upstream channel width setting. When no priority is assigned, the default priority is 128.

For example, the following syntax determines that the upstream channel width of 1.6 MHz replaces the existing upstream channel width when the hop action is triggered and defines the priority level of the hop:

```
MOT(config-spcgrp:spectrum1)#hop action channel-width 1600000
priority 100
```

5. Use the **hop action band** command in Cable Spectrum Group mode to determine the hop for each frequency band during the frequency hop action, as shown below:

```
MOT(config-spcgrp:<WORD>)#hop action band {<start frequency> <end frequency>} [priority <1-255>]
```

where:

WORD is the spectrum group name.

start frequency is the start upstream frequency band from 5000000 to 42000000 Hz.

end frequency is the end upstream frequency band from 5000000 to 42000000 Hz.

1-255 is the priority number of the upstream band hop action. When no priority is assigned, the default priority is 128.

For example, the following syntax determines that the upstream frequency band of 20 MHz to 30 MHz replaces the existing upstream frequency band when the hop action is triggered and defines the priority level of the hop:

```
MOT(config-spcgrp:spectrum1)#hop action band 20000000 30000000 priority 110
```

6. Use the **hop action power-level** command in Cable Spectrum Group mode to change the power-level setting for a hop action, as shown below:

Note: Refer to the Setting the Upstream Power Level section for more information on setting the upstream power level parameters for relative and absolute mode.

```
MOT(config-spcgrp:<WORD>)#hop action power-level {<default>}
[priority <1-255>]
```

where:

WORD is the spectrum group name.

power is the input power level, expressed in dB.

default is the number of dB above or below the default input power level. Refer to Table 6-7 for more information on this setting.

1-255 is the priority number of the upstream power level hop action. When no priority is assigned, the default priority is 128.

Table 6-7 describes the input power level parameters expressed in dB:

Table 6-7Relative Input Power Levels

Power Range	Upstream Channel Width
-160 to +140 dB	200 kHz
-130 to +170 dB	400 kHz
-100 to +200 dB	800 kHz
-70 to +230 dB	1600 kHz
-40 to +260 dB	3200 kHz

Configuring the Spectrum Hopping Error Threshold

A frequency hopping error threshold is configured as a criteria to apply the hopping rules in instances when an unacceptable error rate occurs, which is caused possibly by the poor signal quality of channel.

The hopping threshold error rate is determined by the Forward Error Correction (FEC) error-rate threshold value. If the error-rate threshold is configured, the spectrum manager periodically polls the signal quality table of the member channels to compute the error rate during the polling interval. If the error rate exceeds the threshold, it triggers spectrum hopping for the affected channel. The error rate is a fraction of 1000.

The default hopping threshold error rate is 10 or 1 percent. Use the **hop threshold error** command in Cable Spectrum Group mode to adjust the acceptable hopping threshold error rate, as shown below:

`MOT(config-spcgrp:`<*WORD*>`)#`**hop threshold error** {<*num:1-1000*>}

where:

WORD is the spectrum group name.

num:1-1000 is the error rate as a fraction of 1000.

For example, an error rate of 1 implies 0.1 percent or an error rate of 1000 implies 100 percent.

Configuring the Spectrum Hopping Flap Threshold

A frequency hopping flap threshold is configured as a criteria to apply the hopping rules in instances when one or a minimal number of cable modems (CMs) loose their connection with the BSR (Flap).

The frequency hopping flap threshold is determined by the percentage of CMs that loose their connectivity. If the flap threshold is configured, the spectrum manager periodically scans the flap-list table of the member channels to compute the flap rate during the scan interval. If the flap rate exceeds the threshold, it triggers spectrum hopping for the affected channel.

This frequency hopping threshold is activated with a value of 0 percent by default to prevent the unnecessary triggering of a hopping action. For example, if the downstream cable is disconnected or the downstream frequency is changed, these actions would cause all CMs on the network to flap.

Use the **hop threshold flap** command in Cable Spectrum Group mode to set a value that triggers when a greater than a set percentage of CMs loose their connectivity, as shown below:

`MOT(config-spcgrp:`<*WORD*>`)#`**hop threshold flap** {<*num:1-100*>}

where:

> *WORD* is the spectrum group name.
>
> *num:1-100* is the percentage of CMs from 1 to 100 that loose connectivity on the network.

If an existing percentage other than zero is set, and you need to take some action that causes CMs to flap (such as changing the downstream frequency), use the **no hop threshold flap** command in Cable Spectrum Group mode to deactivate the frequency hopping threshold before taking the action, as shown below:

`MOT(config-spcgrp:`<*WORD*>`)#`**no hop threshold flap** <*num:1-100*>

where:

> *WORD* is the spectrum group name.
>
> *num:1-100* is the set percentage of CMs that loose their connectivity on the network.

Enabling and Disabling Spectrum Roll-back

The spectrum roll-back function is disabled by default and is used to return the upstream channel width or modulation profile setting, that was adjusted during a hop action, to the original configuration when upstream channel conditions improve.

To enable the spectrum roll-back function, use the hop action rollback command in Cable Spectrum Group mode as shown below:

`MOT(config-spcgrp:`<*WORD*>`)#`**hop action rollback**

where:

> *WORD* is the spectrum group name.

To disable the spectrum roll-back function, use the **no hop action rollback** command.

Configuring the Guard Band

Use the **guard-band** command in Cable Spectrum Group mode to set the minimum spectrum separation or spacing between upstream channels in the same spectrum group.

```
MOT(config-spcgrp:<WORD>)#guard-band <0-37000000>
```

where:

WORD is the spectrum group name.

0-37000000 is the guard band separation size expressed in Hertz (Hz) for DOCSIS. The default guard band is 0 Hz.

0-60000000 is the guard band separation size expressed in Hertz (Hz) for Euro-DOCSIS. The default guard band is 0 Hz.

Reviewing the Spectrum Group that You Created

To view the spectrum group that you created, enter the **show cable spectrum-group** command in Global Configuration mode, as shown below:

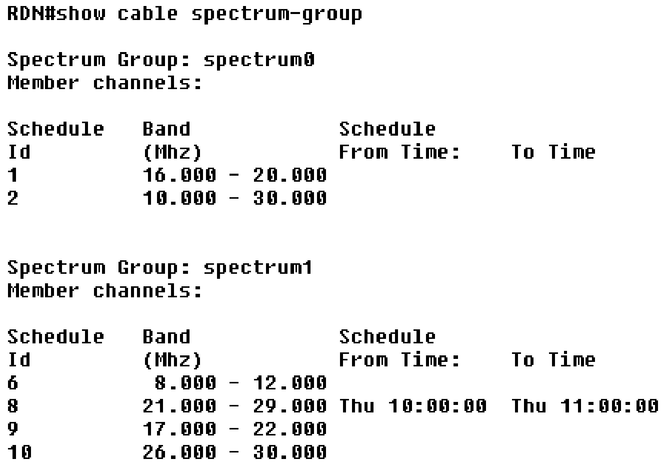
```
MOT(config)#show cable spectrum-group {<WORD> | <cr>}
```

where:

WORD is the spectrum group name.

cr is a command return that displays all configured spectrum groups.

Figure 6-1 shows sample output for the **show cable spectrum-group** command:



```
RDN#show cable spectrum-group

Spectrum Group: spectrum0
Member channels:

Schedule   Band            Schedule
Id         (Mhz)           From Time:    To Time
1          16.000 - 20.000
2          10.000 - 30.000


Spectrum Group: spectrum1
Member channels:

Schedule   Band            Schedule
Id         (Mhz)           From Time:    To Time
6           8.000 - 12.000
8          21.000 - 29.000 Thu 10:00:00  Thu 11:00:00
9          17.000 - 22.000
10         26.000 - 30.000
```

Figure 6-1 show cable spectrum-group Command Output

To view the spectrum allocation map for the spectrum group that you created, enter the **show cable spectrum-group map** command in Global Configuration mode, as shown below:

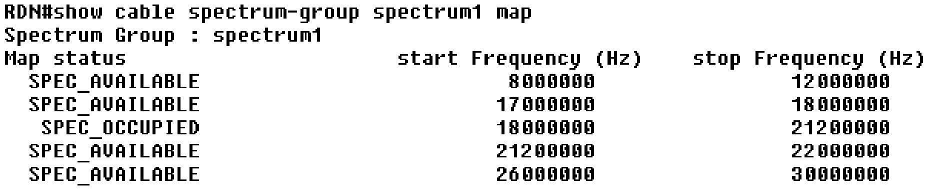
`MOT(config)#`**show cable spectrum-group** <*WORD*> **map**

where:

WORD is the spectrum group name.

Note: In the **show cable spectrum-group map** command output, the SPEC_OCCUPIED message that appears in the Map status column indicates that this section of the upstream spectrum is occupied by the upstream channel of the spectrum group. The SPEC_AVAILABLE message indicates that the section of the upstream spectrum is free to use, and no upstream channel is currently using this section of the upstream spectrum.

Figure 6-2 shows sample output for the **show cable spectrum-group map** command:



```
RDN#show cable spectrum-group spectrum1 map
Spectrum Group : spectrum1
Map status                        start Frequency (Hz)    stop Frequency (Hz)
  SPEC_AVAILABLE                           8000000                12000000
  SPEC_AVAILABLE                          17000000                18000000
   SPEC_OCCUPIED                          18000000                21200000
  SPEC_AVAILABLE                          21200000                22000000
  SPEC_AVAILABLE                          26000000                30000000
```

Figure 6-2 show cable spectrum-group map Command Output

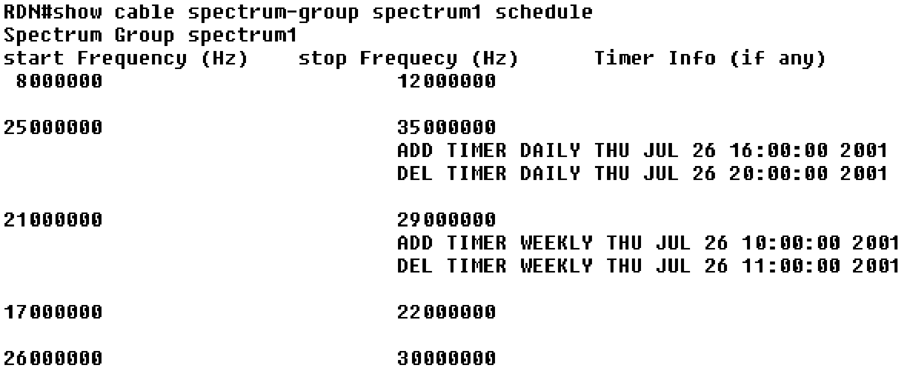
To view the spectrum schedule for the spectrum group that you created, enter the **show cable spectrum-group schedule** command in Global Configuration mode, as shown below:

`MOT(config)#`**show cable spectrum-group** <*WORD*> **schedule**

where:

WORD is the spectrum group name.

Figure 6-3 shows sample output for the **show cable spectrum-group schedule** command:



```
RDN#show cable spectrum-group spectrum1 schedule
Spectrum Group spectrum1
start Frequency (Hz)    stop Frequecy (Hz)      Timer Info (if any)
 8000000                        12000000

25000000                        35000000
                                ADD TIMER DAILY THU JUL 26 16:00:00 2001
                                DEL TIMER DAILY THU JUL 26 20:00:00 2001

21000000                        29000000
                                ADD TIMER WEEKLY THU JUL 26 10:00:00 2001
                                DEL TIMER WEEKLY THU JUL 26 11:00:00 2001

17000000                        22000000

26000000                        30000000
```

Figure 6-3 show cable spectrum-group schedule Command Output

Viewing Your Spectrum Group Configuration

The **show running config** command does not show the configured parameters if the spectrum manager makes changes the upstream frequency, channel width, modulation or power level by hopping action. However, you can use the **show running-config** command in Privileged EXEC mode to view the configuration of a spectrum group that you created, as shown below:

```
MOT#show running-config
```

For example, the following **show running-config** command output shows the configured spectrum group information:

```
cable spectrum-group spectrum1
 time 16:00:00 band 25000000 35000000
 time 20:00:00 delete band 25000000 35000000
 time Thus 10:00:00 band 21000000 29000000
 time Thus 11:00:00 delete band 21000000 29000000
 band 17000000 22000000
 band 26000000 30000000
 collect interval 3600
 hop action frequency 28000000 priority 30
 hop action modulation-profile 2 priority 50
 hop action channel-width 1600000 priority 100
 hop action band 20000000 30000000 priority 110
 hop action roll-back
```

Applying a Spectrum Group to an Upstream Port

When a spectrum group is applied to an upstream port, the upstream port belongs to the spectrum group.

Note: The spectrum manager is unaware of the physical topology of your cable plant, and it is only aware of the spectrum group. Ensure that you apply the same spectrum group to all upstream ports that share the same upstream frequency assignment on the same physical cable plant. Also ensure that the frequency configurations of different spectrum groups do not overlap if they share the same physical plant.

Follow these steps to assign a spectrum group to an upstream port on a cable interface:

1. To enter the cable interface, use the **interface cable** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#interface cable <x>/<y>
   ```

   where:

   x is the slot number of the master cable module.

   y is the cable interface number, which is **0**.

2. To apply a spectrum group to an upstream port, use the **cable upstream spectrum-group** command in Interface Configuration mode, as shown below:

Note: All upstream ports sharing the same return path must be configured to the same spectrum group.

   ```
   MOT(config-if)#cable upstream <NUM> spectrum-group <WORD>
   ```

   where:

   NUM is the upstream port number.

   WORD is the exact spectrum group name applied to the upstream port.

3. To verify if the spectrum group that you assigned is activated for the upstream port, enter the **show cable spectrum-group** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#show cable spectrum-group [<WORD> | <cr>]
   ```

   where:

   WORD is the exact group name applied to the upstream port.

   cr is a command return that displays all spectrum-groups on the cable interface.

4. If you want to see what spectrum group is applied to each upstream port, use the **show running config** command in Privileged EXEC mode, as shown below:

   MOT#**show running-config**

Evaluating Spectrum Management Performance

Use the information and examples contained in the following sections to evaluate your spectrum management configuration and performance:

- Showing Spectrum Data
- Viewing Spectrum Management Configuration Changes
- Determining the Upstream Signal to Noise Ratio
- Determining the MIB Index ID Number of an Upstream Port
- Viewing Spectrum Management Activity
- Viewing Spectrum Management Hopping Actions
- Viewing the Spectrum Management Roll-back Function

Showing Spectrum Data

Use the **show interface cable upstream spectrum** command in Privileged EXEC mode to view the noise power level for the whole spectrum, as shown below:

MOT#**show interface cable** <*x*>/<*y*> **upstream** <*NUM*> **spectrum**

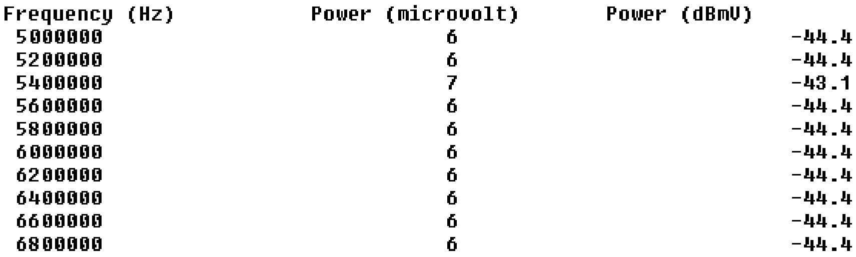
where:

x is the slot number of the cable module.

y is the cable interface number, which is **0**.

NUM is the upstream port number

Figure 6-4 displays the **show interface cable upstream spectrum** command output:



```
Frequency (Hz)              Power (microvolt)         Power (dBmV)
 5000000                                  6                           -44.4
 5200000                                  6                           -44.4
 5400000                                  7                           -43.1
 5600000                                  6                           -44.4
 5800000                                  6                           -44.4
 6000000                                  6                           -44.4
 6200000                                  6                           -44.4
 6400000                                  6                           -44.4
 6600000                                  6                           -44.4
 6800000                                  6                           -44.4
```

Figure 6-4 show interface cable upstream spectrum Command Output

Viewing Spectrum Management Configuration Changes

Follow these steps to view upstream information when the spectrum manager makes changes to an upstream port:

1. Use the **interface cable** command to enter the desired cable interface, as shown below:

   `MOT(config)#`**interface cable** <*x*>/<*y*>

   where:

   x is the slot number of the cable module.

   y is the cable interface number, which is **0**.

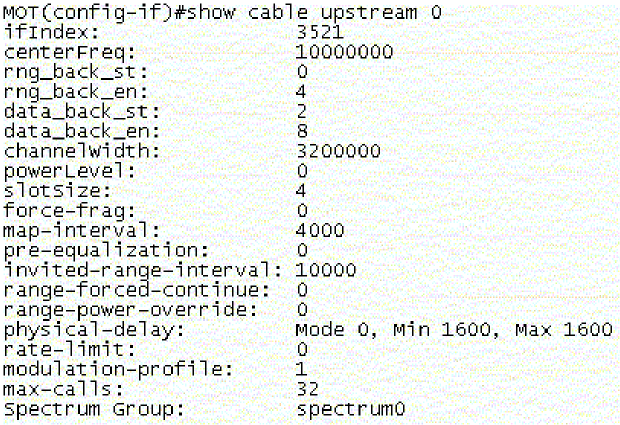
2. Use the **show cable upstream** command to look the current frequency, channel width, modulation or power level for the upstream port:

   `MOT(config-if)#`**show cable upstream** <*NUM*>

   where:

   NUM is the upstream port number.

For example, Figure 6-5 displays upstream port statistics:



```
MOT(config-if)#show cable upstream 0
ifIndex:                   3521
centerFreq:                10000000
rng_back_st:               0
rng_back_en:               4
data_back_st:              2
data_back_en:              8
channelWidth:              3200000
powerLevel:                0
slotSize:                  4
force-frag:                0
map-interval:              4000
pre-equalization:          0
invited-range-interval:    10000
range-forced-continue:     0
range-power-override:      0
physical-delay:            Mode 0, Min 1600, Max 1600
rate-limit:                0
modulation-profile:        1
max-calls:                 32
Spectrum Group:            spectrum0
```

Figure 6-5 show cable upstream Command Output

Determining the Upstream Signal to Noise Ratio

Use the **show interfaces cable upstream signal-quality** command in Privileged EXEC mode to determine the signal power to noise ratio, and error signal quality information as shown below:

`MOT#`**show interfaces cable** <*x*>/<*y*> **upstream** <*NUM*> **signal-quality**

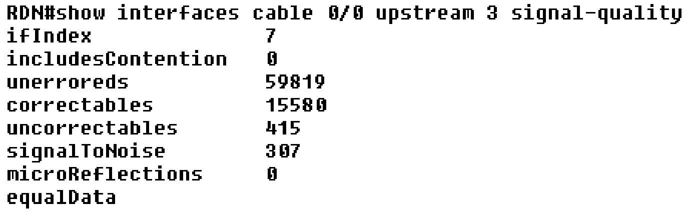
where:

x is the slot number of the cable module.

y is the cable interface number, which is **0**.

NUM is the upstream port number.

For example, Figure 6-6 displays a typical signal-quality result for an upstream port:



```
RDN#show interfaces cable 0/0 upstream 3 signal-quality
ifIndex                 7
includesContention      0
unerroreds              59819
correctables            15580
uncorrectables          415
signalToNoise           307
microReflections        0
equalData
```

Figure 6-6 show interfaces cable upstream signal-quality Command Output

Determining the MIB Index ID Number of an Upstream Port

It is important to learn the MIB Index ID number that is associated to a specific upstream port on a DOCSIS module because the **debug specmgr** and **logging console notifications** command log output only displays the MIB Index ID number.

Follow these steps to determine the MIB Index ID number of an upstream port:

1. Use the **interface cable** command to enter the desired cable interface, as shown below:

   `MOT(config)#`**interface cable** <*x*>/<*y*>

   where:

   x is the slot number of the cable module.

   interface is the cable interface number, which is **0**.

2. Use the **show cable upstream** command to discover what MIB Index ID number is associated with a upstream port on a particular module as shown below:

   `MOT(config-if)#`**show cable upstream** <*NUM*>

   where:

   NUM is the upstream port number.

The following example shows that upstream port 1 on module 0 has a MIB Index ID number (ifIndex) of 5. This number is used to determine the slot and upstream port number that is displayed in the debug specmgr and console logging output.

```
MOT(config)#interface cable 0/0
MOT(config-if)#show cable upstream 1
ifIndex:                5
centerFreq:             13200000
rng_back_st:            0
rng_back_en:            4
data_back_st:           2
data_back_en:           8
channelWidth:           3200000
powerLevel:             100
slotSize:               4
force-frag:             0
map-interval:           4000
pre-equalization:       0
invited-range-interval: 10000
range-forced-continue:  0
range-power-override:   0
physical-delay:         Mode 0, Min 1600, Max 1600
rate-limit:             0
modulation-profile:     2
Spectrum Group:         spectrum_1
```

Viewing Spectrum Management Activity

The **logging console notifications** command can be used to monitor spectrum management activity whenever the frequency, channel width, modulation profile, power level changed manually or changed by the Spectrum Manager, the notification message is displayed.

Use the **logging console notifications** command in Global Configuration mode to turn on console logging and view manual changes or changes made by the spectrum manager, as shown below:

```
MOT(config)#logging console notifications
```

For example, if the upstream frequency was changed to 20000000 Hertz, the following notification output appears:

```
[07/23-10:57:17:SPECMGR]-N-Set to new frequency 20000000 for
channel ifIndex = 4 .
[07/23-10:57:17:console]-N-configuration change by
[enabled-user]: cable upstream 0 frequency 20000000
```

Viewing Spectrum Management Hopping Actions

The **debug specmgr** command is used to monitor all active upstream ports. The **debug specmgr** command output in this section describes what can happen when spectrum management hopping actions occur.

Note: Ensure that the logging console notifications command is activated so that you can view spectrum management changes.

The following **debug specmgt** command output example displays no ingress noise problems on the active upstream port. The command output displays a time stamp, the error rate, the number of word errors, total word count, and the upstream noise power level in one-tenth of a dBmV.

Note: Ensure that you review the criteria for the hop action rules that you have configured when reviewing the **debug specmgt** and console logging output to clearly understand what is happening in the **debug specmgt** command output and console logging output.

Use the **debug specmgr** command in Privileged EXEC mode to monitor one or more active upstream ports, as shown below:

MOT#**debug specmgr**

Note: In the following example, the IfIndex = 7 entry in the debug output represents a single upstream port on a DOCSIS module.

```
MOT#[07/23-11:00:08:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:00:08:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 2901
[07/23-11:00:08:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-197.
```

The following command output example displays what happens when the ingress noise power increases causing the error rate to exceed the error threshold on an upstream port:

Note: Notice that the noise power level increases.

```
[07/23-11:01:58:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 3723
[07/23-11:01:58:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-152.
[07/23-11:02:08:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:02:18:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:02:18:SPECMGR]-D-Error Rate: 25.8228 %, ErrorWord :
816, TotalWord : 3160
[07/23-11:02:18:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-84.
[07/23-11:02:20:SPECMGR]-D-Updating specTimers
```

The following command output example displays what happens when the spectrum manager initiates the first hop action (the hop action rule is 1 with a priority of 30):

```
[07/23-11:02:38:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:02:38:SPECMGR]-D-Error Rate: 1.3425 %, ErrorWord : 41,
TotalWord : 3054
[07/23-11:02:38:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-68.
```

The following command output example shows that the first hop action succeeded to set a new frequency, and the desired noise power level is restored.

```
[07/23-11:03:07:SPECMGR]-N-Set to new frequency 28000000 for
channel ifIndex = 7 .
[07/23-11:03:08:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:03:08:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 3781
```

```
[07/23-11:03:08:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-198.
```

If the noise power increases at the new frequency, the next hop action rule is to change modulation profile (in this example, the hop action rule is 2 with a priority of 50). Notice that the noise power level continues to increase:

```
[07/23-11:04:58:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:04:58:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 3054
[07/23-11:04:58:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-172.
[07/23-11:05:18:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:05:18:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 4044
[07/23-11:05:18:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-102.
[07/23-11:05:28:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:05:28:SPECMGR]-D-Error Rate: 0.0799 %, ErrorWord : 3,
TotalWord : 3754
[07/23-11:05:28:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-102.
```

Ingress noise causes the error rate to exceed the threshold and the second hop action occurs, as shown below:

```
[07/23-11:05:58:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:05:58:SPECMGR]-D-Error Rate: 1.1573 %, ErrorWord : 37,
TotalWord : 3197
[07/23-11:05:58:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-76.
```

The following output shows that the second hop action to change modulation profile 2 is successful. The noise power level does not change, however, since the modulation type is different in modulation profile 2, than in modulation profile 1, the acceptable noise power threshold is different.

```
[07/23-11:05:58:SPECMGR]-N-Set to new mode profile 2 for channel
ifIndex = 7 .
[07/23-11:06:08:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:06:08:SPECMGR]-D-Error Rate: 0.1759 %, ErrorWord : 5,
TotalWord : 2842
[07/23-11:06:08:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-76.
[07/23-11:07:28:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:07:28:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 3805
```

```
[07/23-11:07:28:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-32.
```

In the following output example, ingress noise causes the error rate to exceed the error threshold, and the next hop action changes the upstream channel width:

```
[07/23-11:08:58:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:08:58:SPECMGR]-D-Error Rate: 4.5083 %, ErrorWord :
182, TotalWord : 4037
[07/23-11:08:58:SPECMGR]-D-Channel Noise Power (1/10 dbmv) : 16.
```

The following output displays that the hop action succeeded and that a new channel width has been assigned to the upstream port by the spectrum manager:

```
[07/23-11:08:59:SPECMGR]-N-Set to new width 1600000, miniSlot 8
for channel ifIndex = 7 .
[07/23-11:09:08:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:09:08:SPECMGR]-D-Error Rate: 0.5769 %, ErrorWord : 17,
TotalWord : 2947
[07/23-11:09:08:SPECMGR]-D-Channel Noise Power (1/10 dbmv) : 16.
```

The following output displays that the noise power level is restored:

```
[07/23-11:09:18:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:09:18:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 3040
[07/23-11:09:18:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-183.
```

Viewing the Spectrum Management Roll-back Function

When the spectrum roll-back function is enabled, the spectrum manager returns the upstream channel width or modulation profile setting, that was adjusted during a hop action, to the original configuration when upstream channel conditions improve.

In the following output example, the roll-back function starts when the ingress noise is removed:

```
[07/23-11:09:28:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:09:28:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 2809
[07/23-11:09:28:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-184.
[07/23-11:09:38:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:09:38:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 2143
```

```
[07/23-11:09:38:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-185.
```

The following output shows the existing upstream channel width reverting to its original upstream channel width setting:

```
[07/23-11:09:39:SPECMGR]-N-Revert to width 3200000, miniSlot 4
succeed for channel ifIndex = 7 .
[07/23-11:09:48:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:09:48:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 1927
[07/23-11:09:48:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-182.
[07/23-11:09:58:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:09:58:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 2108
[07/23-11:09:58:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-197.
[07/23-11:10:08:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:10:08:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 1926
[07/23-11:10:08:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-196.
[07/23-11:10:18:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:10:18:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 2434
[07/23-11:10:18:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-197.
```

The following output shows the existing upstream modulation profile reverting to its original upstream modulation profile setting:

```
[07/23-11:10:19:SPECMGR]-N-Revert to mode profile 1 succeed for
channel ifIndex = 7 .
[07/23-11:10:28:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:10:28:SPECMGR]-D-Error Rate: 0.1513 %, ErrorWord : 2,
TotalWord : 1322
[07/23-11:10:28:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-197.
[07/23-11:10:38:SPECMGR]-D-Monitor Channel IfIndex = 7 :
[07/23-11:10:38:SPECMGR]-D-Error Rate: 0.0000 %, ErrorWord : 0,
TotalWord : 395
[07/23-11:10:38:SPECMGR]-D-Channel Noise Power (1/10 dbmv) :
-197.
```

Using Flap Lists

Flap lists are used to collect statistics for determining CM problems on the network. The CM flap list keeps track of the CM MAC address, up and down transitions, registration events, missed periodic ranging packets, upstream power adjustments on the BSR.

The following sections describe how Flap lists are used:

- Setting Flap List Parameters
- Using Flap Lists to Troubleshoot CM Problems
- Tips for Administering Flap Lists

Setting Flap List Parameters

Flap list parameters are configured to define a criteria for the proper functioning of your cable network.

When a CM makes an insertion request more frequently than the defined insertion time (the time allowed for CMs to complete registration), the CM is placed in the flap list for recording. When the CM power adjustment meets or exceeds its predefined threshold, the CM is placed in the flap list. You can specify the power adjustment threshold to a value that will cause a flap-list event to be recorded. A *miss rate* is the number of times a CM does not acknowledge a MAC layer keepalive message from a CMTS. You can specify the number of seconds to record and retain flapping activity for the CMs connected to the CMTS. The number of CMs that can be recorded in the flap list is 8192 CMs.

Use the following options to set flap list parameters:

1. Use the **cable flap-list aging** command in Global Configuration mode to specify flap list aging, the number of minutes a CM is kept in the flap list, as shown below.

   `MOT(config)#`**cable flap-list aging** {<*1-860400*>}

   where:

   1-860400 is the flap list aging value, expressed in minutes. The default flap list aging value is 1440.

2. To specify the flap list insertion time, use the **cable flap-list insertion-time** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#cable flap-list insertion-time <1-86400>
   ```

   where:

   1-86400 is the flap list insertion time, expressed in seconds. The default flap list insertion time is 60.

3. Use the **cable flap-list power-adjust threshold** command in Global Configuration mode to specify the power adjustment threshold value between 0 to 2 dBmV, as shown below. The power adjustment threshold causes a flap-list event to be recorded when the threshold is exceeded:

Note: Motorola recommends that you do not change the power adjustment threshold from the default value, which is 2 dbmV. Ensure that you evaluate the need to enable this function before applying it to your network. A power adjustment threshold of less than 1 dBmV may cause excessive flap list event recording.

```
MOT(config)#cable flap-list power-adjust threshold {<1-10>}
```

where:

1-10 is the power adjustment threshold value, expressed in dB.

4. The default miss threshold for MAC-layer keepalive messages is 6. If you want to change the threshold number of MAC-layer keepalive message misses that will result in the CMs being recorded in the flap list, use the **cable flap-list miss-threshold** command in Global Configuration mode, as shown below:

Note: A high miss rate can indicate intermittent upstream problems, fiber laser clipping, or common-path distortion.

```
MOT(config)#cable flap-list miss-threshold <1-12>
```

where:

1-12 is the keepalive misses threshold value.

5. To specify the maximum number of CMs that can be recorded in the flap list, use the **cable flap-list size** command in Privileged EXEC mode, as shown below:

```
MOT(config)#cable flap-list size <1-8191>
```

where:

1-8191 is a number that defines the maximum number of CMs.

6. To remove a CM from the flap list, use the **clear cable flap-list** command in Global Configuration mode, as shown below.

```
MOT(config)#clear cable flap-list [<mac>] | all
```

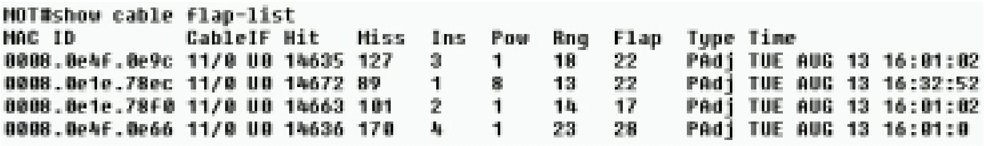
where:

mac is the CM MAC address.

7. To display the CM flap lists and verify cable flap list information, use the **show cable flap-list** command in all modes except User EXEC mode. Refer to Using Flap Lists to Troubleshoot CM Problems for using the various **show cable flap-list** command options.

   MOT#**show cable flap-list**

   Figure 6-7 displays example output for the **show cable flap-list** command.:



```
MOT#show cable flap-list
MAC ID          CableIF Hit    Miss  Ins   Pow   Rng   Flap   Type Time
0008.0e4f.0e9c 11/0 U0 14635 127   3     1     10    22     PAdj TUE AUG 13 16:01:02
0008.0e1e.78ec 11/0 U0 14672 89    1     8     13    22     PAdj TUE AUG 13 16:32:52
0008.0e1e.78f0 11/0 U0 14663 101   2     1     14    17     PAdj TUE AUG 13 16:01:02
0008.0e4f.0e66 11/0 U0 14636 170   4     1     23    28     PAdj TUE AUG 13 16:01:0
```

Figure 6-7 show cable flap-list Command Output

You can also use the **show running-configuration** command, as shown below to display the CM flap lists and verify flap list information:

MOT#**show running-configuration | include flap**

Note: If a value is set to the default, the default value does not display after a **show running-configuration** command.

Using Flap Lists to Troubleshoot CM Problems

The BSR maintains a database of flapping CMs to assist in locating cable plant problems. The flap list feature tracks the upstream and downstream performance of all CMs on the network, without impacting throughput and performance between the CM and BSR, or creating additional packet overhead on the HFC network.

The following tasks are used to troubleshoot CM Problems:

- Viewing Flap List Statistics to Identify Network Health
- Interpreting Flap List Statistics

Viewing Flap List Statistics to Identify Network Health

This section describes the different show cable flap list sorting options and describes the command output fields. CMs appear in the flap list when any of the following conditions are detected:

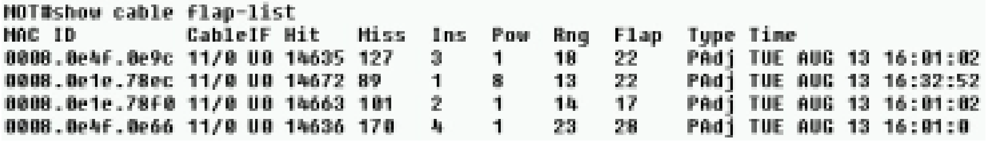
- The CM re-registers more frequently than the configured insertion time.
- Intermittent keepalive messages are detected between the BSR and the CM.
- The CM upstream transmit power changes beyond the configured power adjust threshold.

Follow these steps to view flap list statistics by using different sorting options:

1. To view all flap list statistics for CMs, use the **show cable flap list** command in Privileged EXEC mode as shown below:

   `MOT#`**show cable flap-list**

   The following output displays:



```
MOT#show cable flap-list
MAC ID         CableIF Hit   Miss  Ins  Pow  Rng  Flap  Type Time
0000.0e4f.0e9c 11/0 U0 14635 127   3    1    10   22    PAdj TUE AUG 13 16:01:02
0008.0e1e.78ec 11/0 U0 14672 89    1    8    13   22    PAdj TUE AUG 13 16:32:52
0008.0e1e.78f0 11/0 U0 14663 101   2    1    14   17    PAdj TUE AUG 13 16:01:02
0008.0e4f.0e66 11/0 U0 14636 170   4    1    23   28    PAdj TUE AUG 13 16:01:0
```

Figure 6-8 show cable flap-list Command Output

2. To sort the flap list statistics by the CM flap, use the **show cable flap-list sort-flap** command in Privileged EXEC mode as shown below:

   `MOT#`**show cable flap-list sort-flap**

3. To sort the flap list statistics by the time at which the CM flap occurred, use the **show cable flap-list sort-time** command in Privileged EXEC mode as shown below:

   `MOT#`**show cable flap-list sort-time**

4. To sort the flap list statistics by the cable upstream interface on which the CM flap occurred, use the **show cable flap-list sort-interface** command in Privileged EXEC mode as shown below:

   `MOT#`**show cable flap-list sort-interface**

Table 6-8 lists the **show cable flap-list** command output fields and their descriptions:

Table 6-8 show cable flap-list Command Output Field Display Fields

Field	Identification
MAC ID	Lists the MAC addresses of the CMs sorted by the flap rate or most recent flap time. The first six digits in the CM MAC address indicate the vendor ID of the CM manufacturer, followed by six digits indicating a unique host address. Each CM MAC address is unique.
Cable IF	Detects the cable interface up/down flap. This is the cable interface on the BSR 64000 DOCSIS module. It denotes the DOCSIS module slot number (BSR 64000), the downstream and the upstream port number. The flap list data can be sorted based on the upstream port number which is useful when isolating reverse path problems unique to certain combining groups.
Ins	The Insertions Link process is used by a CM to perform an initial maintenance procedure to establish a connection with the BSR. The Ins column is the flapping CM's (re) insertion count and indicates the number of times the a CM starts and inserts into the network in an abnormal way. An abnormality is detected when the time between link re-establishment attempts is less than the user-configured parameter. This function can identify potential problems in the downstream interface such as incorrectly provisioned CMs repeatedly trying to reestablish a link.
Hit	The Hit and Miss column fields detect the intermittent upstream; the keepalive hits versus misses is the number of times CMs do not respond to the MAC layer keepalive messages. If there are a number of misses, this points to a potential upstream problem.
Miss	
Pow	The Power Adjustment column field shows power adjustment statistics during station maintenance polling. This column indicates the number of times the BSR tells a CM to adjust the transmit power more than the configured threshold. If constant power adjustments are detected, an amplifier problem is usually the cause. The source of failure is found by viewing CMs either in front or behind various amplifiers. • An exclamation point appears when the CM has reached its maximum power transmit level and cannot increase its power level any further.
Flap	Indicates the number of times the CM has flapped.
Rng	Indicates the number of times the CM has ranged.

Table 6-8 show cable flap-list Command Output Field Display Fields

Field	Identification
Type	Indicates the type of event that triggered the flap.
Time	Indicates the most recent time a flap has occurred for a particular CM.

Interpreting Flap List Statistics

This section describes how to interpret flap list statistics in order to troubleshoot the cable network

CM activity follows the sequence below.

- Power-on
- Initial maintenance
- Station maintenance
- Power-off

The initial link insertion is followed by a keepalive loop between the BSR and CM and is called station maintenance. When the link is broken, initial maintenance is repeated to re-establish the link.

Initial maintenance @ Time T1

Station maintenance

Init maintenance @ Time T2

The **Ins** and **Flap** counters in the flap list are incremented whenever **T2 – T1 < N** where **N** is the insertion-time parameter configured using the **cable flap-list insertion-time** command. The default value for this parameter is TBD seconds.

Use the following cause or symptom observations to interpret flap list activity and solve CM problems:

Table 6-9 Troubleshooting CM Problems

Cause or Symptom	Problem
Subscriber CM shows a lot of flap list activity	CM is having communication problems with the BSR.
Subscriber CM shows little or no flap list activity.	The CM is communicating with the BSR effectively, however there is still a problem. The problem can be isolated to the subscriber's CPE computer equipment or the CM connection.
Ten percent of the CMs in the flap list show a lot of activity.	These CMs are most likely having difficulties communicating with the BSR.
CMs have a lot of power adjustment (P-Adj) errors.	CMs have problems with their physical upstream paths or in-home wiring problems. Use corresponding CMs on the same physical upstream port interface with similar flap list statistics to quickly resolve problems outside the cable plant to a particular node or geographic location.
All CMs are incrementing the insertion at the same time.	There is a provisioning server failure.
A CM has more than 50 power adjustments per day.	The CM has a suspect upstream path. Corresponding CMs on the same physical upstream port interface with similar flap list statistics can be used to quickly resolve problems outside the cable plant to a particular node or geographic location.
A CM has roughly the same number of hits and misses and contain a lot of insertions.	There is a problematic downstream path. For example, the downstream power level to the CM may have a power level that is too low.
A high flap list insertion (Ins) time number.	Intermittent downstream synchronization loss. DHCP or CM registration problems.
Low miss/hit ratio, low insertion, low P-adj, low flap counter and old timestamp.	Indicates an optimal network situation.

Table 6-9 Troubleshooting CM Problems

Cause or Symptom	Problem
High ratio of misses over hits (> 10%)	Hit/miss analysis should be done after the "Ins" count stops incrementing. In general, if the hit and miss counts are about the same order of magnitude, then the upstream may be experiencing noise. If the miss count is greater, then the CM is probably dropping out frequently and not completing registration. The upstream or downstream is perhaps not stable enough for reliable link establishment. Very low hits and miss counters and high insertion counters indicate provisioning problems.
High power adjustment counter.	Indicates the power adjustment threshold is probably set at default value of 2 dB adjustment. The CM transmitter step size is 1.5 dB, whereas the headend may command 0.25 dB step sizes. Tuning the power threshold to 6 dB is recommended to decrease irrelevant entries in the flap list. The power adjustment threshold may be set using *<cable flap power threshold <0-10 dB>* from Global Configuration mode. A properly operating HFC network with short amplifier cascades can use a 2-3 dB threshold.
High P-Adj (power adjustment)	This condition can indicate that the fiber node is clipping the upstream return laser. Evaluate the CMs with the highest number of correcteds and uncorrecteds first. If the CMs are not going offline (Ins = 0), this will not be noticed by the subscriber. However, they could receive slower service due to dropped IP packets in the upstream. This condition will also result in input errors on the cable interface.
High insertion rate.	If link re-establishment happens too frequently, then the CM is usually having a registration problem.This is indicated by a high ‘Ins’ counter which tracks the ‘Flap’ counter.

Note: CMs go offline faster than the frequency hop period and can cause the frequency to stay fixed while CMs go offline. Reduce the hop period to 10 seconds to adjust to the hop frequency period.

Table 6-10 describes how to interpret flap list statistics:

Table 6-10 Flap List Statistic Interpretations

Field	Description
Hit and Miss	The HIT and MISS columns are keepalive polling statistics between the BSR and the CM. The station maintenance process occurs for every CM approximately every 10 seconds. When the BSR receives a response from the CM, the event is counted as a Hit. If the BSR does not receive a response from the CM, the event is counted as a Miss. A CM will fail to respond either because of noise or if it is down. CMs which only log Misses and zero Hits are assumed to be powered off. Misses are not desirable since this is usually an indication of a return path problem; however, having a small number of misses is normal. The flap count is incremented if there are M consecutive misses where M is configured in the cable flap miss-threshold parameter. The parameter value ranges from 1-12 with a default of 6. Ideally, the HIT count should be much greater than the Miss counts. If a CM has a HIT count much less than its MISS count, then registration is failing. Noisy links cause the MISS/HIT ratio to deviate from a nominal 1% or less. High Miss counts can indicate: • Intermittent upstream possibly due to noise • Laser clipping • Common-path distortion • Ingress or interference Too much or too little upstream attenuation
P-Adj	The station maintenance poll in the BSR constantly adjusts the CM transmit power, frequency, and timing. The Power Adjustments (P-Adj) column indicates the number of times the CM's power adjustment exceeded the threshold value. The power adjustment threshold may be set using the <*cable flap power threshold* > parameter with a value range of 0-10 dB and a default value of 2 dB. Tuning this threshold is recommended to decrease irrelevant entries in the flap list. Power Adjustment values of 2 dB and below will continuously increment the P-Adj counter. The CM transmitter step size is 1.5 dB, whereas the cable interface may command 0.25 dB step sizes. Power adjustment flap strongly suggests upstream plant problems such as: • Amplifier degradation • Poor connections • Thermal sensitivity Attenuation problem

Table 6-10 Flap List Statistic Interpretations

Field	Description
Flap	The Flap counter indicates the number of times the CM has flapped. This counter is incremented when one of the following events is detected: Unusual CM insertion or re-registration attempts. The Flap and the Ins counters are incremented when the CM tries to re-establish the RF link with the BSR within a period of time that is less than the user-configured insertion interval value. Abnormal Miss/Hit ratio The Flap counter is incremented when N consecutive Misses are detected after a Hit where N can be user-configured with a default value of 6. Unusual power adjustment The Flap and P-adj counters are incremented when the CM's upstream power is adjusted beyond a user-configured power level.
Time	Time is the timestamp indicating the last time the CM flapped. The value is based on the clock configured on the local BSR. If no time is configured, this value is based on the current uptime of the BSR. When a CM meets one of the three flap list criteria, the Flap counter is incremented and Time is set to the current time.

Tips for Administering Flap Lists

Follow these suggestions for administrating flap lists:

- Write script(s) to periodically poll the flap list.
- Analyze and identify CM trends from the flap list data.
- Query the billing and administrative database for CM MAC address-to-street address translation and generate reports. These reports can then be given to the Customer Service Department or the cable plant's Operations and Maintenance Department. Maintenance personnel use the reports to see patterns of flapping CMs, street addresses, and flap statistics that indicate which amplifier or feeder lines are faulty. The reports also help troubleshoot problems in the downstream and/or upstream path, and determine if a problem is related to ingress noise or equipment.

- Save the flap list statistics to a database server at least once a day to keep a record of flap list statistics which includes upstream performance and quality control data. These statistics can be used again at a later time to evaluate trends and solve intermittent problems on the HFC networks. Once the flap list statistics are backed up daily on the database server, the flap list statistics can be cleared.

Managing Multicast Maps

Follow these options to manage multicast maps on the BSR:

- To create a multicast map, use the **cable privacy mcast new** command in Cable Interface mode, as shown below:

  ```
  MOT(config)#cable privacy mcast new {<A.B.C.D>} {<NUM>}
  ```

  where:

  A.B.C.D is the IP address.

  NUM is the prefix length.

- To delete a multicast map, use the **cable privacy mcast del** command in Cable Interface mode, as shown below:

  ```
  MOT(config)#cable privacy mcast del {<A.B.C.D>} {<NUM>}
  ```

  where:

  A.B.C.D is the IP address.

  NUM is the prefix length.

- To configure a multicast address list, use the **cable privacy mcast access** command in Cable Interface mode, as shown below:

  ```
  MOT(config)#cable privacy mcast access {<mac>} {<A.B.C.D>}
  ```

  where:

  mac is the MAC address.

  prefix is the IP address.

- To duplicate a CM certificate to provisioned certificate table, use the **cable privacy provision-cert-add** command in Privileged EXEC mode as shown below:

  `MOT(config)#`**cable privacy provision-cert-add** {<*mac*>}

  where:

  mac is the MAC address.

Pinging a Cable Modem at the MAC Layer

The **ping DOCSIS** command is used to "ping" or find a cable modem (CM) on the network at the MAC layer by entering the CM's MAC or IP address.

When a DOCSIS ping is initiated, the BSR sends a test packet downstream towards the CM to test its connection. In most instances, this command is used to determine if a particular CM is able to communicate at the MAC address layer when a cable modem has connectivity problems at the network layer. For example, if a CM is unable to register and obtain an IP address, the ping DOCSIS command can help you determine if there are provisioning problems associated with the CM.

Follow these steps to use the ping DOCSIS function:

1. To specify the number of DOCSIS ping packets, use the **ping docsis size** command in Privileged EXEC mode, as shown below:

   `MOT#`**ping docsis size** <*n*>

   where:

   n is the number of ping test packets directed toward a CM.

2. To enter Global Configuration mode, use the **config** command in Privileged EXEC mode.

3. To enter the cable interface, use the **interface cable** command in Global Configuration mode.

4. To determine if a CM is online, use the **ping docsis** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#ping docsis [<mac> | <prefix>]
```

Note: Be sure to use the correct MAC or IP address of the CM.

where:

mac is the MAC address of the CM.

prefix is the IP address of the CM.

Resetting the Cable Modem

Use these options to reset cable modems (CMs) on the DOCSIS network:

- To reset a single CM by using its MAC address, use the **clear cable modem reset** command in Privileged EXEC mode, as shown below:

  ```
  MOT#clear cable modem <mac> reset
  ```

  where:

  mac is the CM MAC address.

- To reset specific group of CMs, use the **clear cable modem reset** command in Privileged EXEC mode, as shown below:

  ```
  MOT#clear cable modem {<mac> <mac-mask>} reset
  ```

  where:

  mac is the CM MAC address.

  mac-mask is the MAC address mask that specifies a group of cable modems

- To reset a single CM by using its IP address, use the **clear cable modem reset** command in Privileged EXEC mode, as shown below:

`MOT#`**clear cable modem** <*ip-address*> **reset**

where:

ip-address is the CM IP address.

- To reset all CMs connected to the BSR, use the **clear cable modem all reset** command in Privileged EXEC mode, as shown below:

  `MOT#`**clear cable modem all reset**

Clearing Cable Interface Counters

To clear the counters for a cable interface, use the **clear counters cable** command in any mode as shown below:

MOT#**clear counters cable** <*x*>/<*y*>

where:

x is the slot number of the cable module.

y is the cable interface number, which is **0**.

Gathering DOCSIS Network Information

The following sections describe how to use **show** commands to gather network information from the BSR:

- Displaying Cable Interface Statistics
- Displaying Downstream Parameters
- Displaying Upstream Parameters
- Viewing CM Information
- Displaying Modulation Profiles
- Displaying BPI Configuration Settings

Displaying Cable Interface Statistics

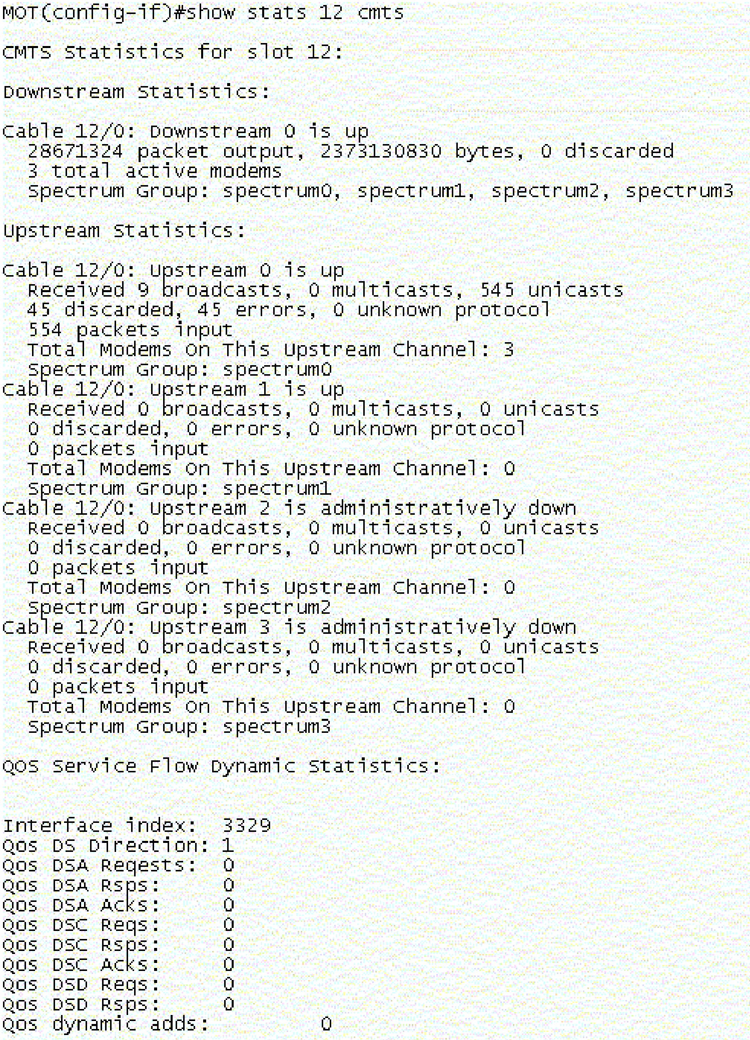
Use the **show stats cmts** command in all modes except User EXEC mode to view cable interface statistics, which includes both downstream and upstream port statistics and QOS service flow dynamic statistics, as shown below:

`MOT(config-if)#`**show stats** <*NUM*> **cmts**

where:

NUM is the slot number of one or more available cable modules.

Figure 6-9 displays the **show stats cmts** command output:



```
MOT(config-if)#show stats 12 cmts

CMTS Statistics for slot 12:

Downstream Statistics:

Cable 12/0: Downstream 0 is up
  28671324 packet output, 2373130830 bytes, 0 discarded
  3 total active modems
  Spectrum Group: spectrum0, spectrum1, spectrum2, spectrum3

Upstream Statistics:

Cable 12/0: Upstream 0 is up
  Received 9 broadcasts, 0 multicasts, 545 unicasts
  45 discarded, 45 errors, 0 unknown protocol
  554 packets input
  Total Modems On This Upstream Channel: 3
  Spectrum Group: spectrum0
Cable 12/0: Upstream 1 is up
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 discarded, 0 errors, 0 unknown protocol
  0 packets input
  Total Modems On This Upstream Channel: 0
  Spectrum Group: spectrum1
Cable 12/0: Upstream 2 is administratively down
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 discarded, 0 errors, 0 unknown protocol
  0 packets input
  Total Modems On This Upstream Channel: 0
  Spectrum Group: spectrum2
Cable 12/0: Upstream 3 is administratively down
  Received 0 broadcasts, 0 multicasts, 0 unicasts
  0 discarded, 0 errors, 0 unknown protocol
  0 packets input
  Total Modems On This Upstream Channel: 0
  Spectrum Group: spectrum3

QOS Service Flow Dynamic Statistics:


Interface index:  3329
Qos DS Direction: 1
Qos DSA Reqests:  0
Qos DSA Rsps:     0
Qos DSA Acks:     0
Qos DSC Reqs:     0
Qos DSC Rsps:     0
Qos DSC Acks:     0
Qos DSD Reqs:     0
Qos DSD Rsps:     0
Qos dynamic adds:         0
```

Figure 6-9 show stats cmts Command Output

Displaying Downstream Parameters

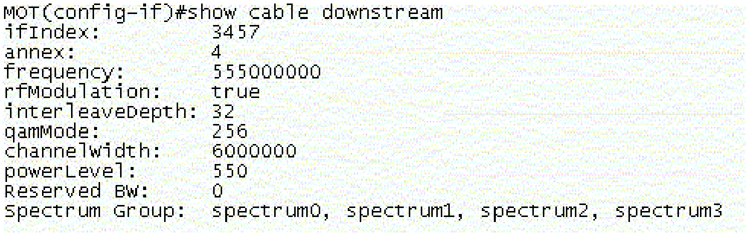
To show the configured parameters for the downstream port, use the **show cable downstream** command in Interface Configuration mode, as shown below:

`MOT(config-if)#`**show cable downstream** <*0-0*>

where:

0-0 is the downstream port number.

Figure 6-10 displays the **show cable downstream** command output:



```
MOT(config-if)#show cable downstream
ifIndex:          3457
annex:            4
frequency:        555000000
rfModulation:     true
interleaveDepth:  32
qamMode:          256
channelWidth:     6000000
powerLevel:       550
Reserved BW:      0
Spectrum Group:   spectrum0, spectrum1, spectrum2, spectrum3
```

Figure 6-10 show cable downstream Command Output

Viewing Downstream Port Information

Use the **show interfaces cable downstream** command in all modes except User EXEC mode to view downstream port statistics, as shown below:

`MOT#`**show interfaces cable** <*x*>/<*y*> **downstream** <*0-0*>

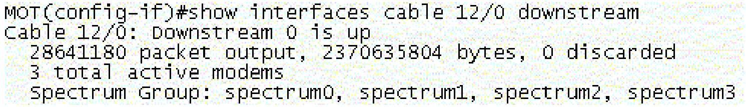
where:

x is the slot number of the cable module.

y is the cable interface number, which is **0**.

port is the downstream port number.

Figure 6-11 displays the **show interfaces cable downstream** command output:



```
MOT(config-if)#show interfaces cable 12/0 downstream
Cable 12/0: Downstream 0 is up
  28641180 packet output, 2370635804 bytes, 0 discarded
  3 total active modems
  Spectrum Group: spectrum0, spectrum1, spectrum2, spectrum3
```

Figure 6-11 show interfaces cable downstream stats Command Output

Displaying Upstream Parameters

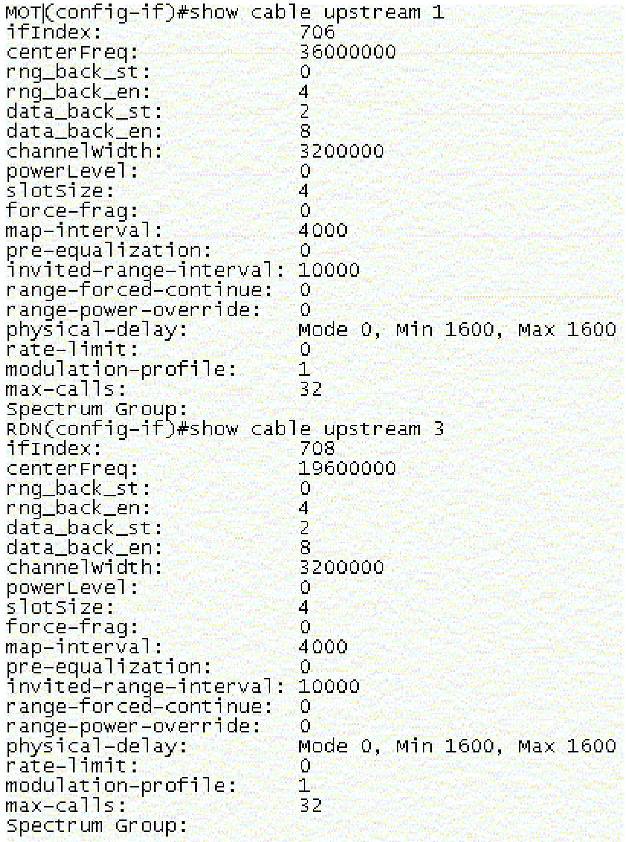
To show the configured upstream parameters, use the **show cable upstream** command in Interface Configuration mode, as shown below:

MOT(config-if)#**show cable upstream** <*NUM*>

where:

NUM is the upstream port number.

Figure 6-12 displays the **show cable upstream** command output:



```
MOT(config-if)#show cable upstream 1
ifIndex:                    706
centerFreq:                 36000000
rng_back_st:                0
rng_back_en:                4
data_back_st:               2
data_back_en:               8
channelWidth:               3200000
powerLevel:                 0
slotSize:                   4
force-frag:                 0
map-interval:               4000
pre-equalization:           0
invited-range-interval: 10000
range-forced-continue:  0
range-power-override:   0
physical-delay:             Mode 0, Min 1600, Max 1600
rate-limit:                 0
modulation-profile:         1
max-calls:                  32
Spectrum Group:
RDN(config-if)#show cable upstream 3
ifIndex:                    708
centerFreq:                 19600000
rng_back_st:                0
rng_back_en:                4
data_back_st:               2
data_back_en:               8
channelWidth:               3200000
powerLevel:                 0
slotSize:                   4
force-frag:                 0
map-interval:               4000
pre-equalization:           0
invited-range-interval: 10000
range-forced-continue:  0
range-power-override:   0
physical-delay:             Mode 0, Min 1600, Max 1600
rate-limit:                 0
modulation-profile:         1
max-calls:                  32
Spectrum Group:
```

Figure 6-12 show cable upstream Command Output

Viewing Upstream Port Information

Use the **show interfaces cable upstream** command in all modes except User EXEC mode to view upstream port statistics, as shown below:

MOT#**show interfaces cable** <*x*>/<*y*> **upstream** [**stats** | **signal-quality** | **spectrum** <*start-freq*> <*end-freq*>]

where:

x is the cable module slot number.

y is the cable interface number, which is **0**.

stats provides upstream information in a statistical format.

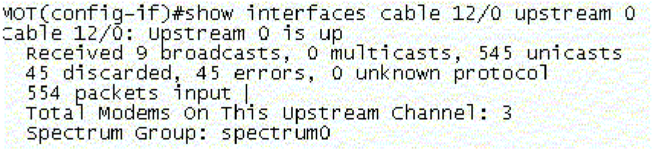
signal-quality displays upstream port RF signal quality information.

spectrum displays upstream port spectrum information for power levels comparing the upstream frequency to the number of microvolts and dBmV.

start-freq is the upstream start of the frequency range from 5000000 to 42000000 Hertz (Hz).

end-freq is the upstream end of the frequency range from 5000000 to 42000000 Hertz (Hz).

Figure 6-13 displays the **show interfaces cable upstream** command output:



```
MOT(config-if)#show interfaces cable 12/0 upstream 0
Cable 12/0: Upstream 0 is up
  Received 9 broadcasts, 0 multicasts, 545 unicasts
  45 discarded, 45 errors, 0 unknown protocol
  554 packets input |
  Total Modems On This Upstream Channel: 3
  Spectrum Group: spectrum0
```

Figure 6-13 show interfaces cable upstream Command Output

Viewing CM Information

The **show cable modem** command, listed in Table 6-11, lets you view statistical information about cable modems (CMs) connected to the BSR. The information gathered helps you to evaluate network performance, troubleshoot registration problems, and determine registration status and learn ranging information.

Table 6-11 describes the **show cable modem** command output column fields.

Table 6-11 Cable Modem Fields

Field	Identification
Interface	CM interface with active connection
Upstream IF Index	Upstream interface to which the cable modem belongs.
Prim SID	Primary Service Identifier number.

Table 6-11 Cable Modem Fields

Field	Identification
Connectivity State	Describes the connectivity state of a cable modem. Refer to Table 6-12 for more information on each connectivity state.
Timing offset	CM current timing adjustment.
Rec Power	CM receive downstream power level in dbmv.
IP address	CM IP address
MAC address	Media Access Control layer address

Use these options to view cable modem information:

- To display information for all CMs on connected to the BSR, use the **show cable modem** command in Privileged EXEC mode:

  MOT#**show cable modem**

- To display information for a specific CM connected to the BSR, use the **show cable modem** command, as shown below:

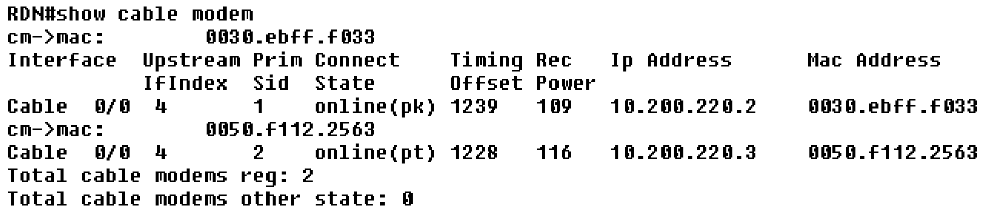
  MOT#**show cable modem** [<*mac*> | <*prefix*>]

  where:

  mac is the CM MAC address.

  prefix is the CM IP address.

  Figure 6-14 describes the **show cable modem** command output:



```
RDN#show cable modem
cm->mac:        0030.ebff.f033
Interface  Upstream Prim Connect    Timing Rec   Ip Address      Mac Address
           IfIndex  Sid  State      Offset Power
Cable  0/0  4       1    online(pk) 1239   109   10.200.220.2    0030.ebff.f033
cm->mac:        0050.f112.2563
Cable  0/0  4       2    online(pt) 1228   116   10.200.220.3    0050.f112.2563
Total cable modems reg: 2
Total cable modems other state: 0
```

Figure 6-14 show cable modem Command Output

Table 6-12 describes the 20 cable modem connectivity states.

Table 6-12 Cable Modem Connectivity States

Connectivity State	Identification
init(o)	Option file transfer was started.
init(t)	Time-of-day (TOD) exchange was started.
init(r1)	CM sent initial ranging parameters.
init(r2)	CM is ranging.
init(rc)	Ranging is complete.
dhcp(d)	DHCP Discover was sent by CM.
dhcp(o)	DHCP Offer was received.
dhcp(req)	DHCP Request was sent by CM.
dhcp(ack)	DHCP Ack was received, IP address was assigned by DHCP server.
online	CM registered; enabled for data.
online(d)	CM registered, but network access for the CM is disabled.
online(un)	CM registered, but not enabled data. Fail to verify modem’s identity by BPI module.
online(pk)	CM registered; baseline privacy interface (BPI) enabled, and key encryption key (KEK) is assigned.
online(pt)	CM registered; BPI enabled, and traffic encryption key (TEK) is assigned.
reject(m)	CM did attempt to register; registration was refused due to bad mic.
reject(c)	CM did attempt to register; registration was refused due to bad COS.
reject(r)	CM did attempt to register, registration was refused due to unavailable resource.
reject(pk)	KEK modem key assignment is rejected.
reject(pt)	TEK modem key assignment is rejected.
offline	CM is considered to be offline.

- Use the following options to view Customer Premises Equipment (CPE) information:
  - If you want to display CPE information for all cable interfaces, use the **show cable modem-cpe** command as shown bellow:

    `MOT#`**show cable modem cpe**

  - If you want to display information for a CPE IP or MAC address, use the **show cable modem-cpe** command, as shown below:

    `MOT#`**show cable modem cpe** [<*mac*> | <*prefix*>]

    where:

    <*mac*> is the CPE MAC address.

    <*prefix*> is the CPE IP address.

  - If you want to display CPE information for a particular upstream port, use the **show cable modem-cpe upstream** command, as shown below:

    `MOT#`**show cable modem cpe** <*x*>/<*y*> **upstream** <*NUM*>

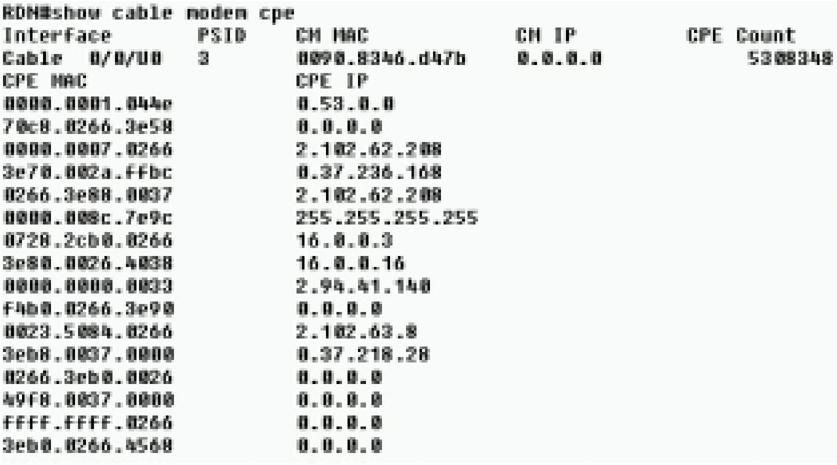
    where:

    x is the slot number of the cable module.

    y is the cable interface number, which is **0**.

    NUM is the upstream port number.

Figure 6-15 displays the cable modem cpe statistics for each cable interface:



```
RDN#show cable modem cpe
Interface          PSID      CM MAC               CM IP              CPE Count
Cable   0/0/U0     3         0090.8346.d47b       0.0.0.0                 5308348
CPE MAC                      CPE IP
0000.0001.044e               0.53.0.0
70c8.0266.3e58               0.0.0.0
0000.0007.0266               2.102.62.208
3e70.002a.ffbc               0.37.236.168
0266.3e88.0037               2.102.62.208
0000.008c.7e9c               255.255.255.255
0728.2cb0.0266               16.0.0.3
3e80.0026.4038               16.0.0.16
0000.0000.0033               2.94.41.140
f4b0.0266.3e90               0.0.0.0
0023.5084.0266               2.102.63.8
3eb8.0037.0000               0.37.218.28
0266.3eb0.0026               0.0.0.0
49f8.0037.0000               0.0.0.0
ffff.ffff.0266               0.0.0.0
3eb0.0266.4568               0.0.0.0
```

Figure 6-15 show cable modem cpe Command Output

- To display information for a SID assigned to a CM on a specific DOCSIS interface, use the **show cable modem detail** command in Privileged Exec mode, as shown below:

  MOT#**show cable modem detail** <*x*>/<*y*> <*sid*>

  where:

  x is the slot number of the cable module.

  y is the cable interface number, which is **0**, to which the CM is connected.

  sid is the Service Identifier assigned to a CMs.

- Use the **show cable modem detail** command in Privileged Exec mode to display information for a specific modem connected to a specific interface, as shown below:

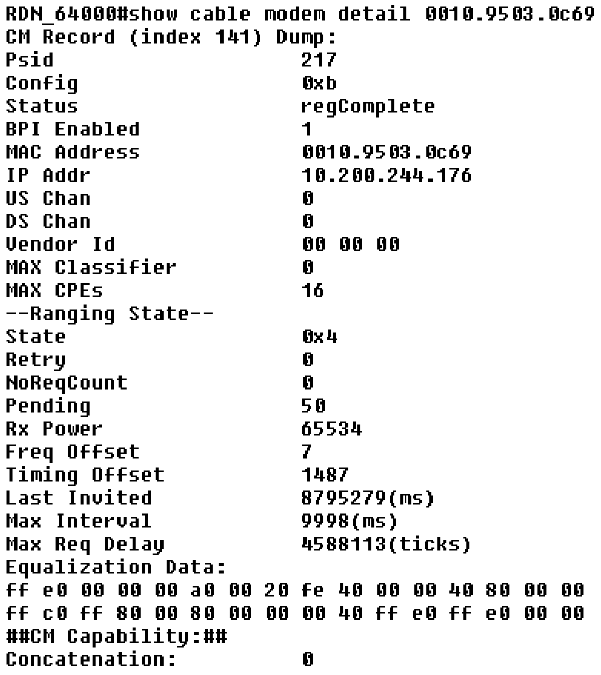
  MOT#**show cable modem detail** [<*mac*> | <*prefix*>]

  where:

  mac is the CM MAC address.

  prefix is the CM IP address.

Figure 6-16 displays output information for the **show cable modem detail** command:



```
RDN_64000#show cable modem detail 0010.9503.0c69
CM Record (index 141) Dump:
Psid                    217
Config                  0xb
Status                  regComplete
BPI Enabled             1
MAC Address             0010.9503.0c69
IP Addr                 10.200.244.176
US Chan                 0
DS Chan                 0
Vendor Id               00 00 00
MAX Classifier          0
MAX CPEs                16
--Ranging State--
State                   0x4
Retry                   0
NoReqCount              0
Pending                 50
Rx Power                65534
Freq Offset             7
Timing Offset           1487
Last Invited            8795279(ms)
Max Interval            9998(ms)
Max Req Delay           4588113(ticks)
Equalization Data:
ff e0 00 00 00 a0 00 20 fe 40 00 00 40 80 00 00
ff c0 ff 80 00 80 00 00 00 40 ff e0 ff e0 00 00
##CM Capability:##
Concatenation:          0
```

Figure 6-16 show cable modem detail Command Output

- The **show cable modem summary** command displays information for the total number of CMs, registered CMs, and unregistered CMs:
  - Registered modems are modems which have reached the Online(d), Online (pk), Online(pt) or Online(un) states.
  - Active modems are those modems in any Init, DHCP or Reject state or substate. All other modems are assumed to be powered off.
  - Unregistered modems are those modems in any Init, DHCP or Reject state or substate. Offline modems are any CMs which have no state, are not communicating, but are remembered because they previously were provisioned. These modems are assumed to be powered off.

  Use the following options to view CM summary information:

- Use the **show cable modem summary** command to display the total number of registered, unregistered and offline CMs for cable interfaces on the BSR 64000:

  `MOT#`**show cable modem summary**

- Use the **show cable modem summary total** command to display the total number of registered, unregistered and offline CMs for a specific cable module:

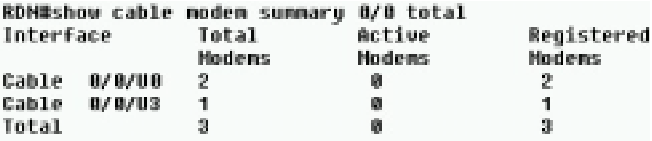
  `MOT#`**show cable modem summary** [<*x*>/<*y*> **total**]

  where:

  x is the cable module slot number.

  y is the cable interface number, which is **0**.

  Figure 6-17 displays the total number of CMs, active CMs and registered CMs for a specific cable interface in the following example:



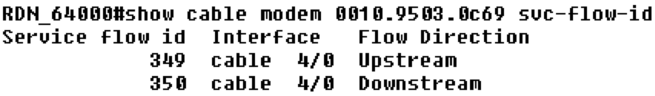
```
RDN#show cable modem summary 0/0 total
Interface          Total          Active          Registered
                   Modems         Modems          Modems
Cable   0/0/U0     2               0               2
Cable   0/0/U3     1               0               1
Total              3               0               3
```

Figure 6-17 show cable modem summary total Command Output

- To view the service flow ID for a CM connected to a slot and cable interface on the BSR, use the **show cable modem svc-flow-id** command in Privileged Exec mode, as shown below:

  `MOT#`**show cable modem svc-flow-id**

  Figure 6-18 displays the **show cable modem svc-flow-id** command output:



```
RDN_64000#show cable modem 0010.9503.0c69 svc-flow-id
Service flow id  Interface   Flow Direction
            349  cable  4/0  Upstream
            350  cable  4/0  Downstream
```

Figure 6-18 show cable modem svc-flow-id Command Output

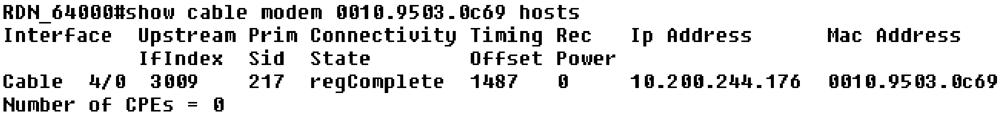
- To display the number of Customer Premises Equipment (CPE) hosts connected to a specific CM, use the **show cable modem hosts** command in Privileged EXEC mode, as shown below:

  `MOT#`**show cable modem** {<*mac*> | <*prefix*>} **hosts**

  where:

  > *mac* is the CM MAC address.
  >
  > *prefix* is the CM IP address.

  Figure 6-19 displays the current number of hosts connected to the CM.



```
RDN_64000#show cable modem 0010.9503.0c69 hosts
Interface  Upstream Prim Connectivity Timing Rec   Ip Address      Mac Address
           IfIndex  Sid  State        Offset Power
Cable  4/0  3009    217  regComplete  1487   0     10.200.244.176  0010.9503.0c69
Number of CPEs = 0
```

Figure 6-19 show cable modem hosts Command Output

- Use the **show cable modem offline** command to display offline CMs only, as shown below:

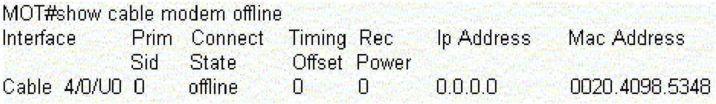
Note: The show cable modem offline command output is updated if the aging timer interval expires for an offline CM. Also, the CM offline table can contain 6100 entries. If this total number is reached and a new CM goes offline, the oldest entry in the table is deleted.

`MOT#`**show cable modem offline** [<*mac*> | <*0-15*>]

where:

> *mac* is the MAC hardware address of the CM.
>
> *0-15* is the slot number of the cable module to which CMs are associated.

Figure 6-20 displays the current number of offline CMs.



```
MOT#show cable modem offline
Interface       Prim  Connect   Timing  Rec       Ip Address      Mac Address
                Sid   State     Offset  Power
Cable  4/0/U0  0     offline    0       0         0.0.0.0         0020.4098.5348
```

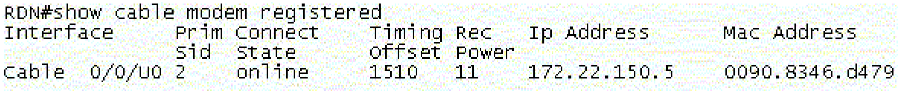
Figure 6-20 show cable modem offline Command Output

- Use the **show cable modem registered** command to display registered CMs only, as shown below:

  `MOT#`**show cable modem registered**

  where:

  Figure 6-21 displays the current number of registered CMs.



```
RDN#show cable modem registered
Interface      Prim Connect     Timing Rec    Ip Address       Mac Address
               Sid  State       Offset Power
Cable  0/0/U0 2    online      1510   11    172.22.150.5     0090.8346.d479
```

Figure 6-21 show cable modem registered Command Output

- Use the **show cable modem unregistered** command to display unregistered CMs only by filtering online and reject states, as shown below:

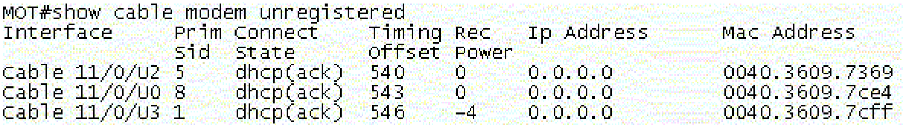
  `MOT#`**show cable modem registered**

  where:

  The following online and reject states are filtered:

  init(o),init(t), init(r1), init(r2), init(rc), dhcp(d), dhcp(req), dhcp(ack), offline

  Figure 6-22 displays the current number of unregistered CMs.



```
MOT#show cable modem unregistered
Interface      Prim Connect     Timing Rec    Ip Address       Mac Address
               Sid  State       Offset Power
Cable 11/0/U2 5    dhcp(ack)   540    0     0.0.0.0          0040.3609.7369
Cable 11/0/U0 8    dhcp(ack)   543    0     0.0.0.0          0040.3609.7ce4
Cable 11/0/U3 1    dhcp(ack)   546    -4    0.0.0.0          0040.3609.7cff
```

Figure 6-22 show cable modem unregistered Command Output

- Use the **show cable modem time-registered** command to display the Spectrum Group for the CM and how long the CM has been registered, as shown below:

  `MOT#`**show cable modem time-registered** [<*mac*> | <*slot*> | **spectrum-group** <*WORD*>]

  where:

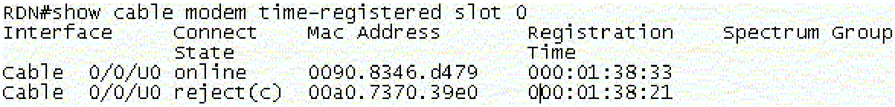
  mac is the MAC hardware address of the CM.

slot is the slot number of the cable module to which CMs are associated.

spectrum-group is used to identify a spectrum group.

WORD is the spectrum group name to which the CM belongs.

Figure 6-23 displays the current number of registered CMs for a cable module slot.



```
RDN#show cable modem time-registered slot 0
Interface     Connect    Mac Address       Registration    Spectrum Group
              State                        Time
Cable  0/0/U0 online     0090.8346.d479    000:01:38:33
Cable  0/0/U0 reject(c)  00a0.7370.39e0    000:01:38:21
```

Figure 6-23 show cable modem time-registered Command Output

- The **show cable modem mac** command displays MAC layer (layer 2) information for CMs.

  Use the following options to view CM MAC layer information:

  - Use the **show cable modem mac** command to view MAC layer information for CMs on a specific cable module, as shown below:

    `MOT#`**show cable modem mac** {<*x*>/<*y*>}

    where:

    x is the slot number of the cable module.

    y is the cable interface number, which is **0**.

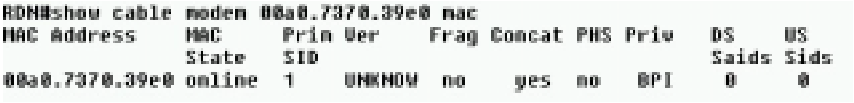
  - Use the **show cable modem mac** command to view MAC layer information for a specific CM, as shown below:

    `MOT#`**show cable modem** <*mac*> **mac**

    where:

    mac is the CM MAC address.

Figure 6-24 displays MAC layer statistics for a specific CM:



```
RDN#show cable modem 00a0.7370.39e0 mac
MAC Address    MAC     Prin Ver    Frag Concat PHS Priv   DS    US
               State   SID                                Saids Sids
00a0.7370.39e0 online  1    UNKNOW  no     yes  no   BPI     0     0
```

Figure 6-24 show cable modem mac Command Output

- The **show cable modem phy** command displays physical hardware information for CMs.

  Use the following options to view CM physical layer information:

  - Use the **show cable modem phy** command to view physical layer information for CMs on a specific cable module, as shown below:

    `MOT#`**show cable modem phy** {<*x*>/<*y*>}

    where:

    x is the slot number of the cable module.

    y is the cable interface number, which is **0**.

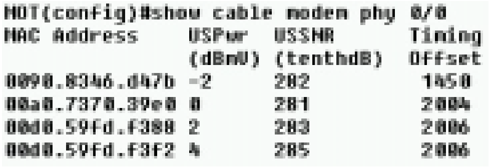
  - Use the **show cable modem phy** command to view physical layer information for a specific CM, as shown below:

    `MOT#`**show cable modem** <*mac*> **phy**

    where:

    mac is the CM MAC address.

    Figure 6-25 displays Physical layer statistics for a specific CM:



```
MOT(config)#show cable modem phy 0/0
MAC Address    USPwr  USSNR      Timing
               (dBmV) (tenthdB)  Offset
0090.8346.d47b -2     202         1450
00a0.7370.39e0 0      281         2004
00d0.59fd.f300 2      203         2006
00d0.59fd.f3f2 4      205         2006
```

Figure 6-25 show cable modem phy Command Output

- The **show cable modem maintenance** command is used to veiw station maintenance statistics, which includes station maintenance retries, station maintenance failures, and recent event timestamps.

  Cable modem (CM) station maintenance ranging, which occurs during the CM registration process, uses periodic time intervals to send a unicast message containing a registered SID between the cable modem and the CMTS.

  Use the following options to display station maintenance statistics:

  - Use the **show cable modem maintenance** command to view all station maintenance statistics, as shown below:

    `MOT#`**show cable modem maintenance**

  - Use the **show cable modem maintenance** command to view station maintenance statistics for CMs on a particular cable interface, as shown below:

    `MOT#`**show cable modem** <*x*>/<*y*> **maintenance**

    where:

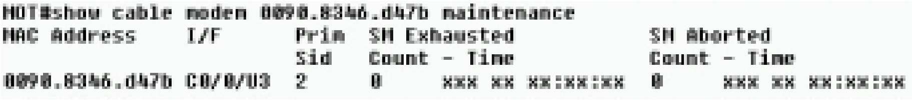
    x is the cable module slot number.

    y is the cable interface number, which is **0**.

  - Use the **show cable modem maintenance** command to view station maintenance statistics for a particular CM, as shown below:

    `MOT#`**show cable modem** <*mac*> **maintenance**

    Figure 6-26 displays station maintance statistics for a CM. The following example demonstrates that there have been no CM station maintance ranging retries or failures:



```
MOT#show cable modem 0090.8346.d47b maintenance
MAC Address     I/F       Prim  SM Exhausted          SM Aborted
                          Sid   Count - Time          Count - Time
0090.8346.d47b C0/0/U3   2     0     xxx xx xx:xx:xx  0     xxx xx xx:xx:xx
```

Figure 6-26 show cable modem maintenance Command Output

- Use the **show cable modem stats** command in Privileged EXEC mode to display IP statistics for the number of unicast bytes that are transmitted and received for each CM MAC address on a cable module, which includes CMs that are off-line, as shown below

  `MOT#`**show cable modem** [<*mac*> | <*0-15*> | <*prefix*>] **stats**

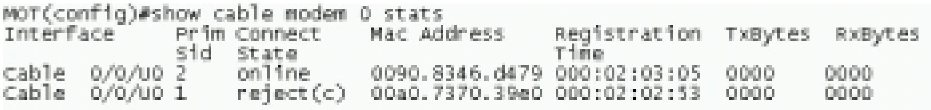
  where:

  mac is the MAC hardware address of the CM.

  prefix is the IP address of the CM.

  0-15 is the slot number of the cable module to which CMs are associated.

  Figure 6-27 displays transmit bytes, receive bytes, and registration CM statistics for a cable module slot.



```
MOT(config)#show cable modem 0 stats
Interface        Prim Connect     Mac Address    Registration  TxBytes  RxBytes
                 Sid  State                      Time
cable  0/0/U0 2    online      0090.8346.d479 000:02:03:05  0000     0000
Cable  0/0/U0 1    reject(c)   00a0.7370.39e0 000:02:02:53  0000     0000
```

Figure 6-27 show cable modem stats Command Output

Displaying Modulation Profiles

A modulation profile contains six burst profiles sent out in a UCD message to configure CM transmit parameters. To display modulation profile group information, use the **show cable modulation-profile** command in Privileged EXEC mode, as shown below:

`MOT#`**show cable modulation-profile**

Table 6-13 describes the fields in the **show cable modulation-profile** command display.

Table 6-13 show cable modulation-profile Fields

Field	Identification
Burst len	Burst length
Diff encode	Indication of diff encode
FEC err corre	Number of corrected Forward Error Correction errors

Table 6-13 show cable modulation-profile Fields

Field	Identification
FEC len	FEC length
Guard time size	Guard time size
Intvl usage code	IUC of upstream transmit burst class
Last code-word	Last codeword shortened
MOD type mod	Upstream modulation type
Preambl length	Length of the preamble
Profile (1-16)	Modulation profile group
Scrambl	Scramble enabled indication
Scrambl seed	Seed of the scrambler

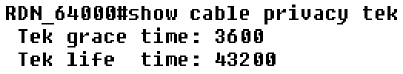
Displaying BPI Configuration Settings

Follow these steps to ensure that the correct Traffic Encryption Key (TEK) and Authorization Key (AK) privacy values are set:

1. Use the **show cable privacy tek** command in Privileged EXEC mode to display TEK grace time and life time values, as shown below:

   MOT#**show cable privacy tek**

   Figure 6-28 displays a sample output from the **show cable privacy tek** command:

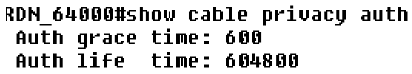
   ```
   RDN_64000#show cable privacy tek
    Tek grace time: 3600
    Tek life  time: 43200
   ```

   Figure 6-28 show cable privacy tek Command Output

2. To display the AK grace time and life time values, use the **show cable privacy auth** command in Privileged EXEC mode, as shown below:

   MOT#**show cable privacy auth**

Figure 6-29 displays a sample output from the **show cable privacy auth** command:



```
RDN_64000#show cable privacy auth
 Auth grace time: 600
 Auth life  time: 604800
```

Figure 6-29 show cable privacy auth Command Output

7

Configuring Routing Policy

Overview

Routing policy can be used to enforce agreements between two or more ISPs concerning the amount and type of traffic that is allowed to pass between them.

Routing policy determines the following:

- Routes to accept from neighboring routers
- Preferences for accepted routes
- Routes to be advertised to neighbors
- Routes to be redistributed into and out of another protocol

Use the following sections to configure routing policy for the BSR:

- Defining Route Maps
- Defining Access Lists and Groups
- Creating Community Lists
- Redistributing Routes
- Applying a Damping Criteria
- Policy-Based Routing
- Gathering Routing Policy Information

Defining Route Maps

Route maps establish the conditions for redistributing routes from one routing protocol to another, and for advertising and learning routes from one router to another. A route map consists of **route-map** commands, match statements that define the conditions that a route must meet, and **set** statements that define the conditions that apply to a route.

Creating a Route Map

To define a route map, use the **route-map** command in Global Configuration mode, as shown in the example below:

```
MOT(config)#route-map <name> [permit | deny] <sequence-number>
```

where:

name is the name that uniquely identifies an instance of a route map; instances with lower sequence numbers are parsed first.

permit specifies perform set operations, if the match conditions are met.

deny specifies deny set operations.

sequence-number identifies an instance of the route map.

Once the route map is created using the **route-map** command, you enter Route Map Configuration mode. Refer to the following sections to define parameters for your route map.

Using Match Statements to Define Routing Conditions

Match statements define the conditions that a route must meet. Each instance may contain multiple match statements. If all match statements in an instance match for a given route, the route meets the conditions of the instance. The order of match statements within an instance is not relevant. If an instance contains no match statements, all routes meet the conditions of the instance, however, they can be denied by an instance with a lower sequence number.

Follow these steps to define the conditions for a route:

1. To match one or more BGP AS-path access lists, use the **match as-path** command in Route Map Configuration mode, as shown in the example below:

   ```
   MOT(config-rmap)#match as-path <as-path-access-list>
   [...<as-path-access-list>]
   ```

   where:

   as-path-access-list is the AS path access list from 1 to 99.

2. To match one or more BGP community lists, use the **match community** command in Route Map Configuration mode, as shown in the example below:

   ```
   MOT(config-rmap)#match community <community-list>
   [...<community-list>]
   ```

3. Match the destination IP address that is permitted by one or more standard or extended access lists with the **match ip address** command. To match destinations with an IP access list, use the **match ip address** command in Route Map Configuration mode, as shown in the example below:

   ```
   MOT(config-rmap)#match ip address <ip-access-list-number>
   [...<ip-access-list-number>]
   ```

4. To match one or more next-hop IP addresses, use the **match ip next-hop** command in Route Map Configuration mode, as shown in the example below:

   ```
   MOT(config-rmap)#match ip next-hop <ip-access-list-number>
   [...<ip-access-list-number>]
   ```

5. To redistribute one or more routes that routers and access servers advertised to the address specified in the access list, use the **match ip route-source** command, in Route-map Configuration mode, as shown in the example below:

   ```
   MOT(config-rmap)#match ip route-source <ip-access-list-number>
   [...<ip-access-list-number>]
   ```

   Use the **no match ip route-source** command to disables route distribution that routers and access servers advertised to the address specified in the access list.

6. To match a routing metric value, use the **match metric** command in Global Configuration mode, as shown in the example below. For BGP this is Multi-Exit Discriminator (MED).

   ```
   MOT(config-rmap)#match metric <metric-value>
   ```

7. To match external route types, use the **match route-type** command in Route Map Configuration mode, as shown in the example below:

   ```
   MOT(config-rmap)#match route-type <external> [type-1 | type-2]
   ```

   where:

   external indicates OSPF routes.

   type 1 matches only type 1 external route (for OSPF).

   type 2 matches only type 2 external route.

   Use the **no match route-type** to disable matches and external route redistribution.

8. To match one or more tag values of the destination protocol and set the rules for routes, use the **match tag** command in Route Map Configuration mode, as shown in the example below:

   `MOT(config-rmap)#`**match tag** <*num:0,4294967295*>
   [...<*num:0,4294967295*>

   where:

   num is a valid value from 0 to 4294967295.

Using Set Statements to Define Routing Conditions

Set statements define the conditions that apply to the route. If a route meets the conditions of an instance, some or all set statements are applied, depending on the usage of the route-map. The order of match statements within an instance is not relevant since either all or none are applied.

If an instance has no set statements and all the match statements match, nothing is set for the route. The route is simply redistributed, advertised, or learned as is (depending on where the route map is applied).

Follow these steps to change attributes of a route.

1. To modify an AS path, use the **set as-path prepend** command in Route Map Configuration mode, as shown in the example below:

   `MOT(config-rmap)#`**set as-path prepend** <*as-number*>
   [...<*as-list-number*>]

   set interface pos, set level, set weight

2. To set the BGP community attribute use the **set community** command in Route Map Configuration mode, as shown in the example below:

   `MOT(config-rmap)#`**set community** <*community-number*>
   [...<*community-number*>]

3. To set the next-hop attribute of a route, use the **set ip next-hop** command in Route Map Configuration mode, as shown in the example below:

   `MOT(config-rmap)#`**set ip next-hop** <*addr*>

4. To set the local preference value, use the **set local-preference** command in Route Map Configuration mode, as shown in the example below:

   ```
   MOT(config-rmap)#set local-preference <pref>
   ```

5. To set the metric, use the **set metric** command in Route Map Configuration mode, as shown in the example below:

   ```
   MOT(config-rmap)#set metric <metric>
   ```

6. To set the metric type, use the **set metric-type** command in Route Map Configuration mode, as shown in the example below:

   ```
   MOT(config-rmap)#set metric-type {internal | external | type-1 | type-2}
   ```

7. To set the BGP origin, use the **set origin** command in Route-map Configuration mode, as shown below:

   ```
   MOT(config-rmap)#set origin {egp | igp | incomplete}
   ```

8. To set a tag value for the destination protocol, use the **set tag** command in Route-map Configuration mode, as shown below:

   ```
   MOT(config-rmap)#set tag <num>
   ```

   where:

   num is an integer between 0 and 294967295.

Example

The configuration example below creates the route map named *locpref*, creates the AS-path access list 1, and applies the route map to a BGP neighbor.

The route map sets the local preference for BGP updates. The route map also uses an AS-path access list to permit any update whose AS-path attribute begins and ends with 400. This sets the local preference on updates coming from AS 400.

```
route-map locpref permit 10
match as-path 1
set local-preference 50
route-map local-pref  permit 20
ip as-path access-list 1 permit ^400$
```

The commands below apply the route map to a BGP neighbor:

```
router bgp 100
```

neighbor 160.20.30.4 route-map locpref in

Defining Access Lists and Groups

An access list is a sequential collection of permit and deny conditions. The BSR tests each condition against conditions in an access list, and supports the following access lists.

Use the following sections to configure access lists and access groups on the BSR:

- Configuring an IP Access List
- Configuring an AS-path Access-list
- Configuring an IP Access Group

Configuring an IP Access List

1. To configure an IP access list, use the **access-list** command in Global Configuration mode, as shown below:

   MOT(config)#**access-list** <*access-list-number*> {**permit** | **deny**} {<*source-address*> <*source-wildcard-bits*> | **any**} {<*destination-address*> <*destination-wildcard-bits*> | **any**}

   where:

   access-list-number is the number of the access list.

   source-address is the source IP address.

   source-wildcard-bits is the wildcard bits of the source address.

   destination-address is the destination IP address.

   destination-wildcard-bits is the wildcard bits of the destination address.

2. Permit and deny conditions in an IP access list apply to IP addresses. To apply an access list to a neighbor router, use the **neighbor distribute-list** command, as shown below:

   MOT(config)#**neighbor distribute-list**

Example

This configuration example filters BGP updates from a BGP neighbor. It configures Access List 4 by specifying its permit and deny conditions. Access list 4 prohibits the propagation of networks specified in the deny statements (10.0.0.0, 162.15.0.0, and 180.10.0.0) and permits all others.

```
access-list 4 deny 10.0.0.0 0.255.255.255 any
access-list 4 deny 162.15.0.0 0.0.255.255 any
access-list 4 deny 180.10.0.0 0.0.255.255 any
access-list 4 permit any
```

Note: All lists have an assumed *deny all* entry as the last statement. If no matches exist, the route or set is denied.

The following commands enable BGP, specify an autonomous system, and apply Access List 4 to a neighbor. The example instructs the router to pass all network information received from BGP neighbor 156.30.10.22 through access list 4.

```
router bgp 100
neighbor 160.25.15.10 distribute-list 4 in
```

Configuring an AS-path Access-list

The permit and deny conditions in an AS-path access list apply to AS numbers. The **neighbor filter-list** command applies an AS-path access list for inbound and outbound updates to a BGP neighbor. The **match AS-path** command adds a match clause to a route map.

To define an AS-path access list, use the **ip as-path access-list** command, as shown below:

```
MOT(config)#ip as-path access-list <access-list-number> {permit | deny}
<path-expression>
```

where:

access-list-number is the access list number.

path-expression is a valid path expression.

Example

This example configures a router with two AS-path access lists. Routes that pass AS-path access-list 1 are sent to one destination. Routes that pass AS-path access-list 2 are accepted from another destination. The commands below specify permit and deny conditions for AS-path access lists.

```
ip as-path access-list 1 permit _200$
ip as-path access-list 1 permit ^100$
ip as-path access-list 2 deny _690$
ip as-path access-list 2 permit .*
```

The commands below enable BGP and specify an autonomous system, define two neighbor peers, assign the AS path list to one of the neighbor BGP peers, and assign a second AS path list to the other neighbor to indicate that outbound routes have the conditions defined in AS-path access-list 2 applied.

```
router bgp 100
neighbor 156.30.10.22 remote-as 200
neighbor 160.25.15.10 remote-as 300
neighbor 156.30.10.22 filter-list 1 out
neighbor 156.30.10.22 filter-list 2 out
```

Configuring an IP Access Group

No access groups defined by default on the BSR. Use the **ip access-group** command in Interface Configuration mode to configure an interface to use an access list.

Note: Use the **no ip access-group** command to delete an access group on an interface.

```
MOT(config-if)#ip access-group {<num:1-199> <num:1300-2699> {in | out}
```

where:

num:1-199 is the standard access list

num:1300-2699 is the extended access list

in incoming packet is processed only if the source-address is in the access-list.

out permits the outgoing packet to be processed only if access-list permits the packet.

Creating Community Lists

You can use the community to control the routing information a BGP speaker accepts, prefers, or distributes to other neighbors. The BGP community attribute passes between peers when they exchange reachability information. You can use the following predefined community attribute keywords with the **set community** command in a route map:

- **no-export**
- **no-advertise**
- **local-as**

Use the **no export** keyword to disallow advertising to EBGP peers. This is useful in a network that uses IBGP heavily but does not want to share its internal routing policies with its EBGP peers. Use the **no-advertise** keyword to prevent communities from being propagated beyond the local router, even to IBGP peers.

Figure 7-1 shows how you can create a route map based on the network shown. The Router Boston sets the value of the local preference attribute based on the value of the community attribute. Any route that has a community attribute of 100 matches community list 1 and has its local preference set to 50. Any route that has a community attribute of 200 matches community list 2 and has its local preference set to 25. All other routes do not have their local preference attributes changed, because all routes are members of the Internet community. All destinations belong to the general Internet community by default.

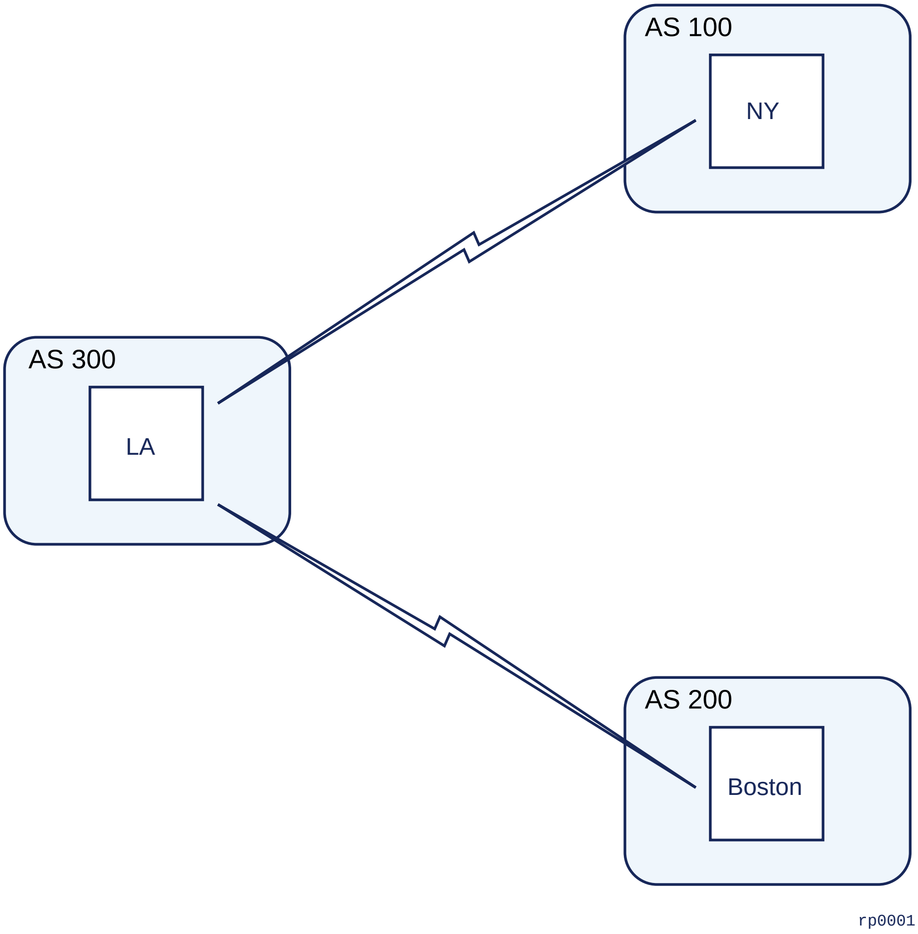


Figure 7-1 Using a Community List

Filtering Routes

To filter routes based on a community list, use the **ip community-list** command, as shown below:

`MOT(config)#`**ip community-list** <*community-list-number*> {**permit** | **deny**} {<*community-number*>} {<*as-community-number*> | **no-export** | **no-advertise** | **local-as** | **internet**}

where:

community-list-number identifies a community list.

community-number is a number that identifies a community.

as-community-number is a number that identifies one or more permit or deny ASs.

Example

This example uses a community list to modify the local preference of routes based on community number. The commands below specify community list 1 to permit routes with community number 100 and community list 2 to permit routes with community number 200.

```
ip community-list 1 permit 100
ip community-list 2 permit 200
```

The first instance of the route map defines the appropriate match and set clauses. The commands below specify route map locpref, instance 10. They set the local preference of the route to 50, if the route is part of the communities defined in Community List 1.

```
route-map locpref permit 10
match community 1
set local preference 50
```

The next commands define the second instance of the route map, route-map locpref, instance 20. The commands also set the local preference of the route to 25, if the route is part of the communities defined in Community List 2.

```
route-map locpref  permit 20
match community 2
set local preference 25
```

The final commands enable BGP and specify the AS for Router Boston in AS 200, specify the AS of the BGP neighbor (Router A) to which the route map applies, and apply the route map localpref for all incoming routes.

```
router bgp 200
neighbor 160.30.21.10 remote-as 300
neighbor 160.30.21.10 route-map locpref in
```

Redistributing Routes

You can advertise networks by redistributing routes learned from one routing protocol into another. Figure 7-2 shows Router New York redistributes the routes learned via OSPF protocol from Routers Boston and LA into BGP.

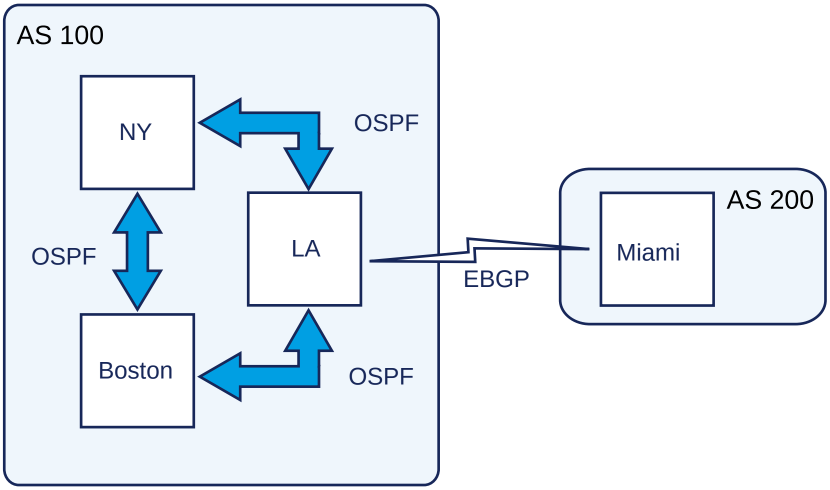


Figure 7-2 Redistributing Routes Learned from OSPF

To redistribute routes from one routing protocol to another, use the **redistribute** command, as shown below:

```
MOT(config)#redistribute <protocol> {bgp | ospf | static [ip] connected |
rip} [tag <tag value>] [route-map <map-tag>] [match {internal | external 1 |
external 2}][metric <metric-value>] [metric-type <type-value>] [weight
<weight>] [subnets <subnets>}]
```

where:

protocol is a protocol type, such as OSPF, RIP, BGP, STATIC, CONNECTED.

static indicates IP or RIP static routes.

connected indicates established routes as result of IP interface.

rip indicates RIP.

tag is a unique name for routing process.

route-map indicates current routing protocol.

match internal indicates routes that are internal to an AS.

external 1 indicates routes that are external to an AS, but are imported into OSPF as either Type 1 or Type 2 external route.

external 2 indicates routes that are external to an AS, but are imported into OSPF as a Type 2 external route.

metric indicates the source protocol from which routes are being redistributed; valid values are BGP, connected, RIP OSPF and static.

Example

This example redistributes routes learned from OSPF into BGP. The commands enable BPP, specify the autonomous system 100, and specify OSPF as the protocol type for the redistribution.

```
router bgp 100
redistribute ospf
```

Applying a Damping Criteria

Route flapping occurs when a link constantly fluctuates between being available and unavailable. Every time a link changes its availability, the upstream neighbor sends an update message to all its neighbors. These routes are advertised globally. This process continues until the underlying problem is fixed.

Route flap damping minimizes instability caused by route flapping. Use policy-based route flap damping to apply the following damping criteria to specific routes:

- *half-life* —half-life period in minutes in the range 1 - 45. The default is 10. When a BGP route has been assigned a penalty, the penalty is decreased by a half after each half-life period (which is 15 minutes by default). Each time a route flaps, the router configured for route flap damping assigns the route a penalty. Penalties are cumulative. BGP stores the penalty for all reachable and unreachable routes that have experienced recent flaps.
- *reuse* — reuse limit in the range 1 - 20000. The default is 750. As the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. The route is added back to the BGP route table and used for forwarding. The process of unsuppressing routers occurs in 10-second increments. Every 10 seconds the router determines which routes are unsuppressed and advertises them globally.
- *suppress* — the suppress limit in the range 1 - 20000. The default is 2000. A route is suppressed when its penalty exceeds this limit.
- *max-suppress-time* — the maximum suppression time in minutes in the range 1 - 255. The default is four times the half-life. This value is the maximum amount of time a route can be suppressed.

Policy-Based Routing

Policy-based routing routes network traffic by establishing protocol-independent data paths. Policy-based routing provides a mechanism for forwarding data based on policies defined by a network administrator. Policy-based routing also provides a mechanism to mark packets so that certain kinds of traffic receive differentiated, preferential service. Instead of routing by the destination address, policy-based routing allows network administrators to implement routing policies that allow or deny data paths based on the identity of a particular end system or a particular application.

Policy-based routing is applied to incoming packets. All packets received on an interface with policy-based routing enabled are considered for policy-based routing. The router passes the packets through a route map. Based on the information defined in the route map, packets are forwarded to the appropriate next hop. Route maps contain a combination of match and set commands. The match commands define the criteria for whether or not packets meet a particular policy. The set commands define how the packets should be routed if they have met the match criteria.

Note: We implement policy-based routing in hardware on the BSR family of routers. Implementing policy-based routing in hardware instead of software causes no degradation in device performance.

This section discusses the following tasks:

- Configuring a Policy-Based Route Map
- Enabling Policy-Based Routing on an Interface
- Enabling Local Policy-Based Routing on a Router

Configuring a Policy-Based Route Map

To enable policy-based routing, you must identify or create a route map to use for policy-based routing. The route map specifies the match and set parameter and the resulting action if all of these parameters are met. You can define additional route map parameters to be used for policy-based routing. These parameters can be added to an existing route map or included in a new route map. See Creating a Route Map on page 7-1 for more information.

Defining Match Criteria

The following match commands can apply when configuring a policy-based route map:

match ip address

match ip next-hop

match ip route-source

Refer to Using Match Statements to Define Routing Conditions on page 7-2 for more information on defining a match criteria using these commands.

Note: If no match parameters are specified in the route map, then all incoming packets are considered for policy-based routing.

Defining the Route

Set commands are used to define a route in the route map to be used for policy-based routing. If an interface or next-hop specified by these commands is unreachable, then destination-based routing is used by default. You must specify at least one of the following **set** commands to define a route:

set interface

set ip next-hop

set default interface

set ip default next-hop

set ip diff-serv

set interface pos defines a list of POS interfaces through which the packets can be routed. If more than one POS interface is specified, then the first interface that is reachable is used for forwarding the packets. Policy-based routing is only supported over point-to-point links. To set the forwarding interface, use the **set interface pos** command in Route Map Configuration mode, as shown in the example below:

```
MOT(config-rmap)#set interface pos <slot/port>
```

To force packets to be dropped and not routed with the default destination-based routing process, use the **set interface null0** command to add "null0" as the last entry in the interface list.

```
MOT(config-rmap)#set interface null0
```

set ip next-hop defines a list of IP addresses that specify the next hop router in the path toward the destination to which the packets should be forwarded. The first IP address associated with a reachable interface is used to route the packets. To set the next-hop attribute of a route, use the **set ip next-hop** command in Route Map Configuration mode, as shown in the example below:

```
MOT(config-rmap)#set ip next-hop <addr>
```

set default interface defines a list of default interfaces. If there is no explicit route available to the destination address of the packet being considered for policy routing, then it is sent to the first reachable interface in the list of specified default interfaces. To set the default output interface, use the **set default interface** command in Route Map Configuration mode, as shown in the example below:

```
MOT(config-rmap)#set default interface <addr>
```

set ip default next-hop defines a list of default next hop IP addresses. Routing to the interface or the next hop specified by this set command occursd only if there is no explicit route for the destination address of the packet in the routing table. To set the next-hop of a route, use the **set ip default next-hop** command in Route Map Configuration mode, as shown in the example below:

```
MOT(config-rmap)#set ip default next-hop <addr>
```

Note: The **set** commands, described previously, will be applied in the following order:

set interface

set ip next-hop

set default interface

set ip default next-hop

The first applicable set command in the list will be applied. For example, if the interface specified by "set interface" is unreachable, then "set ip next-hop" will be applied.

set ip diff-serv defines a value that sets the precedence in the IP packets. To change the differentiated service value, use the **set ip diff-serv** command in Route Map Configuration mode, as shown in the example below:

```
MOT(config-rmap)#set ip diff-serv <0-63>
```

The precedence setting determines which packets will be given transmission priority. When packets with a precedence value are received by another router, the packets are ordered for transmission according to the precedence set. A higher precedence value indicates a higher priority. Refer to Table 7-1 for pre-defined precedence values.

Enabling Policy-Based Routing on an Interface

To enable policy-based routing on an interface and to indicate which route map the router should use, use the **ip policy route-map** command in Interface Configuration mode, as shown in the example below:

```
MOT(config-if)# ip policy route-map <route map name>
```

All packets arriving on the interface will be subject to policy-based routing. Use the **no ip policy route-map** <*route map name*> command to disable policy-based routing on the interface.

Enabling Local Policy-Based Routing on a Router

Local policy-based routing is applied to all packets originating from this router. To enable local policy-based routing on a router and to indicate which route map the router should use, use the **ip local policy route-map** command in Global Configuration mode, as shown in the example below:

```
MOT(config)# ip local policy route-map <route map name>
```

All packets originating on the router will now be subject to local policy-based routing. Use the **no ip local policy route-map** <*route map name*> command to disable local policy-based routing on the router. Use the **show ip local policy** command to display the route map used for local policy-based routing, if one exists.

Gathering Routing Policy Information

Use the following **show** commands to monitor routing policies:

- **show ip access-list**
- **show ip as-path access-list**

- **show ip bgp**
- **show ip bgp dampened-paths**
- **show ip community-list**
- **show ip interface**
- **show ip redistribute**
- **show ip route**
- **show route-map**

1. To display an access list or all access lists, use the **show ip access-list** from any mode, as shown below. The resulting display includes the instances of each access list.

`MOT#`**show ip access-list** [<*number*>]

where:

number is the access list number or numbers.

2. To display the configured AS path access lists, use the **show ip as-path access-list** from Global Configuration mode, as shown below. The resulting display includes the instances of each AS path access list.

`MOT#`**show ip as-path access-list** [<*number*>]

where:

number is the access list number or numbers.

3. To display the entries in the BGP routing table, use the **show ip bgp** from EXEC mode, as shown below.

`MOT(config)#` **show ip bgp** [**network**] [**network-mask**] [**longer-prefixes**]

where:

network is the number of the network in the BGP routing table.

network-mask displays all BGP routes matching the address/mask pair.

longer-prefixes displays route and more specific routes.

4. To display the entries in the routing table, use the **show ip bgp dampened-paths** from any mode, as shown below. The resulting display includes the instances of each access list.

`MOT#` **show ip bgp dampened-paths**[<*number*>]

5. To display the configured community access list, use the **show ip community-list** from Global Configuration mode, as shown below.

   `MOT(config)#`**show ip community-list** [<*number*>]

   where:

   number is the access list number; valid entries are 1 to 199.

6. To display interface multicast information, use the **show ip interface** from EXEC mode, as shown below.

   `MOT#`**show ip interface** [<*number*>]

   show ip interface [**brief**] [**ethernet** <*slot*> /<*port*> | **cable** <*slot*> /<*port*> | **pos** <*slot*> /<*port*> | **loopback** <*num:1,16*> | **tunnel** <*num:0,255*>]

   where:

   brief indicates display a brief summary of IP status and configuration.

   ethernet *slot* / *port* indicates the Ethernet interface slot and port numbers.

   cable *slot* / *port* indicates the cable slot and port numbers.

   pos *num:0,255* indicates the Packet over SONET (POS) slot and port.

   loopback *num:1,16* indicates the loopback interface.

   tunnel *num:0,255* indicates the tunnel interface.

7. To display the routing table entries, use the **show ip route** from Privileged EXEC or Global Configuration mode, as shown below.

   `MOT#`**show ip route** [<*hostname*> | **isis** | **bgp** | **connected** | **ospf** | **rip** | **static** | <*prefix*> [*mask*]]

   where:

   hostname is the DNS host name.

   isis displays the Intermediate-system to Intermediate-system routing (IS-IS) protocol routes.

   bgp display BGP details.

   connected displays connected routes.

ospf displays OSPF protocol transmitting the route.

rip displays RIP protocol transmitting the route.

static displays static routes.

summary displays a summary of all routes.

prefix indicates display IP address; route address.

mask indicates display subnet mask.

8. To display the routing protocols that are redistributed to other routing domains, use the **show ip redistribute** from any mode, as shown below:

`MOT#`**show ip redistribute** [**bgp** | **ospf** | **rip**]

where:

bgp displays routing domains redistributed into BGP.

ospf displays routing domains redistributed into OSPF.

rip displays routing domains redistributed into RIP.

9. To display the configured route maps, use the **show route-map** command from any mode, as shown below. The display includes the instances of each access list.

`MOT#`**show route-map** [*rmap-name*]

8

Configuring IP Multicast Routing

Overview

The IP multicast routing environment allows a host to send packets to a group of hosts called group members. Multicast packets delivered to group members are identified by a single multicast group address and use best-effort reliability.

Hosts can be both senders and receivers. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message. A multicast address is chosen for the receivers in a multicast group. Senders use that address as the destination address of a datagram to reach all members of the group.

Multicast group membership is active; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

How active a multicast group is and what members it has can vary from group to group and from time to time, and a multicast group can be active for long or brief time periods. Group membership can change constantly and have inactive members.

Multicast routing protocols, such as Protocol-Independent Multicast (PIM) and Distance Vector Multicast Routing Protocol (DVMRP) are used to maintain forwarding tables in order to forward multicast datagrams between routers on the network. The BSR uses the Internet Group Management Protocol (IGMP) on a specified interface to learn whether members of a group are present on their directly attached subnetworks. Hosts join multicast groups by sending IGMP report messages.

Use the following sections in this chapter to configure IP multicast routing or display IP multicast routing information on the BSR:

- Enabling IP Multicast Routing on the BSR
- Configuring PIM
- Configuring DVMRP
- Configuring IGMP on an Interface
- Managing IP Multicast Routing on the BSR
- Gathering IP Multicast Information

Enabling IP Multicast Routing on the BSR

IP multicast routing allows the BSR to forward IP multicast packets over the network. Use the **ip multicast-routing** command in Global Configuration mode to enable IP multicast routing on the BSR, as shown below:

```
MOT(config)#ip multicast-routing
```

Configuring PIM

Use the following sections to configure PIM on the BSR:

- About PIM
- Enabling PIM
- Delaying Shortest Path Tree Usage for Better Throughput
- Defining the PIM Domain Border
- Configuring Candidate BSRs
- Configuring Candidate RPs
- Modifying the PIM Router-Query Message Interval

About PIM

Protocol-Independent Multicast (PIM) is used to efficiently route to multicast groups that might span wide-area and inter-domain internetworks. It is referred to as “protocol independent” because it is not dependent on any particular unicast routing protocol.

PIM is IP routing protocol independent because it can use any unicast routing information to forward multicast traffic. Even though PIM is a multicast routing protocol, it uses the unicast routing table to perform Reverse Path Forwarding (RPF), instead of creating an independent multicast routing table. However, unlike most unicast routing protocols, PIM does not send and receive multicast routing updates between routers.

The BSR supports PIM in sparse mode. In sparse mode, routers must join and leave multicast groups explicitly. Upstream routers do not forward multicast traffic out a PIM interface unless the router has sent an explicit request called a join message to receive multicast traffic from a downstream router or if group members are directly connected to the interface. When a host joins a multicast group, its first hop router sends a join message upstream to the rendezvous point (RP) for the group. The RP serves as the root of the shared multicast delivery tree and is responsible for forwarding multicast data from different sources toward the receivers. PIM sparse-mode routers periodically send join messages toward the RP to join a shared tree and directly toward the source if they prefer to join the source tree. The routers also send periodic prune messages to the RP when they move from the shared tree onto the source-based tree. At least one Bootstrap Router (BSR) and one RP needs to be configured in a PIM domain.

Enabling PIM

Follow these steps to configure general PIM parameters:

1. Use the **router pim** command in Global Configuration mode to enter Router Configuration mode, as shown below:

   ```
   MOT(config)#router pim
   ```

2. Use the **network** command in Router Configuration mode to define the network IP address and subnet mask for the PIM network:

   ```
   MOT(config-pim)#network <ip-address> <wild-card>
   ```

   where:

   ip-address is the IP address of the PIM network.

   wild-card is the wild-card mask for the PIM network.

Delaying Shortest Path Tree Usage for Better Throughput

You can control the data threshold rate to delay when the PIM rendezvous point (RP) switches to the Shortest Path Tree (SPT) in order to enhance throughput on your multicast network. The SPT threshold determines when the RP (shared tree) can join the SPT (source tree) for a specified multicast group. If the RP sends at a rate greater than or equal to the specified *kbps* rate, the last-hop router or RP router triggers a PIM Join message to the PIM source router to construct an SPT.

- If you want to specify the multicast traffic threshold that must be reached on the RP router before the multicast traffic is switched over to the SPT, use the **ip pim spt-threshold rp** command in Global Configuration mode, as shown below:

  ```
  MOT(config)#ip pim spt-threshold rp <n> [infinity]
  ```

  where:

  n is the multicast traffic rate in kilobytes per second (kbps).

Note: The default setting for the **ip pim spt-threshold rp** command is 0 kbps, which allows the RP to join the SPT immediately.

infinity indicates that the RP is always used.

- If you want to specify the multicast traffic threshold that must be reached on the last-hop router before multicast traffic is switched over to the SP, use the **ip pim spt-threshold lasthop** command in Global Configuration mode, as shown below:

  ```
  MOT(config)#ip pim spt-threshold lasthop <n> [infinity]
  ```

  where:

  n is the multicast traffic rate in kilobytes per second (kbps).

Note: The default setting for the **ip pim spt-threshold lasthop** command is 1024 kbps, which allows the RP to join the SPT when the 1024 Kbps threshold is reached.

infinity indicates that the RP is always used.

Defining the PIM Domain Border

A border can be configured for the PIM domain, so that bootstrap messages do not cross the border in either direction. Creating a border allows different Bootstrap Routers (BSRs) to be elected on both sides of the PIM border.

Use the **ip pim border** command in Interface Configuration mode to configure a PIM domain boundary on the interface of a border router peering with one or more neighbors outside the PIM domain, as shown below:

```
MOT(config-if)#ip pim border
```

Configuring Candidate BSRs

One or more Bootstrap Routers (BSRs) can be configured to serve as candidates in a PIM domain to avoid a single point of failure. Candidate BSRs should be configured on the backbone portion of the network to help improve the efficiency of the multicast network.

A BSR is elected among the candidate BSRs automatically by using bootstrap messages to determine which BSR has the highest priority. This router then announces to all PIM routers in the PIM domain that it is the BSR.

Select from the following options to configure candidate BSRs:

- If you want to configure the router to be a candidate BSR with the default hash-mask length of 30 bits for an RP selection, use the **ip pim bsr-candidate** command in Interface Configuration mode, as shown below:

  ```
  MOT(config-if)#ip pim bsr-candidate
  ```

- If the default hash mask is used for the candidate BSR, the router takes the first rendezvous point (RP) address from the local RP-mapping cache.

If you want to configure the router to be a candidate BSR and adjust the hash mask length value in order to avoid having two RPs for the same multicast group, use the **ip pim bsr-candidate** command in Interface Configuration mode, as shown below:

Note: It is recommended (but not required) that the hash-mask length be the same across all candidate BSRs.

```
MOT(config-if)#ip pim bsr-candidate <n>
```

where:

n is the hash mask length from 0 to 32 bits.

- If you want to configure another interface on a router to be a candidate BSR, use the **ip pim bsr-candidate ip-address** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#ip pim bsr-candidate ip-address <ip-address>
```

where:

ip-address is the IP address of another interface on the router that is designated as a BSR candidate.

- If you want to configure another interface on the router to be a candidate BSR and adjust the hash mask length value in order to avoid having two RPs for the same multicast group, use the **ip pim bsr-candidate ip-address** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#ip pim bsr-candidate ip-address <ip-address> <n>
```

where:

ip-address is the IP address of another interface on the router that is designated as a BSR candidate.

n is the hash mask length from 0 to 32 bits.

Configuring Candidate RPs

One or more rendezvous points (RPs) can be configured to serve as candidates in a PIM domain to avoid a single point of failure. Candidate RPs should be configured on the backbone portion of the network to help improve the efficiency of the multicast network.

Candidate RPs send candidate RP advertisements to the bootstrap router (BSR) and the BSR then distributes all RP information to the PIM domain. Each router determines which RP has the highest priority and whcih RP to use for a multicast group range. An RP can serve the entire IP multicast address space or a portion of it.

Select from the following options to configure one or more candidate RPs on the PIM domain in Interface Configuration mode:

- If you want a single RP candidate to cover all groups on the PIM domain, use the **ip pim rp-candidate** command, as shown below:

  ```
  MOT(config-if)#ip pim rp-candidate
  ```

- Use the **ip pim rp-candidate group-list** command to configure a specific group range for the RP configured on this interface, as shown below:

Note: If you are configuring routers from other vendors as candidate RPs, ensure that they support PIM Version 2.

  ```
  MOT(config-if)#ip pim rp-candidate group-list <n>
  ```

  where:

  n is the access list reference number from 1 to 99 for group prefixes.

- The default interval, for an RP advertisement is 60 seconds. If you want to change the interval in which an RP candidate is selected, use the **ip pim rp-candidate interval** command, as shown below:

  ```
  MOT(config-if)#ip pim rp-candidate interval <n>
  ```

  where:

n is the interval from 1 to 200 seconds.

- If you want to configure a candidate RP that is associated with this router, use the **ip pim rp-candidate ip-address** command, as shown below:

  ```
  MOT(config-if)#ip pim rp-candidate ip-address <ip-address>
  ```

  where:

  ip-address is an IP address of the candidate RP.

- The RP candidate priority is 0 by default. If you want to use the RP candidate priority for the router, use the **ip pim rp-candidate priority** command, as shown below:

  ```
  MOT(config-if)#ip pim rp-candidate priority <n>
  ```

  where:

  n is the assigned priority of the candidate RP from 0 to 255.

Modifying the PIM Router-Query Message Interval

Router-query messages are used to elect a PIM designated router. The designated router is responsible for sending PIM join and PIM register packets. By default, multicast routers send PIM router-query messages every 60 seconds. To modify this interval, use the following command in interface configuration mode:

```
MOT(config-if)#ip pim query-interval <n>
```

where:

n is the number of seconds from 0 to 65535 that multicast routers send PIM router-query messages

Configuring DVMRP

Use the following sections to learn how configure the Distance Vector Multicast Routing Protocol (DVMRP) on the BSR:

About DVMRP

The Distance Vector Multicast Routing Protocol (DVMRP) is a distance-vector multicast routing protocol that delivers connectionless data to a group of hosts across an internetwork. DVMRP is designed to be used as an interior gateway protocol (IGP) within a multicast domain.

DVMRP is often referred to as a "flood and prune" protocol. DVMRP dynamically creates IP multicast delivery trees by using Reverse Path Forwarding (RPF) to forward multicast traffic away from the source to downstream interfaces. RPF uses the DVMRP routing table to determine the upstream and downstream neighbors. The source sends traffic to an arbitrary group of hosts that are represented by a multicast group address. The multicast router must determine which direction is upstream (towards the source) and which direction (or directions) is downstream. If there are multiple downstream paths, the router replicates the packet and forwards it to the appropriate downstream paths (which may not be all paths). The router forwards a multicast packet once it is received on the upstream interface.

These methods allow the formation of shortest-path trees, which are used to reach all group members from each network source of multicast traffic.

DVMRP routers dynamically discover their neighbors by sending neighbor probe messages periodically to an IP multicast group address that is reserved for all DVMRP routers.

Configuring DVMRP Routing Information

Use the following sections to configure DVMRP routing information on the BSR:

- Enabling DVMRP on the Router
- Configuring the DVMRP Route Expiration Threshold
- Configuring the DVMRP Route Reporting Threshold
- Limiting the Number of DVMRP Routes
- Setting the DVMRP Prune Lifetime Value

Enabling DVMRP on the Router

Follow these steps to configure general DVMRP parameters:

1. Use the **router dvmrp** command in Router Configuration mode to enter Router Configuration mode from Global Configuration mode, as shown below:

   MOT(config)#**router dvmrp**

2. Use the **network** command in Router Configuration mode to define the network IP address and subnet mask for the DVMRP network:

   MOT(config-dvmrp)#**network** <*ip-address*> <*wild-card*>

   where:

   ip-address is the IP address of the DVMRP network.

   wild-card is the wild-card mask for the DVMRP network.

Configuring the DVMRP Route Expiration Threshold

Use the **route expire-interval** command in Router Configuration mode to set the DVMRP route expiration interval, as shown below:

MOT(config-dvmrp)#**route expire-interval** <*n*>

where:

n is the route expiration interval from 5 to 3600 seconds.

Configuring the DVMRP Route Reporting Threshold

Use the **route report-interval** command in Router Configuration mode to set how often DVMRP routes are reported, as shown below:

MOT(config-dvmrp)#**route report-interval**

where:

n is the route reporting interval from 5 to 3600 seconds.

Limiting the Number of DVMRP Routes

Seven-thousand DVMRP route reports can be advertised on the BSR by default. Use the **route-limit** command in Router Configuration mode to change the number of DVMRP route reports that can be advertised per interval on the BSR, as shown below:

MOT(config-dvmrp)#**route-limit** <*n*>

where:

n is the number of DVMRP route reports.

Setting the DVMRP Prune Lifetime Value

DVMRP uses a basic multicast model to build a parent-child database. This database is used to create a forwarding tree that originates at the source where multicast packets are generated. Multicast packets are initially flooded down the forwarding tree making parent-child links. If there are redundant paths (parent-child links) on the forwarding tree, packets are not forwarded along those paths. Forwarding occurs until prune messages are received on the forwarding tree, which further holds back multicast packet broadcasts. Pruning is initiated from the leaf router, where there are no multicast members.

The prune lifetime is the amount of time a prune state is maintained on a router before it times out. Use the **prune lifetime** command in Router Configuration mode to set the life-time value for DVMRP prune messages that are received on parent-child links to improve throughput, as shown below:

MOT(config-dvmrp)#**prune lifetime** <*n*>

where:

n is the prune lifetime value from 5 to 7200 seconds.

Configuring DVMRP on a Routing Interface

Use the following sections to configure DVMRP on the BSR routing interface:

- Filtering Incoming DVMRP Reports
- Filtering Outgoing DVMRP Routing Reports
- Distributing the Default DVMRP Network to Neighbors
- Adding a Metric Offset to the DVMRP Route
- Setting the DVMRP Neighbor Time-out Interval
- Delaying DVRMP Reports
- Setting the DVMRP Probe Interval

- Rejecting a DVMRP Non-pruning Neighbor
- Configuring a DVMRP Summary Address

Filtering Incoming DVMRP Reports

Use the **ip dvmrp accept-filter** command in Interface Configuration mode to block an address range from being forwarded by filtering incoming DVMRP reports, as shown below:

```
MOT(config-if)#ip dvmrp accept-filter <n>
```

where:

n is the accept filter number from 0 to 99.

Filtering Outgoing DVMRP Routing Reports

Use the **ip dvmrp out-report-filter** command in Interface Configuration mode to stop advertising a route originating on this interface, by filtering the outgoing DVMRP report, as shown below:

```
MOT(config-if)#ip dvmrp out-report-filter <n>
```

where:

n is the out report filter number from 1 to 99.

Distributing the Default DVMRP Network to Neighbors

The default DVMRP network is 0.0.0.0. Use the following options to distribute the default DVMRP network to neighboring routers and the multicast backbone (MBONE) or to neighboring routers only for a DVMRP Version 3.6 device:

- If you want the default DVMRP route distributed to the MBONE and neighboring routers, use the **ip dvmrp default-information originate** command in Interface Configuration mode, as shown below:

  ```
  MOT(config-if)#ip dvmrp default-information originate
  ```

- If you want the default DVMRP route distributed to the neighbor only, use the **ip dvmrp default-information originate only** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#ip dvmrp default-information originate only
```

Adding a Metric Offset to the DVMRP Route

The DVMRP metric is the same as a hop-count. The BSR uses increments of 1 to adjust the metric of a DVMRP route advertised in incoming DVMRP reports by default. The DVMRP route metric can be changed to assist or sustain a particular route. The **ip dvmrp metric-offset** command is used to influence which routes are used or preferred. It is also associated with the unicast route being reported for each source network that is reported. The metric is the increment or hop count used to measure the span between the router originating the report and the source network. The source network becomes unreachable when it reaches a metric of 32.

Use the following options to add a metric offset to both incoming and outgoing routing reports:

- Use the **ip dvmrp metric-offset in** command in Interface Configuration mode to add an increment to the incoming DVMRP reports, as shown below:

  ```
  MOT(config-if)#ip dvmrp metric-offset in <n>
  ```

  where:

  n is the increment number from 0 to 31.

Note: The default value for **in** is 1.

- Use the **ip dvmrp metric-offset out** command in Interface Configuration mode to add an increment to outgoing DVMRP reports, as shown below:

  where:

n is the increment number from 0 to 31.

Note: The default value for **out** is 0.

Setting the DVMRP Neighbor Time-out Interval

The DVMRP neighbor time-out interval is the amount of time allowed before a neighbor is removed from the DVMRP neighbor table, if the neighbor does not send a probe (query) or report. Use the **ip dvmrp neighbor-timeout** command in Interface Configuration mode to set the DVMRP neighbor time-out interval, as shown below:

```
MOT(config-if)#ip dvmrp neighbor-timeout <n>
```

where:

n is the time interval from 5 to 3600 seconds.

Delaying DVRMP Reports

The inter-packet delay of a DVMRP report is the time that elapses between transmissions of sets of packets that constitute a report. The number of packets in the set is determined by the *burst* value, which defaults to 2 packets.

Use the **ip dvmrp output-report-delay** command in Interface Configuration mode to configure the delay between each DVMRP route report burst, as shown below:

```
MOT(config-if)#ip dvmrp output-report-delay <n>
```

where:

n is the amount of delay from 1 to 10 seconds.

Use the **ip dvmrp output-report-delay** command in Interface Configuration mode to configure the delay between each DVMRP route report burst and the number DVMRP route reports contained in each burst, as shown below:

```
MOT(config-if)#ip dvmrp output-report-delay <n> <reports>
```

where:

n is the amount of delay from 1 to 10 seconds.

reports is the number of route reports from 1 to 100.

Setting the DVMRP Probe Interval

The DVMRP probe interval is configured to send queries to neighboring multicast routers for DVMRP multicast routes. Use the **ip dvmrp probe-interval** command in Interface Configuration mode to set the DVMRP probe interval, as shown below:

```
MOT(config-if)#ip dvmrp probe-interval <n>
```

where:

n is the probe interval from 5 to 3600 seconds.

Rejecting a DVMRP Non-pruning Neighbor

The BSR accepts all DVMRP neighbors as peers by default. Routers that have old versions of DVMRP, that cannot prune, waste bandwidth by receiving forwarded packets unnecessarily.

If there is a non-pruning version of DVMRP running on a neighbor, use the **ip dvmrp reject-non-pruners** command in Interface Configuration mode to prevent this neighbor from receiving forwarded packets, as shown below:

Note: The **ip dvmrp reject-non-pruners** command prevents peering with neighbors only. If there are any non-pruning routers multiple hops away (downstream toward potential receivers) that are not rejected, then a non-pruning DVMRP network may still exist.

```
MOT(config-if)#ip dvmrp reject-non-pruners
```

Configuring a DVMRP Summary Address

Use the **ip dvmrp summary-address** command in Interface Configuration mode to create a summary address for a group of DVMRP routes, as shown below.

```
MOT(config-if)#ip dvmrp summary-address <ip-address> <subnetwork>
```

Note: One or more specific routes must exist in the unicast routing table before a summary address is advertised.

where:

ip is the DVMRP IP summary address.

subnetwork is the DVMRP subnetwork mask.

Configuring IGMP on an Interface

Use the following sections to learn how configure the Internet Group Management Protocol (IGMP) on an interface:

- About IGMP
- Enabling IGMP
- Controlling Access to IP Multicast Groups
- Changing the IGMP Version
- Modifying the IGMP Host-Query Message Interval
- Specifying the IGMP Querier Time-out Interval
- Changing the Maximum Query Response Time
- Configuring the BSR as a Static Multicast Group Member

About IGMP

IGMP manages the membership of hosts and routers in multicast groups. IP hosts use IGMP to report their multicast group memberships to any immediately neighboring multicast routers. Multicast routers use IGMP to learn, for each of their attached physical networks, which groups have members.

IGMP is an integral part of IP and must be enabled on all routers and hosts that want to receive IP multicasts.

For each attached network, a multicast router can be either a querier or a non-querier. The querier router periodically sends general query messages to solicit group membership information. Hosts on the network that are members of a multicast group send report messages. When a host leaves a group, it sends a leave group message.

IP hosts use IGMP to report their group membership to directly connected multicast routers. IGMP uses class D IP addresses for its group addresses that can range from 225.0.0.0 to 239.255.255.255.

The following multicast addressing rules apply to IGMP:

- The 224.0.0.0 IP address is guaranteed not to be assigned to any group.
- The address 224.0.0.1 is assigned to all systems on a subnetwork.
- The address 224.0.0.2 is assigned to all routers on a subnetwork.

Enabling IGMP

IGMP is enabled on all interfaces on which DVMRP or PIM is configured by default.

Controlling Access to IP Multicast Groups

The BSR learns about multicast group members that are connected to local networks by sending IGMP host-query messages. The BSR then forwards all packets addressed to the multicast group to these group members. IP multicast group access is determined by associating the IGMP access group to an access list. Refer to Chapter 7 for more information on configuring access lists.

Follow these steps to configure to configure access for IP multicast groups:

1. Use the **interface** command in Global Configuration mode to enter the IGMP interface, as shown below:

   `MOT(config-if)#`**interface** {**cable** | **ethernet** | **pos**} *<slot>*/*<interface>*

   where:

   cable is the cable interface.

   ethernet is the Ethernet interface.

   pos is the Packet over SONET interface.

slot is the module slot number.

interface is the interface number.

2. Use the **ip igmp access-group** command in Interface Configuration mode to filter multicast groups that are permitted on the interface:

   `MOT(config-if)#`**ip igmp access-group** <*n*>

   where:

   n is the access list-number from 1 to 99.

Changing the IGMP Version

The BSR uses IGMP Version 2 by default, which allows the IGMP query time-out and the maximum query response time features. All hosts connected to an interface must support the same version of IGMP. If hosts connected to a particular interface only support IGMP Version 1, IGMP Version 1 must be selected for the interface.

Use the **ip igmp version 1** command in Interface Configuration mode to change the version of IGMP the BSR to IGMP Version 1, as shown below:

`MOT(config-if)#`**ip igmp version 1**

If you need to return to IGMP Version 2, use the **ip igmp version 2** command in Interface Configuration mode to return the version of IGMP on the BSR to IGMP Version 2, as shown below:

`MOT(config-if)#`**ip igmp version 2**

Modifying the IGMP Host-Query Message Interval

When the BSR is configured for multicast routing, it can periodically send IGMP host-query messages to connected networks in order to refresh multicast group member information or discover new multicast group members. These messages are sent to the all-systems group address of 224.0.0.1 with a time-to-live (TTL) of 1.

The BSR uses the highest IP address for its multicast network and is responsible for sending IGMP host-query messages to all hosts on the subnetwork. The BSR sends IGMP host-query messages every 60 seconds to keep the IGMP overhead low on hosts and networks connected to the BSR by default.

Use the **ip igmp query-interval** command in Interface Configuration mode to change the IGMP host-query message interval, as shown below:

```
MOT(config-if)#ip igmp query-interval <n>
```

where:

n is the time interval from 1 to 3600 seconds that the BSR sends IGMP host-query messages to connected networks.

Specifying the IGMP Querier Time-out Interval

IGMP Version 2 allows a time interval to be specified for a querier to surrender its function as the querier for the interface over to the BSR. The BSR then becomes the querier for the interface.

Use the **ip igmp querier-timeout** command in Interface Configuration mode to specify the IGMP querier timeout interval, as shown below:

Note: The time interval should be twice the number of seconds as the IGMP query interval.

```
MOT(config-if)#ip igmp querier-timeout <n>
```

where:

n is the time interval from 60 to 3600 seconds.

Changing the Maximum Query Response Time

IGMP Version 2 allows the maximum query response time advertised in IGMP queries to be changed. The maximum query response time allows a router to quickly detect that there are no more directly connected group members on a subnetwork. Decreasing the value allows the BSR to prune groups quickly.

The query response interval is the maximum amount of time that can elapse between when the querier router sends a host-query message and when it receives a response from a host.

Use the **ip igmp query-max-response-time** command to change the query response interval to change the burstiness of IGMP messages on the subnetwork. The query response interval is 10 seconds by default.

```
MOT(config-if)#ip igmp query-max-response-time <n>
```

where:

n is the query response interval from 1 to 25 seconds.

Configuring the BSR as a Static Multicast Group Member

An interface is configured with a static multicast group for the following reasons:

- Performance increases by allowing the BSR to forward multicast packets over the interface to directly connected networks without processing the multicast packets.
- A multicast group member is not associated with a specific network.
- The host cannot report its multicast group membership using IGMP.

Use the **ip igmp static-group** command in Interface Configuration mode to configure the BSR to become a statically connected multicast group member, as shown below:

```
MOT(config-if)#ip igmp static-group <ip-address>
```

where:

ip-address is the multicast group IP address for the BSR.

Managing IP Multicast Routing on the BSR

The following sections are used to manage multicast routing on the BSR:

- Configuring an IP Multicast Static Route
- Changing the Distance for a Unicast Multicast Route
- Changing the Distance for a Static Multicast Route
- Clearing IP Multicast Information

Configuring an IP Multicast Static Route

IP multicast static routes (mroutes) enable unicast and multicast packets to take different paths over combined multicast and unicast network topologies by allowing multicast packets to travel from the router that is configured with the static multicast route to the next multicast router, even if there are one or more unicast routers in the path. The router with the multicast static route uses the IP static multicast route configuration instead of the unicast routing table to determine the path, and no information about this IP multicast static route is advertised or redistributed to any other router on the network.

Use the **ip mroute** command in Global Configuration mode to configure a multicast static route which includes the multicast source address, as shown in the following example:

`MOT(config)#`**ip mroute** {<*source-address*> <*mask*> <*rpf-address*>} [<*n*>]

where:

source-address is the source IP address of the multicast static route.

mask is the source subnetwork IP address of the multicast static route.

rpf-address is the Reverse Path Forwarding neighbor IP address or route.

n is the optional administrative distance of the multicast static route.

Changing the Distance for a Unicast Multicast Route

The *distance* value is used to compare with the same source in the unicast routing table. The route that is configured with the lower administrative distance (which can be either the route in the unicast or DVMRP routing table) takes precedence when computing the Reverse Path Forwarding (RPF) interface for the source of a multicast packet. By default, the administrative distance for DVMRP a route is 0, and takes precedence over unicast routing table routes.

If there are two paths to a source, one through unicast routing (using Protocol Independent Multicast [PIM] as the multicast routing protocol) and another path using DVMRP (unicast and multicast routing), and PIM must be the path to a source, use the **ip mroute unicast distance** command in Global Configuration mode to increase the default administrative distance for the DVMRP route, as shown below:

```
MOT(config)#ip mroute unicast distance <n>
```

where:

n is the administrative distance number from 1 to 255.

Changing the Distance for a Static Multicast Route

Use the **ip mroute static distance** command in Global Configuration mode to set the default administrative distance for a multicast static route, as shown below:

```
MOT(config)#ip mroute static distance <n>
```

where:

n is the administrative distance number from 1 to 255.

Note: The default administrative distance for a multicast route is 0.

Clearing IP Multicast Information

Use the following sections to remove the contents of a particular cache, table or database when the contents are suspected to be invalid:

- Removing a DVMRP Prune
- Removing a DVMRP Route
- Clearing IGMP Statistics
- Removing the IP Multicast Cache

Removing a DVMRP Prune

Use the following options to remove a DVMRP prune:

- Use the **clear ip dvmrp prune *** command in Privileged EXEC mode to clear all dvmrp prunes, as shown below:

  `MOT#`**clear ip dvmrp prune ***

- Use the **clear ip dvmrp prune group** command in Privileged EXEC mode to clear prunes from a specific DVMRP group, as shown below:

  `MOT#`**clear ip dvmrp prune group** <*ip-address*>

  where:

  ip-address is the IP address of the DVMRP group.

- Use the **clear ip dvmrp prune neighbor** command in Privileged EXEC mode to clear prunes from a specific neighbor, as shown below:

  `MOT#`**clear ip dvmrp prune neighbor** <*ip-address*>

  where:

  ip-address is the IP address of the DVMRP neighbor.

Removing a DVMRP Route

Use the following options to remove a DVMRP route:

- If you want to remove a specific DVMRP route, use the **clear ip dvmrp route** command in Privileged EXEC mode, as shown below:

  `MOT#`**clear ip dvmrp route** <*ip-address*>

  where:

  ip-address is the IP address of the DVMRP route.

- If you want to clear all DVMRP routes, use the **clear ip dvmrp route *** command in Privileged EXEC mode, as shown below:

  `MOT#`**clear ip dvmrp route ***

Clearing IGMP Statistics

Use the **clear ip igmp counters** command in Privileged EXEC mode to remove IGMP statistics, as shown below:

```
MOT#clear ip igmp counters
```

Removing the IP Multicast Cache

Use the following options to remove IP multicast routing information:

- If you want to clear the entire IP multicast forwarding cache, use the **clear ip multicast fwd-cache** command in Privileged EXEC mode, as shown below:

  ```
  MOT#clear ip multicast fwd-cache
  ```

- If you want to clear the entire IP multicast protocol cache, use the **clear ip multicast proto-cache** command in Privileged EXEC mode, as shown below:

  ```
  MOT#clear ip multicast proto-cache
  ```

Gathering IP Multicast Information

Use the following sections to gather information for your multicast network:

- Displaying General IP Multicast Information
- Displaying PIM Information
- Displaying DVMRP Information
- Displaying IGMP Information
- Displaying Reverse Path Forwarding Information

Displaying General IP Multicast Information

Use the following options to view IP multicast cache information in Privileged EXEC mode:

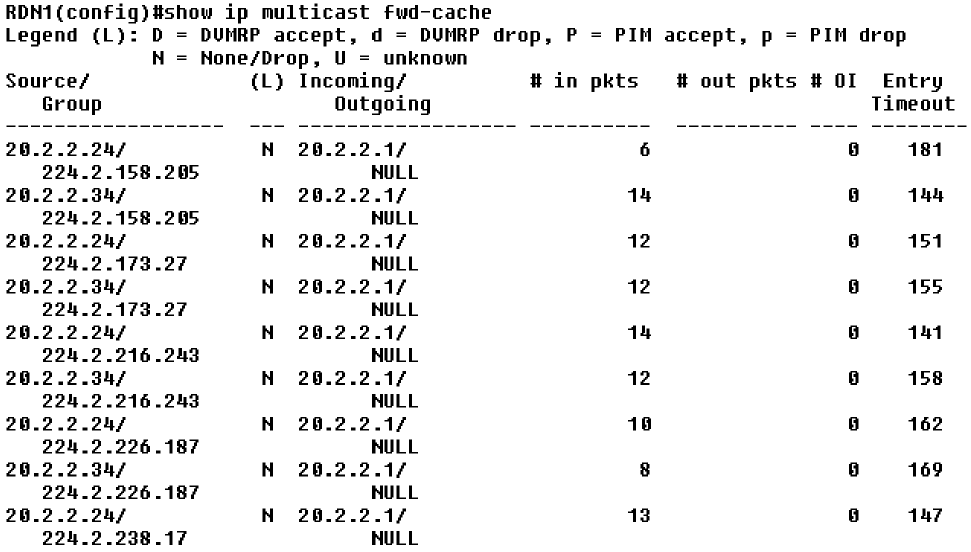
- Use the **show ip multicast cache-summary** command to display the total number of protocol cache and forwarding cache entries, as shown below:

  ```
  MOT#show ip multicast cache-summary
  ```

- Use the **show ip multicast fwd-cache** command to display multicast forwarding cache entries, as shown below:

MOT#**show ip multicast fwd-cache**

For example:



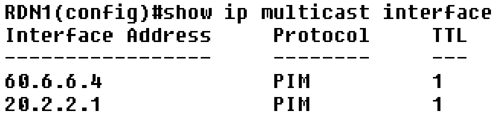
```
RDN1(config)#show ip multicast fwd-cache
Legend (L): D = DVMRP accept, d = DVMRP drop, P = PIM accept, p = PIM drop
            N = None/Drop, U = unknown
Source/             (L) Incoming/          # in pkts   # out pkts # OI  Entry
   Group                   Outgoing                                    Timeout
------------------  ---  ------------------ ----------  ---------- ---- --------
20.2.2.24/           N  20.2.2.1/                   6                0    181
   224.2.158.205              NULL
20.2.2.34/           N  20.2.2.1/                  14                0    144
   224.2.158.205              NULL
20.2.2.24/           N  20.2.2.1/                  12                0    151
   224.2.173.27               NULL
20.2.2.34/           N  20.2.2.1/                  12                0    155
   224.2.173.27               NULL
20.2.2.24/           N  20.2.2.1/                  14                0    141
   224.2.216.243              NULL
20.2.2.34/           N  20.2.2.1/                  12                0    158
   224.2.216.243              NULL
20.2.2.24/           N  20.2.2.1/                  10                0    162
   224.2.226.187              NULL
20.2.2.34/           N  20.2.2.1/                   8                0    169
   224.2.226.187              NULL
20.2.2.24/           N  20.2.2.1/                  13                0    147
   224.2.238.17               NULL
```

Figure 8-1 IP Multicast Forward Cache Information

- Use the **show ip multicast interface** command to list the IP address, multicast protocol (PIM, DVMRP, or IGMP), and time-to-live (TTL) information that is associated with each multicast interface, as shown below:

  MOT#**show ip multicast interface**

  For example:



```
RDN1(config)#show ip multicast interface
Interface Address     Protocol     TTL
-----------------     --------     ---
60.6.6.4              PIM          1
20.2.2.1              PIM          1
```

Figure 8-2 show ip multicast interface Command Output

- Use the **show ip multicast no-oi-fwd-cache** command to display multicast forwarding cache entries, which have no outgoing interfaces (OIs), as shown below:

```
MOT#show ip multicast no-oi-fwd-cache
```

- Use the **show ip multicast oi-fwd-cache** command to display multicast forwarding cache entries that have outgoing interfaces (OIs), as shown below:

```
MOT#show ip multicast oi-fwd-cache
```

- Use the **show ip multicast proto-cache** command to display multicast protocol cache entries, as shown below:

```
MOT#show ip multicast proto-cache
```

Displaying PIM Information

Use the following options to view PIM information in Privileged EXEC mode:

- Use the **show ip pim bsr-router** command to display the Version 2 PIM bootstrap router, as shown below:

```
MOT#show ip pim bsr-router
```

- Use the **show ip pim interface** command to display PIM interface information, as shown below:

```
MOT#show ip pim interface
```

- Use the **show ip pim neighbor** command to display the PIM neighboring router information, as shown below:

```
MOT#show ip pim neighbor
```

- Use the **show ip pim rp** command to display the PIM Rendezvous Point (RP) information, as shown below:

```
MOT#show ip pim rp
```

- Use the **show ip pim rp-hash** command to display the RP to be chosen based on the selected group, as shown below:

```
MOT#show ip pim rp-hash
```

- Use the **show ip pim unresolved-groups** command to display any unresolved PIM multicast groups, as shown below:

```
MOT#show ip pim unresolved-groups
```

Displaying DVMRP Information

Use the following options to view DVMRP information in Privileged EXEC mode:

- Use the **show ip dvmrp information** command to gather general DVMRP information, as shown below:

  ```
  MOT#show ip dvmrp information
  ```

- Use the **show ip dvmrp interface** command to gather information for all the DVMRP interfaces as shown below:

  ```
  MOT#show ip dvmrp interface
  ```

- Use the **show ip dvmrp neighbor** command to gather DVMRP neighbor information, as shown below:

  ```
  MOT#show ip dvmrp neighbor
  ```

- Use the **show ip dvmrp network** command to gather DVMRP network information, as shown below:

  ```
  MOT#show ip dvmrp network
  ```

- Use the **show ip dvmrp route** command to gather DVMRP routing information, as shown below:

  ```
  MOT#show ip dvmrp route
  ```

- Use the **show ip dvmrp summary-route** command to gather DVMRP summary route information, as shown below:

  ```
  MOT#show ip dvmrp summary-route
  ```

Displaying IGMP Information

Use the following options to view IGMP information in Privileged EXEC mode:

- Use the **show ip igmp groups** command to gather IGMP group membership information, as shown below:

  ```
  MOT#show ip igmp groups
  ```

- Use the **show ip igmp interface** command to gather IGMP interface information, as shown below:

  `MOT#`**show ip igmp interface**

- Use the **show ip igmp statistics** command to gather IGMP statistics information, as shown below:

  `MOT#`**show ip igmp interface**

Displaying Reverse Path Forwarding Information

Use the following options to view Reverse Path Forwarding (RPF) information in Privileged EXEC mode:

- Use the **show ip rpf** command to display a specific multicast source router, as shown below:

  `MOT#`**show ip rpf** <*ip-address*>

  where:

  ip-address is the IP address of multicast source router

  - or -

  If you want to show any available multicast source router(s) use the **show ip rpf** command, as shown below:

  `MOT#`**show ip rpf**

9

Configuring RIP

Overview

This chapter describes how to configure the Routing Information Protocol (RIP) for the BSR 64000™:

- About RIP
- Enabling RIP
- Specifying a RIP Version
- Enabling or Disabling Split Horizon
- Enabling Route Summarization
- Applying an Offset List
- Enabling RIP Authentication
- Configuring Interpacket Delay
- Configuring Timers
- Configuring a Passive Interface for RIP
- Redistributing Routes into RIP
- Gathering RIP Information

About RIP

Routing Information Protocol (RIP) is an Interior Gateway Protocol (IGP) that uses distance-vector routing to measure the shortest path between two points on a network. Distance-vector routing requires that each router inform its neighbors of its routing table. For each network path, the receiving router selects the neighbor advertising the lowest cost, and adds this entry to its routing table for re-advertisement. A host using RIP should have interfaces to one or more networks, which are known as directly connected networks.

RIP uses broadcast User Datagram Protocol (UDP) data packets to exchange routing information. RIP listens for these broadcasts on UDP port 520. The BSR uses the advertising process to send routing information updates every 30 seconds. If a router does not receive an update from another router for 180 seconds, it marks the routes served by the non-updating router as unusable. If the router does not receive an update after 300 seconds, it removes all routing table entries for the non-updating router.

RIP uses the hop count to rate the value of different routes. A directly connected network has a hop count of one; an unreachable network has a hop count of 16. This small range of metrics makes RIP unsuitable for large networks.

The route tag field in a RIP message allows boundary routers in an autonomous system (AS) to exchange information about external routes. Route tags separate internal RIP routes from external RIP routes that were imported from an Exterior Gateway Protocol (EGP) or another IGP. Routers that support protocols other than RIP should allow configuration of route tags for routes imported from different sources.

The subnet mask field in a RIP (RIPv2 only) message contains the subnet mask applied to the IP address to set the non-host portion of the address. If the subnet mask field is not used, the subnet mask is calculated. On an interface where a RIPv1 router operates on information in a RIPv2 routing entry, the following rules apply:

- Information internal to one network must never be advertised to another network.
- Information about a more specific subnet may not be advertised where RIPv1 routers would consider it a host route.
- Supernet routes (routes where a netmask is less specific than the natural network mask) must not be advertised where they could be misinterpreted by RIP routers.

The next hop field in a RIP (RIPv2 only) message contains the next destination IP address. A value of zero in this field indicates that the next destination is the origin of the RIP message. To reduce unnecessary load on hosts that do not listen to RIPv2 messages. RIP update packets use IP multicast address 224.0.0.9.

Specifications

The BSR supports the following Request for Comment (RFC) specifications:

RFC 1058 — *Routing Information Protocol*

RFC 2453 — *RIP Version 2*

Enabling RIP

In order to use RIP on the BSR, RIP must be enabled. The remaining tasks described in this chapter are optional.

Follow these steps to enable RIP on the BSR:

1. Use the **router rip** command in Global Configuration mode to enable a RIP routing process on the BSR, as shown in the example below:

   `MOT(config)#`**router rip**

   This enables RIP and places you in Router Configuration mode. Use the **no router rip** command to disable RIP.

2. You may specify multiple **network** sub-commands. RIP routing updates are sent and received only through interfaces on the network that you specify. If you do not specify the IP address related to the interface, RIP updates do not advertise the network associated with this interface. Use the **network** command in Router Configuration mode to associate a network with the RIP routing process, as shown in the example below:

   `MOT(config-rip)#`**network** <*ip-address*> [<*subnetmask*>]

   where:

   ip-address is the network number.

   subnetmask is the network mask to the new address so that RIP runs on that specific network.

Example

The following example configuration defines RIP as the routing protocol to be used on all interfaces connected to networks 138.82.0.0, 182.41.4.0, and 10.10.10.0:

```
router rip
network 138.82.0.0 255.255.0.0
network 182.41.5.0 255.255.255.0
network 10.10.10.0 255.255.255.0
```

Specifying a RIP Version

By default, the software receives RIPv1 and v2 packets, but sends only RIPv2 packets. You can configure the software to receive and send only RIPv1 packets. You can also configure the software to receive and send only RIPv2 packets.

1. To configure the software to receive and send only RIPv1 packets or only RIPv2 packets, use the **version** command in Router RIP Configuration mode, as shown in the example below:

```
MOT(config-rip)#version {1 | 2}
```

where:

1 configures an interface to receive and send only RIPv1 packets.

2 configures an interface to receive and send only RIPv2 packets.

The **version** command specifies a RIP version used globally by the router. This controls only the RIP default. You can configure a particular interface to behave differently.

2. To control the RIP version an interface sends or receives, use the **ip rip send version** or **ip rip receive version** command in Interface Configuration mode. To configure an interface to receive only RIPv1 packets and/or RIPv2 packets, use the following **ip rip receive version** command in Interface Configuration mode:

```
MOT(config-if)#ip rip receive version {0 | 1 | 2}
```

where:

0 configures an interface to receive RIPv1 and RIPv2 packets.

1 configures an interface to receive only RIPv1 packets.

2 configures an interface to receive only RIPv2 packets.

3. Use the **ip rip send version** command in Interface Configuration mode to configure an interface to send only RIPv1 or RIPv2 packets, as shown below:

```
MOT(config-if)#ip rip send version {1 | 2 | 0 | 3}
```

where:

0 configures an interface to send only RIPv1 and RIPv2 packets.

1 configures an interface to send only RIPv1 packets.

2 configures an interface to send only RIPv2 packets.

3 configures an interface not to send RIP packets.

Examples

The following example configures the interface to send RIPv1 packets from the interface:

```
ip rip send version 1
```

The next example configures the interface to send RIPv2 packets from the interface:

```
ip rip send version 2
```

This example configures the interface to receive only RIPv1 packets:

```
ip rip receive version 1
```

Enabling or Disabling Split Horizon

Routers that use distance-vector routing protocols and that connect to broadcast-type IP networks use the split horizon with poisoned reverse mechanism to prevent routing loops. Split horizon with poisoned reverse advertises route information with a metric of 16 on any (unreachable) interface from which that information originated. This usually optimizes communications among multiple routers, particularly when links are broken.

Follow these steps to enable or disable split horizon:

1. To enable split horizon, use the **ip split-horizon** command in Interface Configuration mode, as shown in the example below:

   ```
   MOT(config-if)#ip split-horizon
   ```

2. To disable split horizon, use the following **no ip split-horizon** command in Interface Configuration mode:

   ```
   MOT(config-if)#no ip split-horizon
   ```

Enabling Route Summarization

RIPv2 supports route summarization, which condenses routing information and reduces the router load and the perceived network complexity. The larger the network, the more important route summarization is. Without route summarization, a router retains a route to every subnet in its network. With summarization, the router can reduce some sets of routes to a single advertisement. Route propagation and routing information are reduced significantly.

In Figure 9-1, Router Dallas maintains one route for all destination networks beginning with B, and Router Providence maintains one route for all destination networks beginning with A. Router Dallas tracks all routes because it exists on the boundary between A and B.

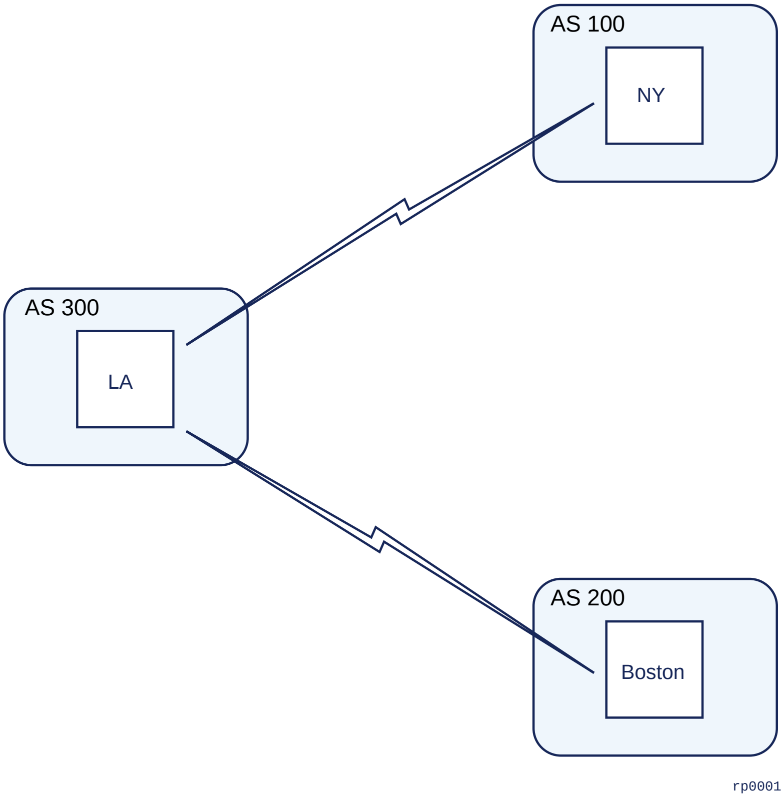


Figure 9-1 Route Summarization

1. To enable route summarization, use the **auto-summary** command in Router RIP Configuration mode, as shown in the example below. It is disabled by default.

   MOT(config-rip)# **auto-summary**

2. If you disconnect subnetworks, disable automatic route summarization to advertise the subnetworks, using the **no auto-summary** command. When route summarization is disabled, the software transmits subnet and host routing information across classful network boundaries.

Applying an Offset List

To increase incoming and outgoing metrics to routes learned via RIP, use an offset list. You can also limit the offset list using an access list or an interface. To increase the value of routing metrics using an offset list, use the **offset-list** command in Router RIP Configuration mode, as shown below:

MOT(config-rip)#**offset-list** [<*access-list-number*> | *any*] {**in** | **out**} **offset** [*type*/*number*]

where:

access-list-number is the standard access list number to be applied; values are from 1 to 99. If *offset* is 0, no action is taken.

any is any access list number.

in applies the access list to incoming metrics.

out applies the access list to outgoing metrics.

offset is the positive offset from 0 to16 to be applied to metrics for networks matching the access list. If the offset is 0, no action is taken.

type is the interface type to which the offset list is applied.

number is the interface number to which the offset list is applied.

Examples

In the following example, the router applies an offset of 3 to the router delay component of access list 13:

```
offset-list 13 out 3
```

In the following example, the router applies an offset of 4 to routes learned from Ethernet interface 1/0:

```
offset-list 13 in 3 ethernet 1/0
```

Enabling RIP Authentication

RIPv1 does not support authentication. If you send and receive RIPv2 packets, you can enable RIP authentication on a particular interface. The BSR supports plain-text password authentication and MD5-encrypted password authentication on a RIP interface.

Use one of the following options to enable RIP authentication:

- Use the **ip rip authentication key** command in Interface Configuration mode to enable plain text password authentication, as shown below:

  `MOT(config-if)#`**ip rip authentication key** <*password*>

  where:

  password specifies the 16 character password authentication key.

- Use the **ip rip message-digest-key md5** command in Interface Configuration mode to enable encrypted Message Digest Five (MD5) password authentication, as shown below:

  `MOT(config-if)#`**ip rip message-digest-key** <*n*> **md5** <*password*>

  where:

  n specifies the a key number from 1 to 255

  password specifies the MD5-encrypted 16 character password.

Configuring Interpacket Delay

By default, the software adds no delay between packets in a multiple-packet RIP update being sent. If you have a high-end router sending to a low-speed router, you might want to add such interpacket delay to RIP updates, in the range of 8 to 50 milliseconds. To do so, use the **output-delay** command in Router RIP Configuration mode, as shown in the following example:

```
MOT(config-rip)#output-delay <delay>
```

where:

delay is the delay, in milliseconds, between packets in a multiple-packet RIP update; valid values are 8 to 50 milliseconds; default is no delay.

Use the **no output-delay** command to return to the default.

Configuring Timers

Routing protocols use timers to determine time intervals for route information adjustment.A number of seconds to the setting to prevent collisions.

1. To determine the current timers, use the **show ip protocols** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show ip protocols
   ```

2. To set the RIP network timers, use the **timers basic** command in Router RIP Configuration mode, as shown below.

   ```
   MOT(config-rip)#timers basic <update-timer> <invalid> <timeout>
   <flush-timer>
   ```

   where:

   update-timer is the update timer value in seconds between periodic routing updates; default is 30.

   invalid is the interval of time in seconds after which a route is declared invalid; this interval should be at least three times the *update-timer* value. A route becomes invalid when there is an absence of updates that refresh the route. The route is marked inaccessible and advertised as unreachable. The route, however, is still used to forward packets.

   timeout is the interval value in seconds for routing updates; default is 180.

flush-timer is the interval value in seconds that elapse before a route is removed from the routing table; default is 300.The interval specified must be greater than the *invalid* value.

Note: You can adjust basic RIP timers, but they must be the same for all routers and servers.

Example

The following example establishes a 60 second routing update timer, a 360 second route timeout timer, and a 600 second route-flush timer:

```
timers basic 60 360 600
```

Configuring a Passive Interface for RIP

You can configure a passive interface to prevent other routers on a local network from learning about routes dynamically. A passive interface does not transmit routing updates.

1. Use the **passive-interface** command in Router Configuration mode to create a passive RIP interface, as shown in the following example:

   `MOT(config-rip)#`**passive-interface** {**cable** | **ethernet** | **gigaether** | **pos** | **serial**} <*slot*>/<*interface*>

   where:

   cable is the cable interface.

   ethernet is the Ethernet/Fast Ethernet interface.

   gigaether is the Gigabit Ethernet interface.

   pos is the Packet over SONET interface.

   serial is the Serial interface.

   slot is the module slot number.

interface is the interface number.

2. Use the **passive-interface default** command in Router Configuration mode to suppress routing updates on all RIP interfaces, as shown below:

```
MOT(config-rip)#passive-interface ethernet <slot>/<interface>
```

where:

slot is the module slot number.

interface is the interface number.

Redistributing Routes into RIP

Each routing protocol uses different metrics to transfer routes. Some protocols use hop count metrics, while others use bandwidth and delay attributes to define metrics. When a specific route is redistributed from one routing protocol or domain into another, a common metric must be applied by the receiving protocol. Routes are redistributed to advertise networks on another routing protocol.

Note: The metric values for applying non-RIP routes are limited to values from 1 to 16, which are the metrics used by RIP. RIP metrics are established by hop-counts.

Follow these steps to redistribute routes into RIP:

1. Enter the routing process in which the routes are to be redistributed, as shown below:

```
MOT(config)#router rip
```

2. Choose from one or more of the following options to redistribute routes from a specified protocol:

   - Use the **redistribute connected** command in Router Configuration mode to redistribute connected routes into RIP, as shown below:

```
MOT(config-rip)#redistribute connected {metric <n> | route-map
<map-name> | <cr>}
```

where:

metric <*n*> is the redistribution metric number for connected routes from 1 to 16.

route-map <*map-name*> is the route-map name for the connected route.

cr is a command return that redistributes of all connected routes.

- Use the **redistribute isis** command in Router Configuration mode to redistribute IS-IS routes into RIP, as shown below:

  `MOT(config-rip)#`**redistribute isis** {**match** [**level-1** | **level-1-2** | **level-2**] | **metric** <*n*> | **route-map** <*map-name*> | <*cr*>}

  where:

  The **match** argument is used to choose level 1 ISIS routes only, level 1 and 2 ISIS routes, or level 2 ISIS routes only.

  metric <*n*> is the redistribution metric number for ISIS routes from 1 to 16.

  route-map <*map-name*> is the route-map name for the ISIS route.

  cr is a command return that redistributes of all ISIS routes.

- Use the **redistribute ospf** command in Router Configuration mode to redistribute OSPF routes into RIP, as shown below:

  `MOT(config-rip)#`**redistribute ospf** {[**external** | **internal**] | **metric** <*n*> | **route-map** <*map-name*> | <*cr*>}

  where:

  The **external** argument is used to redistribute external OSPF routes.

  The **internal** argument is used to redistribute internal OSPF routes.

  metric <*n*> is the redistribution metric number for OSPF routes from 1 to 16.

  route-map <*map-name*> is the route-map name for the OSPF route.

  cr is a command return that redistributes of all OSPF routes.

- Use the **redistribute bgp** command in Router Configuration mode to redistribute BGP routes into RIP, as shown below:

  `MOT(config-rip)#`**redistribute bgp** {**metric** <*n*> | **route-map** <*map-name*> | <*cr*>}

where:

metric <*n*> is the redistribution metric number for BGP routes from 1 to 16.

route-map <*map-name*> is the route-map name for the BGP route.

cr is a command return that redistributes of all BGP routes.

- Use the **redistribute static** command in Router Configuration mode to redistribute static routes into RIP, as shown below:

  `MOT(config-rip)#`**redistribute static** {**metric** <*n*> | **route-map** <*map-name*> | <*cr*>}

  where:

  metric <*n*> is the redistribution metric number for static routes from 1 to 16.

  route-map <*map-name*> is the route-map name for the static route.

  cr is a command return that redistributes of all static routes.

Assigning a Default Metric Value for Redistributed Routes

The default metric function is used to eliminate the need for separate metric definitions for each routing protocol redistribution.

Follow these steps to assign a default metric value for all routes redistributed into RIP:

1. Use the **router rip** command to enter the RIP routing process in Global Configuration mode, as shown below:

   `MOT(config)#`**router rip**

2. Use the **default-metric** command in Router Configuration mode to force a routing protocol to use the same metric value for all distributed routes from other routing protocols, as shown below:

   `MOT(config-rip)#`**default-metric** <*n*>

   where:

   n is the default value for all routes that are redistributed into RIP.

Gathering RIP Information

Follow these steps to monitor RIP on the BSR:

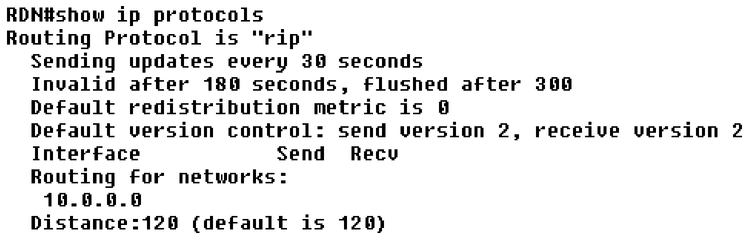
1. To display RIP routes and the status of each rip route on the BSR interface, use the **show ip route rip** command in Privileged EXEC mode as shown in the following example:

   MOT#**show ip route rip**

2. To display the configured network parameters for RIP, use the **show ip protocols** in Privileged EXEC mode, as shown in the following example:

   MOT#**show ip protocols**

   Figure gives an example of the **show ip protocols** command output:



```
RDN#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds
  Invalid after 180 seconds, flushed after 300
  Default redistribution metric is 0
  Default version control: send version 2, receive version 2
  Interface           Send  Recv
  Routing for networks:
   10.0.0.0
  Distance:120 (default is 120)
```

Figure 9-2 show ip protocols Command Output

3. To display the entire contents of the private RIP database when triggered extensions to RIP are enabled, use the **show ip rip database** command in Privileged EXEC mode, as shown in the following example:

   MOT#**show ip rip database**

4. To view the routing table, enter the **show ip route** command in Privileged EXEC mode, as shown in the following example:

   MOT#**show ip route**

10

Configuring IS-IS

Overview

The integrated Intermediate System to Intermediate System (IS-IS) is a link state based intra-domain routing protocol used to build a complete and consistent picture of a network's topology by sharing link state information across all network Intermediate System (IS) devices. IS-IS is based on an SPF routing algorithm and shares all the advantages common to other link-state protocols. It also routes both IP packets and pure OSI packets with no extra encapsulation by design. IS-IS supports type of service (TOS) identifiers, IP subnetting, variable subnet masks, external routing, and authentication.

IS-IS routing decisions are made on two levels. Level 1 routers know the topology of their network within an area and level 2 routers are used to route between different areas within a routing domain. If a level 1 router has no knowledge of a specific destination address, it passes the traffic to a level 2 router. Level 2 routers know which addresses are reachable through each Level 2 router, and they do not need to know the topology within Level 1 areas. Level 2 routers can also exchange information with external routers outside their routing domain.

The following task is required to implement IS-IS on your network:

- Enabling IS-IS
- Redistributing Routes into IS-IS

The following optional tasks are used to manage IS-IS on your network:

- Managing IS-IS on the BSR
- Configuring IS-IS on an Interface
- Gathering IS-IS Information

Enabling IS-IS

You must enable the IS-IS process and assign IS-IS to a specific interfaces in order to implement IS-IS on your network.

Follow these steps to enable IS-IS on the router:

1. Use the **router isis** command in Global Configuration mode to enable IS-IS routing and specify an IS-IS process for IP communication, which places you in router configuration mode.

   ```
   MOT(config)#router isis [tag]
   ```

   where:

   tag is the name for the routing process. If the tag is not specified, a null tag is assumed.

Network Entity Titles (NETs) define the area address and the system ID for an IS-IS router. Most end systems and intermediate systems have one NET. Intermediate systems that participate in multiple areas can have multiple NETs.

2. Use the **net** command to define a Network Entity Title (NET) in Router Configuration mode, as shown below:

Note: Under most circumstances, one NET is configured.

```
MOT(config-isis)#net <title>
```

where:

title is the IS-IS area ID and system ID for an IS-IS router.

The following example configures IS-IS for IP routing, with an area ID of 01.0001 and a system ID of 0000.0000.0002:

```
MOT(config-isis)#net 01.0001.0000.0000.0002.00
```

Use the following criteria to interpret the IS-IS NET address format:

- The first portion of the NET address is the area number, which is a variable number from 1 through 13 bytes. The first byte of the area number (01) is the authority and format indicator (AFI). The next bytes are the assigned domain (area) identifier, which can be from 0 through 12 bytes. In the example above, the area identifier is 0001.
- The next six bytes form the system identifier (SYSID). The SYSID can be any six bytes that are unique throughout the entire domain. The system identifier commonly is either the Media Access Control (MAC) address or the IP address expressed in binary-coded decimal (BCD).
- The last byte (00) is the n-selector.

3. Use the **interface** command in Global Configuration mode to select the interface on which you plan to configure IS-IS.

   `MOT(config)#`**interface** <*type*> <*slot*> / <*interface*>

   where:

   type is the BSR interface type.

   slot is the module number.

   interface is the interface number on which IS-IS is configured.

4. Use the **ip router isis** command in Interface Configuration mode to enable IS-IS routing on the interface, as shown below:

   `MOT(config-if)#`**ip router isis**

   - or -

   Use the **ip router isis passive** command in Interface Configuration mode to allow the IS-IS interface to receive IS-IS network information, but not send IS-IS network information, as shown below:

   `MOT(config-if)#`**ip router isis passive**

Redistributing Routes into IS-IS

Each routing protocol uses different metrics to transfer routes. Some protocols use hop count metrics, while others use bandwidth and delay attributes to define metrics. When a specific route is redistributed from one routing protocol or domain into another, a common metric must be applied by the receiving protocol. Routes are redistributed to advertise networks on another routing protocol.

Follow these steps to redistribute routes into IS-IS:

1. Enter the IS-IS routing process in which the routes are to be redistributed, as shown below:

   ```
   MOT(config)#router isis
   ```

2. Choose from one or more of the following options to redistribute routes from a specified protocol:

   - Use the **redistribute ospf** command in Router Configuration mode to redistribute OSPF routes into IS-IS, as shown below:

     ```
     MOT(config-isis)#redistribute ospf {[external | internal] | metric <n> | metric-type [external | internal] | route-map <map-name> | <cr>}
     ```

     where:

     The **external** argument is used to redistribute external OSPF routes.

     The **internal** argument is used to redistribute internal OSPF routes.

     metric <*n*> is the redistribution metric number for OSPF routes.

     The **metric-type external** argument is used to redistribute external IS-IS metric-type.

     The **metric-type-internal** argument is used to redistribute internal IS-IS metric-type.

     route-map <*map-name*> is the OSPF route-map name.

     cr is a command return that redistributes of all OSPF routes.

   - Use the **redistribute connected** command in Router Configuration mode to redistribute connected routes into IS-IS, as shown below:

```
MOT(config-isis)#redistribute connected {metric <n> | route-map
<map-name> | <cr>}
```

where:

metric <*n*> is the redistribution metric number for connected routes.

route-map <*map-name*> is the route-map name for the connected route.

cr is a command return that redistributes of all connected routes.

- Use the **redistribute bgp** command in Router Configuration mode to redistribute BGP routes into IS-IS, as shown below:

  ```
  MOT(config-isis)#redistribute bgp {metric <n> | route-map
  <map-name> | <cr>}
  ```

  where:

  metric <*n*> is the redistribution metric number for BGP routes.

  route-map <*map-name*> is the BGP route-map name.

  cr is a command return that redistributes of all BGP routes.

- Use the **redistribute rip** command in Router Configuration mode to redistribute RIP routes into IS-IS, as shown below:

  ```
  MOT(config-isis)#redistribute rip {metric <n> | route-map
  <map-name> | weight <n> | <cr>}
  ```

  where:

  metric <*n*> is the redistribution metric number for RIP routes.

  route-map <*map-name*> is the route-map name for the RIP route.

  weight <*n*> sets the network weight value from 0 to 65535 for redistributing RIP routes into IS-IS.

  cr is a command return that redistributes of all RIP routes into IS-IS.

- Use the **redistribute static** command in Router Configuration mode to redistribute static routes into ISIS, as shown below:

  ```
  MOT(config-isis)#redistribute static {metric <n> | metric-type [1
  |2] | route-map <map-name> | subnets | tag | <cr>}
  ```

  where:

metric <*n*> is the redistribution metric number for static routes.

metric-type 1 redistributes OSPF External Type 1 metrics.

metric-type 2 redistributes OSPF External Type 2 metrics.

route-map <*map-name*> is the route-map name for the static route.

subnets allows the consideration of subnets for redistribution into IS-IS.

tag sets a tag for routes redistributed into IS-IS.

cr is a command return that redistributes of all static routes.

Assigning a Default Metric Value for Redistributed Routes

The default metric function is used to eliminate the need for separate metric definitions for each routing protocol redistribution.

Follow these steps to assign a default metric value for all routes redistributed into IS-IS:

1. Use the **router isis** command to enter the IS-IS routing process in Global Configuration mode, as shown below:

   ```
   MOT(config)#router isis
   ```

2. Use the **default-metric** command in Router Configuration mode to force a routing protocol to use the same metric value for all distributed routes from other routing protocols, as shown below:

   ```
   MOT(config-isis)#default-metric <n>
   ```

   where:

   n is the default metric value for all routes that are redistributed into ISIS.

Managing IS-IS on the BSR

The following sections are used to manage IS-IS on the BSR:

- Specifying Router-Level Support
- Forcing a Default Route or Route Map into an IS-IS Domain

- Configuring the Administrative Distance for IS-IS
- Configuring IS-IS Area or Domain Passwords
- Summarizing IP Address Ranges
- Enabling the LSP Overload Bit
- Configuring a Passive Interface for IS-IS

Specifying Router-Level Support

Use the **is-type** command in Router Configuration mode to specify that the BSR acts as a Level 1 (intra-area) router, as both a Level 1 router and a Level 2 (inter-area) router, or as a Level-2 router only, as shown below:

```
MOT(config-isis)#is-type {level-1 | level-1-2 | level-2-only}
```

Forcing a Default Route or Route Map into an IS-IS Domain

When routes are redistributed to an IS-IS routing domain, the BSR can be configured to force a default route or route map into the IS-IS routing domain:

Follow these steps to force the default route into an IS-IS routing domain:

1. Use the **router isis** command in Global Configuration mode to enter the IS-IS router configuration mode, as shown below:

   ```
   MOT(config)#router isis [<tag>]
   ```

   where:

   > *tag* is the tag name for IS-IS routing process. If this parameter is not specified, a null tag is assumed.

2. Use the **default-information originate** command in Router Configuration mode to force a default route into the IS-IS routing domain, as shown below:

   ```
   MOT(config-isis)#default-information originate
   ```

Configuring the Administrative Distance for IS-IS

An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. Numerically, an administrative distance is an integer between 0 and 255. The higher the value, the lower the trust rating. For example, an administrative distance of 255 means the routing information source cannot be trusted and should be ignored. The default administrative distance for IS-IS is 115.

Use the **distance** command in Router Configuration mode to set the administrative distance for the IS-IS router, as shown below:

`MOT(config-isis)#`**distance** <*n*>

where:

n is the IS-IS routing administrative distance from 1 to 255.

Configuring IS-IS Area or Domain Passwords

Passwords can be assigned to areas and domains. The area authentication password is inserted in Level 1 (station router level) LSPs, CSNPs, and Partial Sequence Number PDUs (PSNPs). The routing domain authentication password is inserted in Level 2 (the area router level) LSP, CSNP, and PSNPs.

Follow these steps to configure either area or domain authentication passwords:

- If you want to configure an IS-IS area authentication password, use the **area-password** command in Router Configuration mode, as shown below:

  `MOT(config-isis)#`**area-password** <*password*>

  where:

  password is the unencrypted text password that is 1-8 characters in length.

- If you want to configure an IS-IS routing domain authentication password, use the **domain-password** command in Router Configuration mode, as shown below:

  `MOT(config-isis)#`**domain-password** <*password*>

  where:

  password is the unencrypted text password that is 1-8 characters in length.

Summarizing IP Address Ranges

A range of IP addresses listed in an LSP can be represented by a summary address. Routes learned from other routing protocols also can be summarized. The metric used to advertise the summary is the smallest metric of all the more specific routes.

Use the **summary-address** command in Router Configuration mode to create a summary address for a range of IP addresses, as shown below:

`MOT(config-isis)#`**summary-address** <*ip-address*> <*mask*> {**level-1** | **level-1-2** | **level-2**}

Enabling the LSP Overload Bit

The LSP overload bit is enabled for the IS-IS routing process to ensure that no paths through the BSR are seen by other routers in the IS-IS area when the Link State routing table on the BSR is incomplete or inaccurate. However, IP and CLNS prefixes directly connected to the BSR continue to be reachable.

Use the **set-overload-bit** command in Router Configuration mode to allow other routers on the network to ignore IS-IS routing problems on the BSR in their SPF calculations until the IS-IS routing process on the BSR has recovered from its problems, as shown below:

`MOT(config-isis)#`**set-overload-bit**

Configuring a Passive Interface for IS-IS

You can configure a passive interface to prevent other routers on a local network from learning about routes dynamically. A passive interface does not transmit routing updates.

Use the **passive-interface** command in Router Configuration mode to create a passive IS-IS interface, as shown in the following example:

`MOT(config-isis)#`**passive-interface** {**cable** | **ethernet** | **gigaether** | **pos**} <*slot*>/<*interface*>

where:

cable is the cable interface.

ethernet is the Ethernet/Fast Ethernet interface.

gigaether is the Gigabit Ethernet interface.

pos is the Packet over SONET interface.

slot is the module slot number.

interface is the interface number.

Configuring IS-IS on an Interface

The following configuration tasks performed on the designated IS-IS interface are optional. They are used to adapt IS-IS to your network. When you configure IS-IS parameters on a BSR interface, ensure that they are consistent with other routers on your network.

- Specifying the Interface Circuit Type of an IS-IS Interface
- Configuring IS-IS Link-State Cost Metrics
- Setting the Advertised Hello Interval
- Specifying the Advertised Hello Multiplier
- Setting the Advertised CSNP Interval
- Setting the LSP Interval
- Setting the LSP Retransmission Interval
- Setting the LSP Retransmit Throttle Interval
- Setting the Designated Router Priority
- Assigning a Password to an IS-IS Interface

Specifying the Interface Circuit Type of an IS-IS Interface

The default IS-IS interface circuit type is for Level 1 and Level 2. Use the **isis circuit-type** command in Interface Configuration mode to select the IS-IS interface circuit type of adjacency desired for neighbors on the BSR interface (IS-IS interface circuit type):

```
MOT(config-if)#isis circuit-type {level-1 | level-1-2 | level-2-only}
```

where:

level-1 indicates that a Level 1 adjacency may be established if there is at least one area address in common between this system and its neighbors.

level-1-2 indicates that a Level 1 and 2 adjacency is established if the neighbor is also configured as level-1-2 and there is at least one area in common. If there is no area in common, a Level 2 adjacency is established.

level-2-only indicates that a Level 2 adjacency is established only if the neighbor is configured exclusively to be a Level 2 router.

Configuring IS-IS Link-State Cost Metrics

You can configure a cost for a specified interface. A cost is an arbitrary routing metric value assigned for crossing or intersecting networks. This metric can be applied to both Level 1 and/or Level 2 routing.

Use the **isis metric** command in Interface Configuration mode, to configure the metric cost for the specified IS-IS interface:

```
MOT(config-if)#isis metric <cost> {level-1 | level-2}
```

where:

cost is the assigned routing metric value for the interface.

level-1 is for Level 1 IS-IS routing.

level-2 is for Level 2 IS-IS routing.

Setting the Advertised Hello Interval

IS-IS hello packets are broadcast to discover the identity of neighboring IS-IS systems and to determine whether the neighbors are Level 1 or Level 2 intermediate systems.

The hello-interval multiplier is the amount of time that the IS-IS routing interface can tolerate not receiving hello packets from its neighboring IS-IS interface before declaring the neighbor as being down.

Use the **isis hello-interval** command in Interface Configuration mode to specify the length of time between hello packets that the BSR sends on either the Level 1 or Level 2 IS-IS router interface, as shown below:

```
MOT(config-if)#isis hello-interval <seconds> {level-1 | level-2}
```

where:

seconds is the ISIS hello interval.

level-1 is for Level 1 IS-IS routing.

level-2 is for Level 2 IS-IS routing.

The hello interval can be configured independently for Level 1 and Level 2, except on serial point-to-point interfaces. (Because there is only a single type of hello packet sent on serial links, it is independent of Level 1 or Level 2.) Specify an optional level for X.25, and Frame Relay multiaccess networks.

Specifying the Advertised Hello Multiplier

Use the **isis hello-multiplier** command in Interface Configuration mode to specify the the number of Intermediate System-to-Intermediate System (IS-IS) hello packets a neighbor can miss before the BSR determines the adjacency between the BSR interface and the neighbor is down, as shown below:

```
MOT(config-if)#isis hello-multiplier <n>
```

where:

n is the number of missing hello packets from 1 to 65535.

Setting the Advertised CSNP Interval

Complete Sequence Number PDUs (CSNPs) hold a complete list of all LSPs in the IS-IS routing database. CSNPs are sent periodically on all links, and the receiving systems use the information in the CSNP to update and synchronize their LSP databases. The designated router multicasts CSNPs on broadcast links in place of sending explicit acknowledgments for each LSP.

By default, IS-IS sends CSN packets periodically. If the BSR is the designated router on a LAN, IS-IS sends CSN packets every 10 seconds. If the BSR is on a point-to-point interface, it sends CSN packets every 3600 seconds (once an hour). Depending on your network topology you may want to modify the default interval to protect against LSP flooding.

Use the **isis csnp-interval** command in Interface Configuration mode to adjust the IS-IS CSNP interval for intranet connections if the intranet is a part of a multiaccess meshed network on the interface, as shown below:

`MOT(config-if)#`**isis csnp-interval** <*seconds*> {**level-1** | **level-2**}

where:

seconds is the ISIS CSNP interval from 1 to 65535 seconds.

level-1 indicates that the interface is a Level 1 IS-IS interface.

level-2 indicates that the interface is a Level 2 IS-IS interface.

Setting the LSP Interval

IS-IS Link-state PDUs (LSPs) hold information about the state of adjacencies to neighboring IS-IS systems. LSPs are flooded periodically throughout an area.

Use the **isis lsp-interval** command in Interface Configuration mode to configure the time delay between successive link state packet (LSP) transmissions, as shown below:

`MOT(config-if)#`**isis lsp-interval** <*milliseconds*>

where:

milliseconds is the time delay between successive LSPs from 1 to 65535 milliseconds.

Setting the LSP Retransmission Interval

When LSPs are dropped, LSPs are retransmitted. Use the **isis retransmit-interval** command in Interface Configuration mode to set the number of seconds between retransmission of each LSP for point-to-point links, as shown below:

Note: The number of seconds should be greater than the expected round-trip delay between any two routers on the attached network. Set this parameter conservatively to avoid unnecessary retransmission. Increase the number of seconds for networks that have serial lines and virtual links.

```
MOT(config-if)#isis retransmit-interval <seconds>
```

where:

seconds is the number of seconds between LSP broadcasts.

Setting the LSP Retransmit Throttle Interval

To configure the amount of time between any LSP retransmissions on a point-to-point interface, use the **isis retransmit-throttle-interval** interface configuration command in Interface Configuration mode as shown below:

```
MOT(config-if)#isis retransmit-throttle-interval <milliseconds>
```

where:

milliseconds is the minimum delay from 1 to 65535 milliseconds between LSP retransmissions on the interface.

Setting the Designated Router Priority

A BSR uses hello packets to advertise its priority to become a designated router. IS-IS uses the advertised priorities on all multiaccess networks to elect a designated router for the network. This router is responsible for sending network LSP advertisements, which describe all the routers attached to the network. These advertisements are flooded throughout a single area. The priority value is meaningful only on a multiaccess network. It has no meaning on a point-to-point interface.

A router’s priority for becoming the designated router is indicated by an arbitrary number. Routers with a higher value are more likely to become the designated router. By default, routers have a priority value of 64.

Use the **isis priority** command to select the designated router priority in Interface Configuration mode, as shown below:

Note: Priorities can be configured for Level 1 and Level 2 individually.

```
MOT(config-if)#isis priority <n> {level-1 | level-2}
```

n is a number from 0 to 127 that gives a priority value to the designated router.

level-1 indicates a Level 1 IS-IS router.

level-2 indicates a Level 2 IS-IS router.

Assigning a Password to an IS-IS Interface

You can assign different passwords for the different IS-IS routing levels. Specifying Level 1 or Level 2 configures the password for only Level 1 or Level 2 routing, respectively. By default, authentication is disabled.

Use the **isis password** command in Interface Configuration mode to configure the authentication password for the specified interface, perform the following task in interface configuration mode:

```
MOT(config-if)#isis password <password> {level-1 | level-2}
```

where:

password is the unencrypted text password that is 1-8 characters in length.

level-1 indicates a Level 1 IS-IS router.

level-2 indicates a Level 2 IS-IS router.

Gathering IS-IS Information

The following sections are used to gather information for your IS-IS network:

- Displaying IS-IS Database Information
- Displaying the Shortest Path First Log
- Displaying Connectionless Network Service Information

Displaying IS-IS Database Information

Use the following **show isis database** command options in Privileged EXEC mode to display all or specific IS-IS database information:

- Use the **show isis database detail** command to display detailed link state database information.
- Use the **show isis database level-1** command to display Level 1 IS-IS routing link state database.
- Use the **show isis database level-2** command to display Level 2 IS-IS routing link state database.
- Use the **show isis database detail** <*lspid*> command to display the link state protocol (LSP) identifier.
- Use the **show isis database** command to display all IS-IS database information.

Table 10-1 describes the **show isis database** command output fields:

Table 10-1 show isis database Command Output Field Descriptions

Output Field	Description
LSPID	LSP identifier.
LSP Seq Num	Sequence number for the LSP. Allows other systems to determine if they have received the latest information from source.
LSP Checksum	Checksum of the LSP packet.
LSP Holdtime	Number of seconds the LSP remains valid.
ATT	Attach bit. Indicates that router is a Level 2 router and can reach other areas.

Table 10-1 show isis database Command Output Field Descriptions

Output Field	Description
P	P bit. Detects if Intermediate System is capable of area partition repair.
OL	Overload bit. Determines if Intermediate System is congested.

Displaying the Shortest Path First Log

Use the **show isis spf-log** command in Privileged EXEC mode to display how often and why the router has run a full SPF calculation for the Level 1 and Level 2 IS-IS router, as shown below:

```
MOT#show isis spf-log
```

Table 10-1 describes the **show isis spf-log** command output fields:

Table 10-2 show isis spf-log Command Output Field Descriptions

Output Field	Description
When	The amount of time since a full SPF calculation took place given in hours:minutes:seconds. The previous 20 calculations are logged.
Duration	Number of milliseconds to complete this SPF run. The elapsed time is in actual clock time, not CPU time.
Nodes	Number of routers and pseudonodes (LANs) that make up the topology calculated in this SPF run.
Count	Number of events that triggered this SPF run. When there is a topology change, often multiple LSPs are received in a short time period. A router waits 5 seconds before running a full SPF run, so it can include all new information. This count denotes the number of events (such as receiving new LSPs) that occurred while the router was waiting its 5 seconds before running full SPF.
Last trigger LSP	Whenever a full SPF calculation is triggered by a new LSP, the LSP ID is stored in the router.
Triggers	Refer to Table 10-3 for a list of reasons that triggered a full SPF calculation.

Table describes a list of possible SPF triggers:

Table 10-3 Reasons for SPF Log Trigger Events

Trigger	Reason
PERIODIC	Typically, every 15 minutes a router runs a periodic full SPF calculation.
NEWSYSID	A new system ID through the NET was configured on this router.
NEWAREA	A new area (through NET) was configured on this router.
NEWLEVEL	A new level (through is-type) was configured on this router.
NEWMETRIC	A new metric was configured on an interface of this router.
IPBACKUP	An IP route disappeared, which was not learned through IS-IS, but through another protocol with better administrative distance. IS-IS will run a full SPF to install an IS-IS route for the disappeared IP prefix.
IPQUERY	A **clear ip route** command was issued on this router.
ATTACHFLAG	This router is now attached to the Level 2 backbone or it has just lost contact to the Level 2 backbone.
ADMINDIST	Another administrative distance was configured for the IS-IS process on this router.
AREASET	Set of learned area-addresses in this area changed.
NEWADJ	This router has created a new adjacency to another router.
DBCHANGED	A clear isis * command was issued on this router.
BACKUPOVFL	An IP prefix disappeared. The router knows there is another way to reach that prefix, but has not stored that backup route. The only way to find the alternative route is to run a full SPF run.
NEWLSP	A new router or pseudonode appeared in the topology.
LSPEXPIRED	Some LSP in the LSDB has expired.
LSPHEADER	ATT/P/OL bits or is-type in an LSP header changed.
TLVCODE	TLV code mismatch, indicating that different TLVs are included in the newest version of an LSP.
TLVCONTENT	TLV contents changed. This normally indicates that an adjacency somewhere in the area has come up or gone down. Look at the "Last trigger LSP" to get an indication of where the instability may have occurred.

Displaying Connectionless Network Service Information

The following options are accessed from Privileged EXEC mode to display Connectionless Network Service Information (CNSI) information:

- The **show clns es-neighbors** [**detail**] command displays End System neighbor adjacencies that the BSR knows.

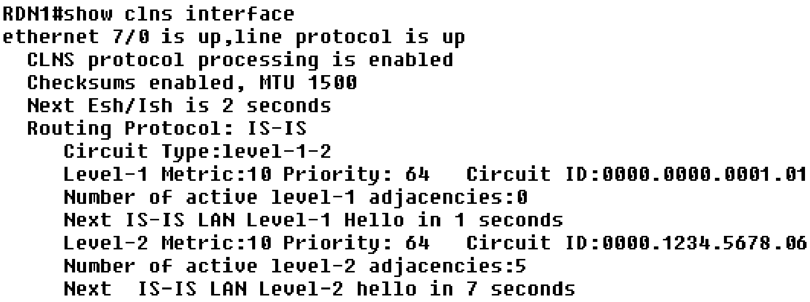
  Table 10-4 describes the **show clns es-neighbors** command output:

Table 10-4 show clns es-neighbors Command Output Fields

Output Field	Description
System ID	System ID of the IS-IS router.
Interface	Interface on which the router was discovered.
State	Adjacency state. Up and Init are the states of the ES or IS neighbor.
Type	Interface type of the neighboring ES router.

- The **show clns interface** command displays the CLNS interface status and configuration.

  Figure 10-1 displays sample **show clns interface** command output:



```
RDN1#show clns interface
ethernet 7/0 is up,line protocol is up
  CLNS protocol processing is enabled
  Checksums enabled, MTU 1500
  Next Esh/Ish is 2 seconds
  Routing Protocol: IS-IS
     Circuit Type:level-1-2
     Level-1 Metric:10 Priority: 64   Circuit ID:0000.0000.0001.01
     Number of active level-1 adjacencies:0
     Next IS-IS LAN Level-1 Hello in 1 seconds
     Level-2 Metric:10 Priority: 64   Circuit ID:0000.1234.5678.06
     Number of active level-2 adjacencies:5
     Next  IS-IS LAN Level-2 hello in 7 seconds
```

Figure 10-1 show clns interface Command Output

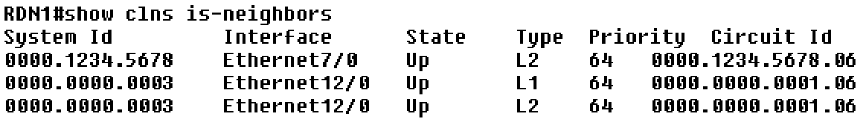
Table 10-5 describes the **show clns interface** Output Fields:

Table 10-5 show clns interface Output Fields

Output Field	Description
interface	The specific interface is described in the output and is described as being up (functional) and the line protocol as being up (functional) or administratively down.
CLNS protocol processing	Describes whether or not the CLNS protocol is enabled or disabled.
Checksums enabled	The checksums can be enabled or disabled.
MTU	The maximum transmission unit size for a packet on this interface.
Next Esh/Ish	Shows when the next ES hello or IS hello packet is sent on this interface.
Routing Protocol	Describes the routing protocol on this interface. Below this field, information for Level 1 and/or Level 2 is displayed.
Circuit Type	Indicates whether the interface has been configured for local routing (Level 1), area routing, (Level 2), or local and area routing (Level 1 and 2).
Metric	Indicates the routing metric assigned to the Level 1 or Level 2 router.
Priority	Indicates the priority of the IS on this interface.
Circuit ID	Indicates the ISIS circuit ID.
Number of active level-1 adjacencies	Indicates the number of active Level 1 adjacencies.
Number of active level-2 adjacencies	Indicates the number of active Level 2 adjacencies.

- The **show clns is-neighbors** command displays IS-IS related information for IS-IS router adjacencies.

  Figure 10-2 displays sample **show clns is-neighbors** command output:



```
RDN1#show clns is-neighbors
System Id         Interface      State    Type  Priority  Circuit Id
0000.1234.5678    Ethernet7/0    Up       L2    64    0000.1234.5678.06
0000.0000.0003    Ethernet12/0   Up       L1    64    0000.0000.0001.06
0000.0000.0003    Ethernet12/0   Up       L2    64    0000.0000.0001.06
```

Figure 10-2 show clns Command Output

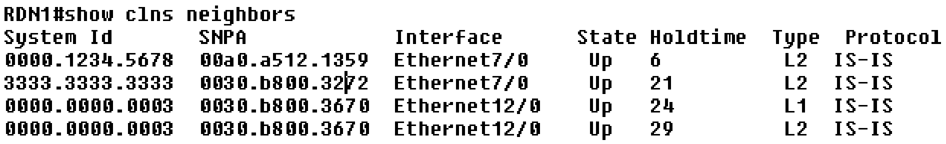
Table 10-6 describes the **show clns is-neighbors** command output:

Table 10-6 show clns is-neighbors Command Output Fields

Output Field	Description
System ID	System ID of the IS-IS router.
Interface	Interface on which the router was discovered.
State	Adjacency state. Up and Init are the states of the ES or IS neighbor.
Init	System is an IS and is waiting for an IS-IS hello message. IS-IS regards the neighbor as not adjacent.
Up	Believes the ES or IS is Reachable
Type	Displays whether the IS-IS router type is Level 1 or Level 2.
Priority	Indicates the routing priority.
Circuit ID	Indicates the ISIS circuit ID.

- The **show clns neighbors** command displays both ES and IS neighbors.

Figure 10-3 displays sample **show clns neighbors** command output:



```
RDN1#show clns neighbors
System Id       SNPA            Interface     State Holdtime  Type  Protocol
0000.1234.5678  00a0.a512.1359  Ethernet7/0    Up   6          L2   IS-IS
3333.3333.3333  0030.b800.3272  Ethernet7/0    Up   21         L2   IS-IS
0000.0000.0003  0030.b800.3670  Ethernet12/0   Up   24         L1   IS-IS
0000.0000.0003  0030.b800.3670  Ethernet12/0   Up   29         L2   IS-IS
```

Figure 10-3 show clns neighbors Command Output

Table 10-7 describes the **show clns neighbors** command output:

Table 10-7 show clns neighbors Command Output Fields

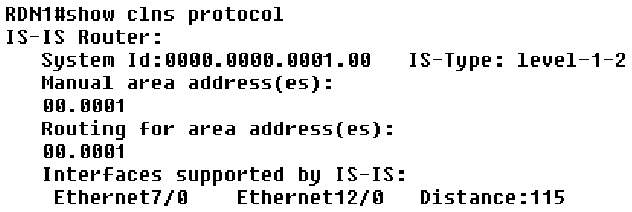
Output Field	Description
System ID	System ID of the IS-IS router.
SNPA	Subnetwork Point of Attachment, which is the data-link layer address.
Interface	Interface in which the system was learned from.
State	Adjacency state. Up and Init are the states of the ES or IS neighbor.

Table 10-7 show clns neighbors Command Output Fields

Output Field	Description
Init	System is an IS and is waiting for an IS-IS hello message. IS-IS regards the neighbor as not adjacent.
Up	Regards the ES or IS as reachable.
Holdtime	The number of seconds before this adjacency entry times out.
Type	The level of the IS-IS router.
Protocol	The protocol through which the adjacency was learned. Valid protocol sources are IS-IS, IGRP, or through a static route.

- The **show clns protocol** command displays the protocol-specific information for each IGRP routing process in the router.

  Figure 10-4 displays sample **show clns protocol** command output:



```
RDN1#show clns protocol
IS-IS Router:
   System Id:0000.0000.0001.00   IS-Type: level-1-2
   Manual area address(es):
   00.0001
   Routing for area address(es):
   00.0001
   Interfaces supported by IS-IS:
    Ethernet7/0    Ethernet12/0   Distance:115
```

Figure 10-4 show clns protocol Command Output

Table 10-8 describes the **show clns protocol** command output:

Table 10-8 show clns protocol Command Output Fields

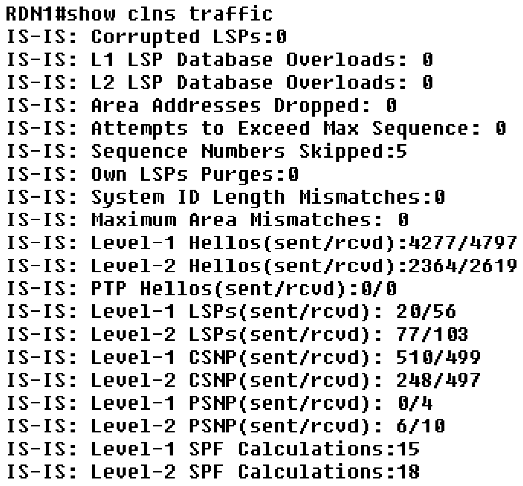
Output Field	Description
IS-IS Router	Indicates that the IS-IS protocol is enabled on the BSR.
System Id	Identification value of the system.
IS-Type:	Indicates the IS-IS routing level (Level 1, Level 2 or both) is enabled on the router.
Manual area address(es):	Area addresses that have been configured.

Table 10-8 show clns protocol Command Output Fields

Output Field	Description
Routing for area address(es):	List of manually configured and learned area addresses.
Interfaces supported by IS-IS:	List of interfaces on the BSR that support IS-IS.
Distance:	Configured IS-IS administrative routing distance.

- The **show clns traffic** command lists the CLNS packets the BSR has processed.

  Figure 10-5 displays sample **show clns traffic** command output:



```
RDN1#show clns traffic
IS-IS: Corrupted LSPs:0
IS-IS: L1 LSP Database Overloads: 0
IS-IS: L2 LSP Database Overloads: 0
IS-IS: Area Addresses Dropped: 0
IS-IS: Attempts to Exceed Max Sequence: 0
IS-IS: Sequence Numbers Skipped:5
IS-IS: Own LSPs Purges:0
IS-IS: System ID Length Mismatches:0
IS-IS: Maximum Area Mismatches: 0
IS-IS: Level-1 Hellos(sent/rcvd):4277/4797
IS-IS: Level-2 Hellos(sent/rcvd):2364/2619
IS-IS: PTP Hellos(sent/rcvd):0/0
IS-IS: Level-1 LSPs(sent/rcvd): 20/56
IS-IS: Level-2 LSPs(sent/rcvd): 77/103
IS-IS: Level-1 CSNP(sent/rcvd): 510/499
IS-IS: Level-2 CSNP(sent/rcvd): 248/497
IS-IS: Level-1 PSNP(sent/rcvd): 0/4
IS-IS: Level-2 PSNP(sent/rcvd): 6/10
IS-IS: Level-1 SPF Calculations:15
IS-IS: Level-2 SPF Calculations:18
```

Figure 10-5 show clns protocol Command Output

Table 10-9 describes the **show clns traffic** command output:

Table 10-9 show clns traffic Command Output Fields

Output Field	Description
Corrupted LSPs	The number of corrupted LSPs recorded on BSR.
L1 LSP Database Overloads	The number of times that the Level 1 LSP database has overloaded.
L2 LSP Database Overloads	The number of times that the Level 2 LSP database has overloaded.
Area Addresses Dropped	The number of area addresses dropped by the BSR.
Attempts to Exceed Max Sequence	The maximum sequence number is 2^{32} -1 the number of times the BSR reaches that number when the generating new LSPs.
Own LSPs Purges	The number of LSPs received, which have the same system ID as the BSR has.
System ID Length Mismatches	The number of IS-IS packets received, which have an ID length other than 6.
Maximum Area Mismatches	The number of IS-IS packets received, which have a maximum area number other than 3.
Level-1 Hellos (sent/rcvd)	Lists the number of Level 1 IS-IS hello packets sent and received.
Level-2 Hellos (sent/rcvd)	Lists the number of Level 2 IS-IS hello packets sent and received.
PTP Hellos (sent/rcvd)	Lists the number of point-to-point IS-IS hello packets sent and received.
Level-1 LSPs (sent/rcvd)	Lists the number of Level 1 link-state PDUs sent and received.
Level-2 LSPs (sent/rcvd)	Lists the number of Level 2 link-state PDUs sent and received.
Level-1 CSNP (sent/rcvd)	Lists the number of Level 1 CSNPs sent and received.
Level-2 CSNP (sent/rcvd)	Lists the number of Level 2 CSNPs sent and received.

Table 10-9 show clns traffic Command Output Fields

Output Field	Description
Level-1 PSNPs (sent/rcvd)	Lists the number of Level 1 PSNPs sent and received.
Level-2 PSNPs (sent/rcvd)	Lists the number of Level 2 PSNPs sent and received.
Level-1 SPF Calculations	Lists the number of times a Level 1 shortest -path-first (SPF) tree was computed.
Level-2 SPF Calculations	List the number of times a Level 2 SPF tree was computed.

11

Configuring OSPF

Overview

OSPF supports IP sub-networking and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets. It advertises the states of its local network links and runs within a single Autonomous System (AS) to determine optimum routes. Each participating OSPF router within the AS has an identical database of the AS topology. OSPF uses the database information to calculate a routing table by constructing a shortest-path tree. It recognizes AS topology changes and calculates new, loop-free routes.

OSPF requires coordination among many internal routers, area border routers (ABRs), and autonomous system boundary routers (ASBRs). Basic configuration of OSPF-based routers or access servers uses all default parameter values, no authentication, and interfaces assigned to areas. If you customize your environment, you must ensure the coordinated configuration of all routers.

To configure OSPF, complete the tasks in the following sections. You must perform the basic tasks, which include enabling OSPF and defining an OSPF area and area ID. The advanced tasks are optional, but include some parameters that you may choose to change. If a parameter default is satisfactory, you can ignore its associated task.

Use the following section to implement OSPF on the BSR 64000™:

- Enabling OSPF

The following optional tasks are used to manage OSPF on the BSR 64000™:

- Redistributing Routes into OSPF
- Configuring OSPF Area Parameters
- Managing OSPF on the BSR
- Configuring OSPF on an Interface
- Gathering OSPF Information

Specifications

The BSR supports the following Request for Comment (RFC) specifications:

RFC 2328 — *OSPF Version 2*

RFC 1587 — *The OSPF NSSA Option*

Enabling OSPF

To enable OSPF, create the OSPF routing process, specify the range of IP addresses associated with the routing process, and assign the area IDs associated with that range of IP addresses.

To create the OSPF routing process, perform the following steps:

1. To enable OSPF routing, use the **router ospf** command in Global Configuration mode, as shown below:

   `MOT(config)#`**router ospf**

   This enables OSPF routing and places you in Router Configuration mode.

2. To define an OSPF interface and define its area ID, use the **network area** command in Router Configuration mode, as shown below:

   `MOT(config-ospf)#`**network** <*ip-address*> <*mask*> **area** <*area-id*>

   where:

   ip-address is the IP address of the OSPF network.

   mask is the IP address type mask with *don't care* bits (wildcard mask).

   area-id is the network area ID, OSPF address range, either decimal value or IP address. If areas are associated with IP subnets, subnet area may be specified.

Example

The following example creates the OSPF routing process and adds two OSPF ranges (3.3.3.0/24 and 3.3.3.0/24) with each range belonging to a different area. Area 0 is configured for 3.3.3.0/24 and Area 1 is configured for 3.3.3.0/24.The example creates two OSPF interfaces. One interface is in the backbone area (Area 0) using IP address 3.3.3.1. The other interface is in the non-backbone area (Area 1) using IP address 3.3.3.1.

```
ip address 3.3.3.1 255.255.255.0
ip address 3.3.3.1 255.255.255.0
router ospf
```

network 3.3.3.0 0.0.0.255 area 0
network 3.3.3.3 0.0.0.255 area 1

Redistributing Routes into OSPF

Each routing protocol uses different metrics to transfer routes. Some protocols use hop count metrics, while others use bandwidth and delay attributes to define metrics. When a specific route is redistributed from one routing protocol or domain into another, a common metric must be applied by the receiving protocol. Routes are redistributed to advertise networks on another routing protocol.

Follow these steps to redistribute routes into OSPF:

1. Enter the OSPF routing process in which the routes are to be redistributed, as shown below:

   `MOT(config)#`**router ospf**

2. Choose from one or more of the following options to redistribute routes from a specified protocol:

   - Use the **redistribute bgp** command in Router Configuration mode to redistribute BGP routes into OSPF, as shown below:

     `MOT(config-ospf)#`**redistribute bgp** {**metric** <*n*> | **route-map** <*map-name*> | <*cr*>}

     where:

     metric <*n*> is the redistribution metric number for BGP routes.

     route-map <*map-name*> is the BGP route-map name.

     cr is a command return that redistributes of all BGP routes.

   - Use the **redistribute connected** command in Router Configuration mode to redistribute connected routes into OSPF, as shown below:

     `MOT(config-ospf)#`**redistribute connected** {**metric** <*n*> | **route-map** <*map-name*> | <*cr*>}

     where:

     metric <*n*> is the redistribution metric number for connected routes from 1 to 16.

route-map is the route-map name for the connected route.

cr is a command return that redistributes of all connected routes.

- Use the **redistribute isis** command in Router Configuration mode to redistribute IS-IS routes into OSPF, as shown below:

  `MOT(config-ospf)#`**redistribute isis** {**match** [**level-1** | **level-1-2** | **level-2**] | **metric** <*n*> | **route-map** <*map-name*> | <*cr*>}

  where:

  The **match** argument is used to choose level 1 ISIS routes only, level 1 and 2 ISIS routes, or level 2 ISIS routes only.

  metric <*n*> is the redistribution metric number for ISIS routes from 1 to 16.

  route-map <*map-name*> is the route-map name for the ISIS route.

  cr is a command return that redistributes of all ISIS routes.

- Use the **redistribute rip** command in Router Configuration mode to redistribute RIP routes into OSPF, as shown below:

  `MOT(config-ospf)#`**redistribute rip** {**metric** <*n*> | **metric-type** [**1** |**2**] | **route-map** <*map-name*> | **subnets** | **tag** | <*cr*>}

  where:

  metric <*n*> is the redistribution metric number for RIP routes from 1 to 16.

  metric-type 1 redistributes OSPF External Type 1 metrics.

  metric-type 2 redistributes OSPF External Type 2 metrics.

  route-map <*map-name*> is the route-map name for the OSPF route.

  subnets allows the consideration of RIP subnets for redistribution into OSPF.

  tag sets a tag for routes redistributed into OSPF.

  cr is a command return that redistributes of all OSPF routes.

Assigning a Default Metric Value for Redistributed Routes

The default metric function is used to eliminate the need for separate metric definitions for each routing protocol redistribution.

Follow these steps to assign a default metric value for all routes redistributed into OSPF:

1. Use the router ospf command to enter the OSPF routing process in Global Configuration mode, as shown below:

   ```
   MOT(config)#router ospf
   ```

2. Use the default-metric command in Router Configuration mode to force a routing protocol to use the same metric value for all distributed routes from other routing protocols, as shown below:

   ```
   MOT(config-ospf)#default-metric <n>
   ```

   where:

   n is the default metric value for all routes that are redistributed into OSPF.

Configuring OSPF Area Parameters

Use the following sections to configure OSPF area parameters:

- Configuring OSPF Area Authentication Parameters
- Configuring OSPF Stub Areas
- Configuring OSPF Not So Stubby Area
- Configuring Route Summarization between OSPF Areas
- Configuring Route Summarization into OSPF Area

Configuring OSPF Area Authentication Parameters

Use the steps and options to define area authentication parameters for your network:

1. Select the OSPF interface on which the OSPF authentication key password must be configured by using the **interface** command in Global Configuration mode.
2. Issue the **ip ospf authentication-key** command in Interface Configuration mode to assign a password on the routing interface for neighboring OSPF routers to use on a network segment that uses OSPF simple password authentication, as shown below:

```
MOT(config-if)#ip ospf authentication-key <password>
```

where:

password is the unencrypted (clear text) with 1 to 8 characters.

3. Exit Interface Configuration mode by using the **end** command.
4. Enter the OSPF router on the BSR, using the **router ospf** command in Global Configuration mode.
5. Use the **area authentication** command in Router Configuration mode to enable OSPF area authentication that permits (cleartext) password protection against unauthorized access to an area, as shown below:

   ```
   MOT(config-ospf)#area <area-id> authentication
   ```

   where:

   area-id is the area number.

   - or -

   Use the **area authentication message-digest** command in Router Configuration mode to enable OSPF area authentication that provides encrypted MD5 password protection against unauthorized access to an area, as shown below:

   ```
   MOT(config-ospf)#area <area-id> authentication message-digest
   ```

   where:

   area-id is the area number.

Configuring OSPF Stub Areas

Stub areas do not receive information on external routes. Instead, the Area Border Router (ABR) generates a default external route into the stub area for destinations outside the Autonomous System (AS). A stub area allows a default route, intra-area routes, and inter-area routes, but disallows autonomous system (AS) external routes, virtual links, and autonomous system boundary router (ASBR) routes.

Use the following steps to configure a stub area:

1. Issue the **router ospf** command in Global Configuration mode to enter OSPF Router Configuration mode.

2. Issue the **area stub** command in Router Configuration mode to configure an OSPF area as a stub area, as shown below:

Note: If there is more than one router within a stub area, ensure that the area that you are creating as a stub area is defined as a stub area on each of these routers.

```
MOT(config-ospf)#area <area-id> stub
```

where:

area-id is the OSPF area ID number.

Use the following options to further define your OSPF stub network:

- Use the optional **area stub no-summary** command in Router Configuration mode to prevent an area border router (ABR) from sending further Type 3 summary link-state advertisements (LSAs) into the stub area, as shown below:

  ```
  MOT(config-ospf)#area <area-id> stub no-summary
  ```

  where:

  area-id is the area number.

- Use the **area default-cost** command in Router Configuration mode to assign a specific cost to the default summary route sent into the stub area by an area border router (ABR) only, as shown below:

  ```
  MOT(config-ospf)#area {<area-id> | <ip-address>} default-cost <cost>
  ```

  where:

  area-id is the OSPF area ID number.

  ip-address is the IP address associated with the OSPF area ID.

  cost is the outgoing OSPF cost metric for packets sent from the interface, which is an unsigned 16-bit integer from 0 to 65535.

Configuring OSPF Not So Stubby Area

The Not So Stubby Area (NSSA) is similar to the OSPF stub area. The BSR does not flood Type 5 external LSAs from the backbone into the NSSA area, but it can import AS external routes in a limited fashion within the area. NSSA allows importing of Type 7 AS external routes within NSSA area by redistribution. These Type 7 LSAs are translated into Type 5 LSAs by NSSA ABRs and are flooded throughout the whole routing domain. Summarization and filtering are supported during the translation.

Use NSSA to simplify administration if you are an Internet Service Provider (ISP) or a network administrator and must connect a central site that uses OSPF to a remote site that uses a different routing protocol. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

1. To specify area parameters needed to configure OSPF NSSA, use the **area nssa** command in Router Configuration mode, as shown below:

   `MOT(config-ospf)#`**area** <*area-id*> **nssa** [**default-information-originate**] [**no-redistribution**] [**no-summary**] <*cr*>

   where:

   area-id identifies the NSSA.

   default-information-originate allows Type 7 LSAs to be imported into the NSSA.

   no-redistribution indicates no routes are redistributed to this NSSA.

   no summary disallows summary LSAs into the NSSA.

Note: A carriage return entered after the **area *area-id*** entry defines the area as an NSSA.

2. To control summarization and filtering of Type 7 LSA into Type 5 LSA during translation, use the optional **summary-address** in Router Configuration mode, as shown below. This command specifies an IP address and address mask that cover redistributed routes so that one summary route is advertised.

MOT(config-ospf)#**summary-address** <*ip-address*> <*mask*> [**tag**] <*num*>

where:

ip-address is the IP summary address.

mask is the IP summary address mask.

num is the 32-bit tag value for filtering externally derived routing information.

Example

The following example enables NSSA authentication on area 1:

```
router ospf
redistribute rip subnets
network 180.21.54.0. 0.0.0.255 area 1
area 1 nssa
```

Configuring Route Summarization between OSPF Areas

Route summarization causes an ABR to advertise a single summary route to other areas. An ABR advertises networks in one area to the backbone. If the network numbers in an area have contiguous assignments, you can configure the ABR to advertise a summary route that covers all the individual networks within the area that are in the specified range.

Follow these options to configure OSPF route summarization:

- Use the **area range advertise** command in Router Configuration mode to specify an address range for which a single route is advertised, as shown below:

  MOT(config-ospf)#**area** <*area-id*> **range** <*ip-address*> <*mask*> **advertise**

  where:

  area-id is the number or IP address for the area.

  ip-address is the IP address for an individual network within the area.

  mask is the subnet mask for the address.

  advertise indicates advertise the range.

- Use the **area range not-advertise** command in Router Configuration mode to specify an address range for a single route that is not advertised, as shown below:

  `MOT(config-ospf)#`**area** <*area-id*> **range** <*ip-address*> <*mask*> **not-advertise**

  where:

  > *area-id* is the number or IP address for the area.
  >
  > *ip-address* is the IP address for an individual network within the area.
  >
  > *mask* is the subnet mask for the address.
  >
  > **not-advertise** indicates do not advertise the range.

Example

This example configures an ABR to summarize the aggregate range 1.1.0.0/16.

```
ip address 2.2.10.1 255.255.255.0
ip address 2.1.11.1 255.255.255.0
router ospf
network 2.2.0.0 0.0.255.255 area 0

area 0 range 2.2.0.0 255.255.0.0
```

Configuring Route Summarization into OSPF Area

When redistributing routes from other protocols into OSPF, each route is advertised individually in an external LSA. However, you can configure the software to advertise a single route for all the redistributed routes that are covered by a specified network address and mask. Doing so helps decrease the size of the OSPF link state database and the routing table.

To specify an IP address and mask that covers redistributed routes so that only one summary route is advertised, use the **summary-address** command in Router Configuration mode, as shown below:

`MOT(config-ospf)#`**summary-address** <*ip-address*> <*mask*> **tag** <*tag-value*>

where:

> *ip-address* is the IP summary address.
>
> *mask* is the IP summary address subnet mask.

tag-value is the 32-bit tag value for filtering externally derived routing information.

Example

In the following example, summary address 20.1.0.0 includes address 20.1.1.0, 20.1.2.0, 20.1.3.0, and so forth. Only the address 20.1.0.0 is advertised in an external LSA.

```
summary-address 20.1.0.0 255.255.0.0
```

Managing OSPF on the BSR

This section discusses the following optional OSPF tasks:

- Establishing a Virtual Link
- Assign a Default Route for an ASBR
- Controlling OSPF Link Cost Metrics
- Allowing Dynamic OSPF Virtual Links
- Changing OSPF Administrative Distances
- Configuring Route Calculation Timers
- Blocking OSPF LSA Flooding

Establishing a Virtual Link

In OSPF, all areas must be connected to a backbone area. If there is a break in backbone continuity or the backbone is purposefully partitioned, you can establish a virtual link. The two endpoints of a virtual link are Area Border Routers (ABRs). The virtual link must be configured in both routers. The configuration information in each router consists of the other virtual endpoint (the other ABR), and the non-backbone area that the two routers have in common (the transit area).

Note: Virtual links cannot be configured through stub areas.

Choose from the following options to establish a virtual link that connects an OSPF area to the backbone area (area 0.0.0.0) without being physically connected to the OSPF backbone area:

- If you want to configure an OSPF virtual link that contains the default parameters for the **hello-interval**, **retransmit-interval**, **transmit-delay**, **dead-interval**, **authentication-key**, and **message-digest-key** parameters, use the **area virtual-link** command in Router Configuration mode, as shown below:

  `MOT(config-ospf)#`**area** {<*area-id*> | <*ip-address*>} **virtual-link** <*router-id*>

  where:

  area-id is the OSPF area IP address or number.

  ip-address is IP address associated with the OSPF area ID.

  <*router-id*> is the router ID 32-bit IP address associated with the virtual link neighbor.

- If you want to configure the time in seconds between hello packets on and interface for the OSPF virtual link, use the **area virtual-link hello-interval** command in Router Configuration mode, as shown below:

  `MOT(config-ospf)#`**area** {<*area-id*> | <*ip-address*>} **virtual-link** <*router-id*> **hello-interval <***seconds***>**

  where:

  area-id is the OSPF area IP address or number.

  ip-address is IP address associated with the OSPF area ID.

  <*router-id*> is the router ID 32-bit IP address associated with the virtual link neighbor.

  <*seconds*> is the hello interval value in seconds that must be the same for all routers and access servers attached to a common network.

- If you want to configure the expected round-trip delay Link State Advertisement (LSA) retransmit interval between two routers on the attached network for the OSPF virtual link, use the **area virtual-link retransmit-interval** command in Router Configuration mode, as shown below:

  `MOT(config-ospf)#`**area** {<*area-id*> | <*ip-address*>} **virtual-link** <*router-id*> **retransmit-interval** <*seconds*>

  where:

  > *area-id* is the OSPF area IP address or number.
  >
  > *ip-address* is IP address associated with the OSPF area ID.
  >
  > <*router-id*> is the router ID 32-bit IP address associated with the virtual link neighbor.
  >
  > <*seconds*> is the retransmission interval that is more than the expected delay from 1 to 65535 seconds.

- If you want to configure the Link State Advertisement (LSA) transmit delay between two routers on the attached network for the OSPF virtual link, use the **area virtual-link transmit-delay** command in Router Configuration mode, as shown below:

  `MOT(config-ospf)#`**area** {<*area-id*> | <*ip-address*>} **virtual-link** <*router-id*> **transmit-delay** <*seconds*>

  where:

  > *area-id* is the OSPF area IP address or number.
  >
  > *ip-address* is IP address associated with the OSPF area ID.
  >
  > <*router-id*> is the router ID 32-bit IP address associated with the virtual link neighbor.
  >
  > <*seconds*> is the approximate time in seconds to transmit an LSA packet.

- If you want to configure the interval that determines when the OSPF virtual link neighbor is down, use the **area virtual-link dead-interval** command in Router Configuration mode, as shown below:

  `MOT(config-ospf)#`**area** {<*area-id*> | <*ip-address*>} **virtual-link** <*router-id*> **dead-interval** <*seconds*>

where:

area-id is the OSPF area IP address or number.

ip-address is IP address associated with the OSPF area ID.

<*router-id*> is the router ID 32-bit IP address associated with the virtual link neighbor.

<*seconds*> is the number of seconds that the router does not receive hello packets from its neighbor before declaring the neighbor is down.

- If you want to set an unencrypted cleartext password for the OSPF virtual link, use the **area virtual-link authentication-key** command in Router Configuration mode, as shown below:

```
MOT(config-ospf)#area {<area-id> | <ip-address>} virtual-link <router-id>
authentication-key <password>
```

where:

area-id is the OSPF area IP address or number.

ip-address is IP address associated with the OSPF area ID.

<*router-id*> is the router ID 32-bit IP address associated with the virtual link neighbor.

<*password*> is the password that is 1 to 8 characters in length.

- If you want to set an encrypted password for the OSPF virtual link, use the **area virtual-link message-digest-key** command in Router Configuration mode, as shown below:

```
MOT(config-ospf)#area {<area-id> | <ip-address>} virtual-link <router-id>
message-digest-key <key-id>
```

where:

area-id is the OSPF area IP address or number.

ip-address is IP address associated with the OSPF area ID.

<*router-id*> is the router ID 32-bit IP address associated with the virtual link neighbor.

message-digest-key is the OSPF MD5 Authentication Key ID.

- Use the **show ip ospf virtual-links** command in Privileged EXEC mode to display information about the established virtual links, as shown below:

  MOT#**show ip ospf virtual-links**

- Use the **show ip ospf virtual links** command in Privileged EXEC mode to display the router ID of an OSPF router, as shown below:

  MOT#**show ip ospf virtual-links**

Example

The following example establishes a virtual link with default values for all optional parameters:

```
router ospf
network 72.0.0.0 0.255.255.255 area 72.0.0.0
area 72.0.0.0 virtual-link 72.4.5.6
```

The following example establishes a virtual link with MD5 authentication:

```
router ospf
network72.0.0.0 0.255.255.255 area 72.0.0.0
area 72.0.0.0 virtual-link 72.5.6 message-digest-key 3 md5 tag3665dr53
```

Assign a Default Route for an ASBR

Once routes are redistributed into an OSPF routing domain, the router becomes an Autonomous System Border Router (ASBR) that must be manually forced to generate a default route into the OSPF routing domain.

Use the **default-information originate** command in Router Configuration mode to force the ASBR to generate a default route into the OSPF routing domain, as shown below:

MOT(config-ospf)#**default-information originate** [**always**] [**metric** <*metric-value*>] [**metric-type** <*type-value*>]

where:

always indicates always advertise the default route even when the software does not have one.

metric is the metric for generating the default route; default is 10; valid values are from 0 to 16777214.

metric-value is the OSPF link state metric value; valid entries are 1 or 2.

Example

The following example specifies a metric of 100 for the default route redistributed into the OSPF routing domain and an external metric type of Type 1:

```
router ospf
redistribute rip metric 100 subnets
default-information originate always metric 100 metric-type1
```

Controlling OSPF Link Cost Metrics

The BSR OSPF routing process calculates the OSPF cost metric for an interface according to the bandwidth of this interface. The cost of an interface depends on the type of interface. The OSPF cost metric is calculated as the reference bandwidth divided by the bandwidth of the interface.

Use the **auto-cost reference-bandwidth** command in Router Configuration mode to set the automatic cost metric that the OSPF routing process uses to differentiate the cost of multiple high-bandwidth links, as shown below:

```
MOT(config-ospf)#auto-cost reference-bandwidth <ref-bw>
```

where:

ref-bw is a value from 1 to 4292967 Mbps.

Allowing Dynamic OSPF Virtual Links

Automatic detection and creation of OSPF links is disabled by default. Use the **auto-virtual-link** command in Router Configuration mode if you want to allow the OSPF routing process on the BSR to automatically detect and create OSPF virtual links.

```
MOT(config-ospf)#auto-virtual-link
```

Changing OSPF Administrative Distances

The administrative distance number between 0 and 255 rates the credibility of routing information from one or more routers. A routing source assigned a low administrative distance is trusted more than a routing source that is assigned a high administrative distance value.

Note: If the assigned administrative distance for a routing source is 255, it is not trusted and is ignored.

The default administrative distance for inter-area and intra-area OSPF routes, and external routes is 110. Use either the following options to set the administrative distance for OSPF routes:

- To set all three OSPF distances to the same value, use the **distance** command in Router Configuration mode, as shown below:

  ```
  MOT(config-ospf)#distance <distance>
  ```

  where:

  distance is the administrative distance; valid values are from 1 to 255.

Routes within an OSPF area are intra-area; routes to another OSPF area are inter-area; and routes from another routing domain learned through the redistribution of a route have an external designation.

- Use the **distance ospf** command in Router Configuration mode to set an individual value for the administrative distance value for an intra-area OSPF route, inter-area OSPF Route, and external route, as shown below:

  ```
  MOT(config-ospf)#distance ospf intra-area <distance> inter-area
  <distance> external <distance>
  ```

  where:

  intra-area <*distance*> represents the administrative distance number from 1 to 255 for all routes within an area.

inter-area <*distance*> represents the administrative distance number from 1 to 255 for all routes from one area to another area.

external <*distance*> represents the administrative distance number from 1 to 255 for routes learned by redistribution from other routing domains.

Configuring Route Calculation Timers

Timers are used by routing protocols to determine time intervals for when route information is adjusted:

Use the **timers spf** command in Router Configuration mode to configure the delay time after OSPF receives a topology change until it starts a shortest path first (SPF) calculation and hold time between two consecutive SPF calculations, as shown in the following example:

`MOT(config-ospf)#`**timers spf** <*spf-delay*> <*spf-holdtime*>

where:

spf-delay is the time delay from 0 to 65535 seconds between receiving a change to the SPF calculation.

spf-holdtime is the hold-time from 0 to 65535 seconds between consecutive SPF calculations.

Configuring OSPF on an Interface

The following sections are used to configure interface-specific parameters for OSPF and how to set up OSPF on a passive or loopback interface:

- Configuring General OSPF Interface Parameters
- Blocking OSPF LSA Flooding
- Configuring a Passive Interface for OSPF
- Forcing Router ID Choice with Loopback Interface

Configuring General OSPF Interface Parameters

You can change certain interface-specific OSPF parameters using the **ip ospf hello-interval**, **ip ospf dead-interval**, and **ip ospf authentication-key** commands. If you change these parameters, ensure that the configurations for all routers on your network have compatible values. You can change the parameters shown in Table 11-1:

Table 11-1 OSPF Parameters

Parameter	Description	Default	Values
cost	Metric value for sending a packet on an OSPF interface; the higher the bandwidth, the lower the cost	10^{8}/ bandwidth	1 to 65535
retransmit-interval	Time interval between LSA retransmissions	5 seconds	1 to 65535 seconds
transmit-delay	Time interval for LSA retransmissions	1second	1 to 65535 seconds
priority	Value used to determine OSPF designated router	1seconds	0 to 255 seconds
hello-interval	Time interval between hello packets	10 seconds	1 to 65535 seconds
dead-interval	Time interval between router hello packets after which neighboring routers consider the router down	40 seconds	1 to 65535 seconds
authentication-key	Password for use by neighboring OSPF routers that use OSPF simple password authentication	None	Character string up to 8 bytes
message-digest-key	Key enabling OSPF MD5 authentication	disabled	enabled or disabled

1. To specify the cost of sending a packet on an OSPF interface, use the **ip ospf cost** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#ip ospf cost <n>
```

where:

n is the OSPF path cost from 1 to 65535 Mbps.

2. To specify the number of seconds between link state advertisement (LSA) retransmissions for adjacencies belonging to an OSPF area, use the **ip ospf retransmit-interval** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip ospf retransmit-interval <seconds:1,65535>
   ```

3. To set the estimated number of seconds to transmit a link state update packet on an OSPF interface, use the **ip ospf transmit-delay** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip ospf transmit-delay <seconds:1,65535>
   ```

4. To set priority to help determine the OSPF designated router for a network, use the **ip ospf priority** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip ospf priority <num>
   ```

   where:

   num is a value between 0 and 255.

5. To specify the length of time between the hello packets that the software sends on an OSPF interface, use the **ip ospf hello-interval** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip ospf hello-interval <seconds:1,65535>
   ```

6. To set the number of seconds that hello packets must be absent before the device neighbors declare the OSPF router *down*, use the **ip ospf dead-interval** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip ospf dead-interval <seconds>
   ```

   where:

   seconds represents the number of seconds; valid values are between 1 and 65535; default is 40.

7. To assign a password for neighboring OSPF routers to use on a network segment that uses OSPF simple password authentication, use the **ip ospf authentication-key** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#ip ospf authentication-key <password>
   ```

where:

password is the unencrypted (clear text) with 1 to 8 characters.

8. To enable OSPF MD5 authentication, use the **ip ospf message-digest-key** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#ip ospf message-digest-key <key-id> md5 <key>
```

Examples

The following example sets the cost of sending a packet over an interface to 30:

```
interface ethernet 7/0
ip ospf cost 30
```

The following example sets the time interval between transmissions of an LSA to 20 seconds on an interface:

```
interface ethernet 7/0
ip ospf restransmit-interval 20
```

The following example sets the time it takes to transmit a link state update to 10seconds on an interface:

```
interface ethernet 7/0
ip ospf transmit-delay 10
```

The following example sets the router priority value to 10 on an interface:

```
interface ethernet 7/0
ip ospf priority 10
```

The following example sets the interval between hello packets to 15 seconds on an interface:

```
interface ethernet 7/0
ip ospf hello-interval 30
```

Blocking OSPF LSA Flooding

The OSPF LSA age indicates whether the LSA is valid. The LSA is discarded when it reaches the maximum age of one hour. During the aging process, the originating router sends a refresh packet every 30 minutes to keep the LSA from expiring, regardless of network topology changes. The router tracks and refreshes the LSAs it generates; it tracks and ages the LSAs it receives from other routers. Each LSA is refreshed when it is 30 minutes old, independent of other LSAs.

By default, OSPF floods new LSAs over all interfaces in the same area, except the interface on which the LSA arrives. Some redundancy is desirable, because it ensures robust flooding. However, too much redundancy can waste bandwidth and might destabilize the network due to excessive link and CPU usage in certain topologies.

To block the flooding of OSPF LSAs on broadcast, non-broadcast, and point-to-point networks, use the **ip ospf database-filter all out** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#ip ospf database-filter all out
```

Examples

The following example prevents flooding of OSPF LSAs to broadcast, non-broadcast, or point-to-point networks accessible through Ethernet interface 7/0:

```
interface ethernet 7/0
ip ospf database-filter all out
```

Configuring a Passive Interface for OSPF

To prevent OSPF from flooding an interface, you can configure the interface as a passive network. This prevents OSPF from sending hello packets for that interface. Both devices are able to see each other via the hello packet generated for the receiving interface.

You can configure a passive interface to prevent other routers on a local network from learning about routes dynamically. A passive interface does not transmit routing updates.

Use the **passive-interface** command in Router Configuration mode to create a passive OSPF interface, as shown in the following example:

`MOT(config-ospf)#`**passive-interface** {**cable** | **ethernet** | **gigaether** | **pos** | **serial**} *<slot>*/*<interface>*

where:

cable is the cable interface.

ethernet is the Ethernet/Fast Ethernet interface.

gigaether is the Gigabit Ethernet interface.

pos is the Packet over SONET interface.

serial is the Serial interface.

slot is the module slot number.

interface is the interface number.

Forcing Router ID Choice with Loopback Interface

OSPF uses the largest IP address configured on the interfaces as its router ID. If the interface associated with this IP address goes down, or if the address is removed, the OSPF process must recalculate a new router ID and send again all its routing information.

If a loopback interface is configured with an IP address, the software uses this IP address as its router ID, even if other interfaces have larger IP addresses. Since loopback interfaces never go down, this provides greater stability in the routing table. OSPF automatically prefers a loopback interface over any other kind, and it chooses the highest IP address among all loopback interfaces. If no loopback interfaces are present, the highest IP address in the router is chosen.

To configure an IP address on a loopback interface, use the following commands:

1. To create a loopback interface and move to Interface Configuration mode, use the **interface loopback** command in Global Configuration mode, as shown below:

   `MOT(config)#`**interface loopback 1**

2. To assign an IP address to this interface, use the **ip address** command in Cable Interface Configuration mode, as shown below:

   `MOT(config-if)#`**ip address** *<A.B.C.D> <A.B.C.D>*

Example

The following example configures a loopback interface:

```
interface loopback 1
ip address 10.10.10.1 255.255.255.255
```

Gathering OSPF Information

There are several **show** commands that can be used to view and gather information about your OSPF network. These show commands are available in all command modes except for User EXEC mode, and most user access groups.

Use the following sections to gather information for your OSPF network:

- Displaying OSPF Routing Information
- Displaying OSPF Memory Information

Displaying OSPF Routing Information

Use the following sections to display OSPF routing information:

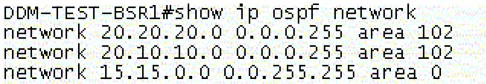
- Showing Network Information
- Showing Border Routers
- Showing Neighboring Routers
- Showing Virtual Links

Showing Network Information

Use the **show ip ospf network** command to display the IP network addresses, wildcard masks, and the area numbers for all OSPF areas, as shown below:

```
MOT#show ip ospf network
```

The following command output displays:



```
DDM-TEST-BSR1#show ip ospf network
network 20.20.20.0 0.0.0.255 area 102
network 20.10.10.0 0.0.0.255 area 102
network 15.15.0.0 0.0.255.255 area 0
```

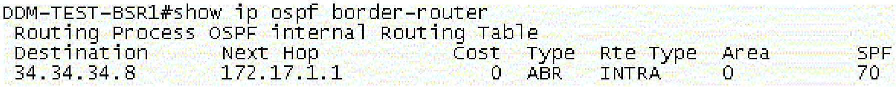
Figure 11-1 show ip ospf network Command Output

Showing Border Routers

Use the **show ip ospf border-routers** command to display the autonomous system boundary router (ASBR) and an area border router (ABR) routing tables.

`MOT#`**show ip ospf border-routers**

The following command output displays:



```
DDM-TEST-BSR1#show ip ospf border-router
 Routing Process OSPF internal Routing Table
 Destination       Next Hop             Cost   Type   Rte Type   Area          SPF
 34.34.34.8        172.17.1.1              0   ABR    INTRA      0             70
```

Figure 11-2 show ip ospf border-routers Command Output

Showing Neighboring Routers

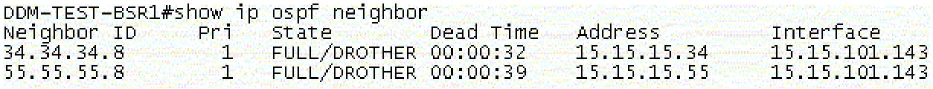
Use the following options to display OSPF neighbor information:

- Use the **show ip ospf neighbor** command to display information about all OSPF neighbors, as shown below:

Note: The **show ip ospf neighbor** can be accessed by the ISP user group.

`MOT#`**show ip ospf neighbor**

The following command output displays:



```
DDM-TEST-BSR1#show ip ospf neighbor
Neighbor ID      Pri   State         Dead Time    Address          Interface
34.34.34.8         1   FULL/DROTHER  00:00:32     15.15.15.34      15.15.101.143
55.55.55.8         1   FULL/DROTHER  00:00:39     15.15.15.55      15.15.101.143
```

Figure 11-3 show ip ospf neighbor Command Output

- or -

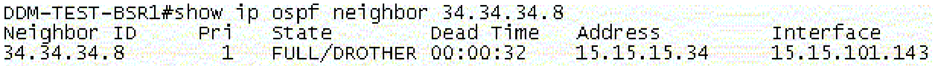
- Use the **show ip ospf neighbor** command to display a specific OSPF neighbor by entering its IP address, as shown below:

  MOT#**show ip ospf neighbor** <*neighbor-ip-address*>

  where:

  neighbor-ip-address is the OSPF neighbor IP address.

  The following command output displays:



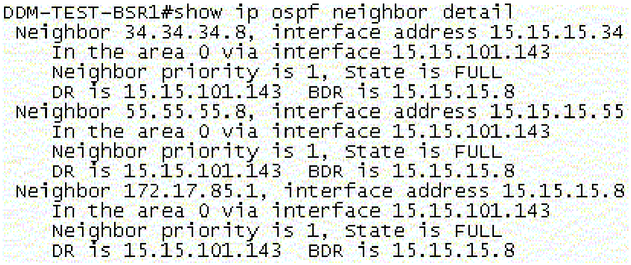
```
DDM-TEST-BSR1#show ip ospf neighbor 34.34.34.8
Neighbor ID      Pri   State         Dead Time    Address          Interface
34.34.34.8         1   FULL/DROTHER  00:00:32     15.15.15.34      15.15.101.143
```

Figure 11-4 show ip ospf neighbor Command Output (With Neighbor IP Address)

- Use the **show ip ospf neighbor detail** command to display detailed information for all OSPF neighbors, as shown below:

  MOT#**show ip ospf neighbor detail**

  The following command output displays:



```
DDM-TEST-BSR1#show ip ospf neighbor detail
 Neighbor 34.34.34.8, interface address 15.15.15.34
    In the area 0 via interface 15.15.101.143
    Neighbor priority is 1, State is FULL
    DR is 15.15.101.143  BDR is 15.15.15.8
 Neighbor 55.55.55.8, interface address 15.15.15.55
    In the area 0 via interface 15.15.101.143
    Neighbor priority is 1, State is FULL
    DR is 15.15.101.143  BDR is 15.15.15.8
 Neighbor 172.17.85.1, interface address 15.15.15.8
    In the area 0 via interface 15.15.101.143
    Neighbor priority is 1, State is FULL
    DR is 15.15.101.143  BDR is 15.15.15.8
```

Figure 11-5 show ip ospf neighbor detail Command Output

Showing Virtual Links

The **show ip ospf virtual-links** command displays parameters regarding the current state of the OSPF virtual links.

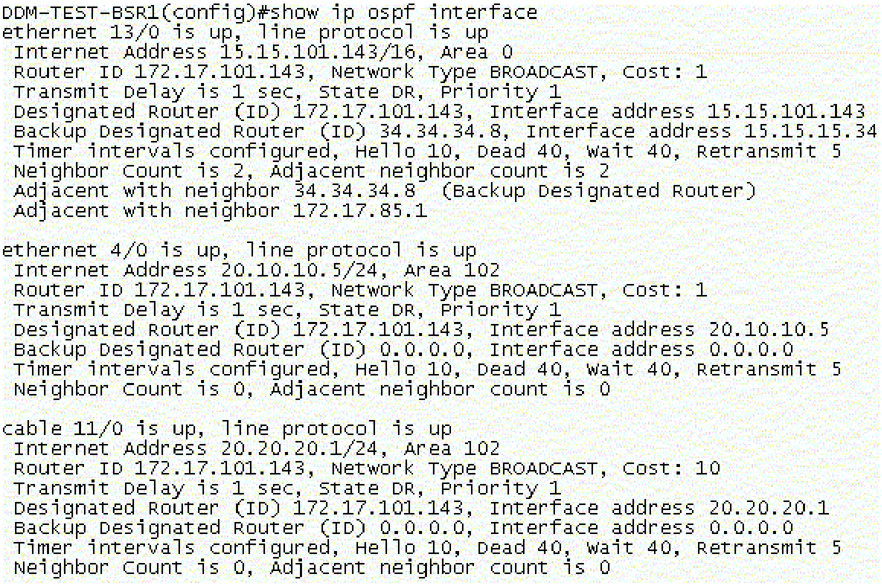
MOT#**show ip ospf virtual-links**

Displaying OSPF Interface Information

Use the following options to display OSPF interface information:

- Use the **show ip ospf interface** command to display information about all interfaces on which OSPF is configured.

  The following command output displays:



```
DDM-TEST-BSR1(config)#show ip ospf interface
ethernet 13/0 is up, line protocol is up
 Internet Address 15.15.101.143/16, Area 0
 Router ID 172.17.101.143, Network Type BROADCAST, Cost: 1
 Transmit Delay is 1 sec, State DR, Priority 1
 Designated Router (ID) 172.17.101.143, Interface address 15.15.101.143
 Backup Designated Router (ID) 34.34.34.8, Interface address 15.15.15.34
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
 Neighbor Count is 2, Adjacent neighbor count is 2
 Adjacent with neighbor 34.34.34.8  (Backup Designated Router)
 Adjacent with neighbor 172.17.85.1

ethernet 4/0 is up, line protocol is up
 Internet Address 20.10.10.5/24, Area 102
 Router ID 172.17.101.143, Network Type BROADCAST, Cost: 1
 Transmit Delay is 1 sec, State DR, Priority 1
 Designated Router (ID) 172.17.101.143, Interface address 20.10.10.5
 Backup Designated Router (ID) 0.0.0.0, Interface address 0.0.0.0
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
 Neighbor Count is 0, Adjacent neighbor count is 0

cable 11/0 is up, line protocol is up
 Internet Address 20.20.20.1/24, Area 102
 Router ID 172.17.101.143, Network Type BROADCAST, Cost: 10
 Transmit Delay is 1 sec, State DR, Priority 1
 Designated Router (ID) 172.17.101.143, Interface address 20.20.20.1
 Backup Designated Router (ID) 0.0.0.0, Interface address 0.0.0.0
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
 Neighbor Count is 0, Adjacent neighbor count is 0
```

Figure 11-6 show ip ospf interface Command Output

  - or -

- Use the **show ip ospf interface** command with the known IP address of the OSPF interface, as shown below:

MOT#**show ip ospf interface** <*ip-address*>

where:

ip-address is the interface IP address.

- or -

- Use the **show ip ospf interface** command with modifiers to display information about individual interfaces on which OSPF is configured, as shown below:

MOT#**show ip ospf interface** {**cable** | **ethernet** | **gigaether** | **loopback** <*ln*> | **pos** | **tunnel** <*tn*>} <*slot*>/<*interface*>

where:

cable is the CMTS interface.

ethernet is the Ethernet/FastEthernet IEEE 802.3 interface.

gigaether is the Gigabit Ethernet interface.

loopback is the loopback interface.

ln is the loopback interface number from 1 to 16.

pos is the Packet Over SONET interface.

tunnel is the tunnel interface.

tn is the tunnel interface number from 0 to 255.

slot is the module slot number.

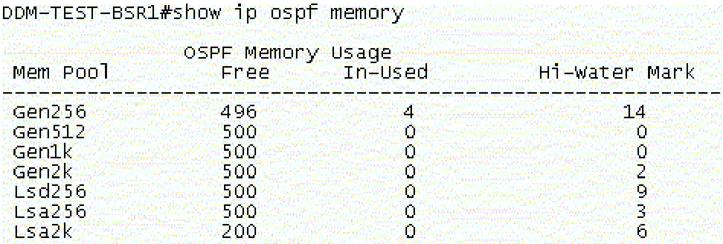
interface is the interface number.

Displaying OSPF Memory Information

The **show ip ospf memory** command displays OSPF memory usage information.

MOT#**show ip ospf memory**

The following command output displays:



```
DDM-TEST-BSR1#show ip ospf memory

               OSPF Memory Usage
 Mem Pool            Free      In-Used          Hi-Water Mark
----------------------------------------------------------------
 Gen256              496          4                      14
 Gen512              500          0                       0
 Gen1k               500          0                       0
 Gen2k               500          0                       2
 Lsd256              500          0                       9
 Lsa256              500          0                       3
 Lsa2k               200          0                       6
```

Figure 11-7 show ip ospf memory Command Output

Displaying OSPF Database Information

Use the following options to display OSPF neighbor information

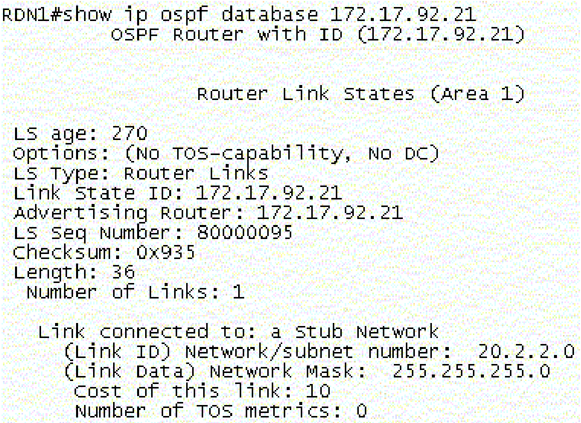
- Use the **show ip ospf database** command to display information for a specific OSPF router, as shown below:

  `MOT#`**show ip ospf database** <*ip-address*>

  where:

  ip-address is the link state ID or IP address of the OSPF router.

  The following output displays when you enter the IP address for an OSPF router:



```
RDN1#show ip ospf database 172.17.92.21
         OSPF Router with ID (172.17.92.21)


                Router Link States (Area 1)

 LS age: 270
 Options: (No TOS-capability, No DC)
 LS Type: Router Links
 Link State ID: 172.17.92.21
 Advertising Router: 172.17.92.21
 LS Seq Number: 80000095
 Checksum: 0x935
 Length: 36
  Number of Links: 1

   Link connected to: a Stub Network
     (Link ID) Network/subnet number:  20.2.2.0
     (Link Data) Network Mask:  255.255.255.0
      Cost of this link: 10
      Number of TOS metrics: 0
```

Figure 11-8 show ip ospf database Command Output Using an OSPF Router IP Address

- Use the **show ip ospf database adv-router** command to view the Link State Advertisements (LSAs) for the advertising router, as shown below:

  MOT#**show ip ospf database adv-router** <*ip-address*>

  where:

  ip-address is the IP address of the advertising router.

- Use the **show ip ospf database asbr-summary** command to view Autonomous System Boundary Router (ASBR) summary link states, as shown below:

  MOT#**show ip ospf database asbr-summary** <*ip-address*>

  where:

  ip-address is the link state identifier (IP address) of the ASBR.

- Use the **show ip ospf database asbr-summary adv-router** command to display summary link state information for Advertising Router link states, as shown below:

  MOT#**show ip ospf database asbr-summary adv-router** <*ip-address*>| **self-originate**}

  where:

  ip-address is the link state identifier (IP address) of the ASBR.

- Use the **show ip ospf database asbr-summary self-originate** command to display summary link state information for self-originating link states for an ASBR, as shown below:

  MOT#**show ip ospf database asbr-summary adv-router** <*ip-address*>| **self-originate**}

  where:

  ip-address is the link state identifier (IP address) of the ASBR.

- Use the **show ip ospf database external** command to display external LSAs, as shown below:

  MOT#**show ip ospf database external** [<*ip-address*> | **adv-router** | **self-originate** | <*cr*>

where:

ip-address is the IP address of the specific link-state ID.

self-originate displays LSAs from the local router.

cr displays all external LSAs.

- Use the **show ip ospf database**
- Use the **show ip ospf database**
- Use the **show ip ospf database**

show ip ospf database [**external** | **network** | **nssa-external** | **router** | **summary**] [<*link-state-id*>] [**self-originate**]

network	network LSAs
nssa-external	NSSA external LSA information
router	router LSAs
summary	summary LSAs
link-state-id	router links, link state ID always the same as the advertising router, network IP address, value dependent upon advertisement LSA type
self-originate	LSAs from the local router

12

Configuring BGP

Overview

This chapter describes how to configure Border Gateway Protocol (BGP) for the BSR 64000™ system using the command line interface (CLI). For a complete description of the CLI commands discussed in this chapter, refer to the *BSR 64000 Command Reference Guide*. This chapter discusses the following topics:

- About BGP
- Configuring Basic BGP Connectivity
- Configuring Advanced BGP Connectivity
- Configuring Global BGP Tasks
- Configuring BGP Update Flows
- Configuring Routing Policy
- Handling Access Lists
- Creating a Community List
- Redistributing Routes into BGP

About BGP

BGP, an Exterior Gateway Protocol (EGP) allows you to set up an inter-domain routing system that automatically guarantees the loop-free exchange of routing information between autonomous systems (ASs). An AS is a set of routers that use a single routing policy running under a single technical administration. An AS runs Interior Gateway Protocols (IGPs) such as Routing Information Protocol (RIP), and Open Shortest Path First (OSPF) within its boundaries.

BGP is used by almost all routers to connect ASs to network backbones. With BGP, each route comprises a network number, a list (AS path) of ASs that information passed through, and a list of other path attributes. A BGP system exchanges network reachability information with other BGP systems, including AS path information. This information allows routing loops to be pruned and AS-level policy decisions to be enforced.

Routers that belong to the same AS and exchange BGP updates run *internal* BGP (IBGP). Routers that belong to different ASs and exchange BGP updates run *external* BGP (EBGP). With few exceptions, the commands for configuring IBGP and EBGP are identical. Figure 12-1 shows exchanges with IBGP and EBGP running between routers.

BGP minimizes routing traffic outside ASs and manages the peer relationship between border routers that connect ASs within a backbone of the network infrastructure. IGPs concentrate on finding the shortest or quickest route between endpoints within an AS; BGP is typically used between ASs. The Multi-Exit Discriminator (MED) metric attribute value is configured using route maps. Updates sent to an IBGP peer also include unchanged MED information, enabling all **peer**s in the same AS to make a consistent path selection.

BGP supports classless inter-domain routing (CIDR). This allows reduction of the routing table sizes by creating aggregate routes, which result in supernets. CIDR eliminates the concept of network classes within BGP and supports the advertising of IP prefixes. CIDR routes are also carried by OSPF and RIP.

Note: Set up your router with BGP if your AS or router is dual or multihomed (if it has two or more backbone connections, either direct or daisy-chained thorough another point of presence). Also, use BGP if the BSR provides IP routing to a downstream site or customer or if you must preserve AS path information in your network. If your site is single-homed and you do not provide IP services or AS paths, consider using static routes instead. Because external routing requirements are relatively simple, static routing is easier to set up and to maintain, and it requires less overhead.

BGP uses AS path information to prevent routing loops. In BGP, each AS delineates the route in the path. This enables routers to look for loops by examining the information sent to them about the path. Figure 12-1 shows how BGP advertises routes to neighbors in an AS path.

1. Router Boston originates a route to Router New York.
2. Router New York forwards the route to Router Los Angeles, after adding its AS to the AS path.
3. Router Los Angeles receives the route and ascertains that it comes from another AS.

4. Router Los Angeles adds its own AS to the AS path and forwards to Router Miami.
5. Router Miami receives the route and ascertains that it comes from another AS.
6. Router Miami adds its own AS to the AS path and forwards the route to Router Dallas.
7. Router Dallas receives the route, determines that the route is a loop since its own AS is contained in the AS path, and discards the route.

You can configure BGP operating parameters manually. BGP selects among different routes by comparing specific path attributes or metrics for each route. The local administrator can configure each of these and assign them different values.

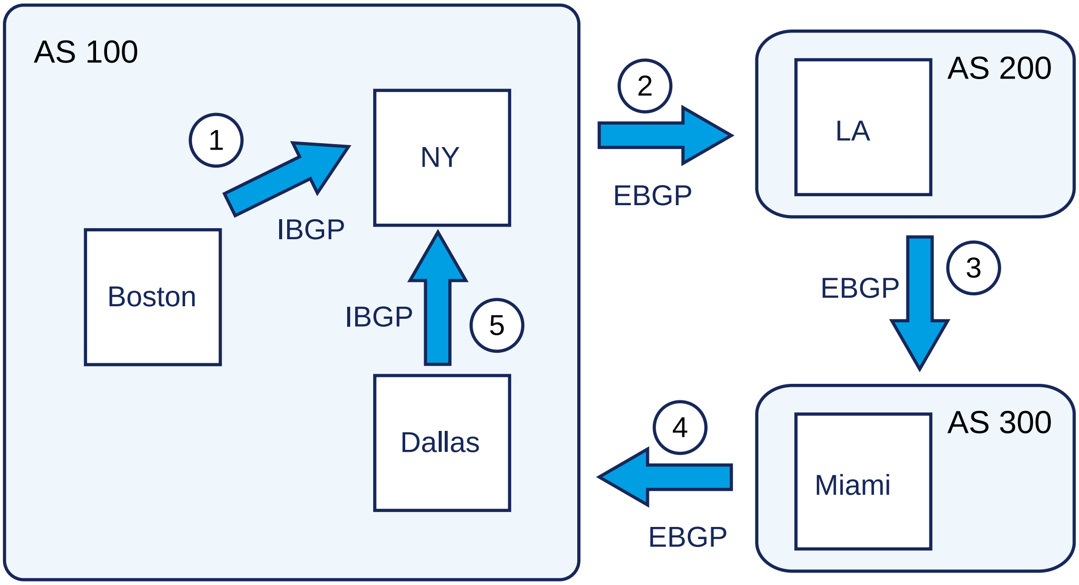


Figure 12-1 Advertising BGP Routes

BGP Peers

BGP provides a means for BGP peers, or neighbors, to exchange routing information within an AS (IBGP) and with peers within other ASs (EBGP). Information is exchanged between peers about the following:

- New active routes and their attributes
- Inactive routes
- Unusual conditions that require connection termination

BGP does not require routing information to be refreshed. Advertised route information is considered valid by its neighbors until the first router explicitly advertises that the information is no longer valid or until the BGP session is lost.

BGP Updates

BGP routers exchange routing information in the form of BGP updates. BGP updates contain the following attributes associated with routes that a BGP peer advertises to its neighbors:

- A list of ASs the routing update passed through
- The AS routing update origin
- Next hop information
- Metrics specifying route preference

BGP Sessions

After exchanging a series of messages, the BGP peers establish a session over TCP. BGP session partners rely on TCP to manage the underlying connection. Once a TCP connection is established, a BGP router uses port 179 to communicate full routing information with another BGP peer. As long as the connection is up, the BGP partners can exchange a very simple set of messages with minimal overhead.

The BGP protocol includes the exchange of *keep-alive* messages between peers. A keep-alive message is a signal from one endpoint to another, indicating that the first end point is still active. Keep-alive messages are necessary to keep BGP peers aware of the health of the connection, because TCP does not provide this service.

Specifications

The BSR supports the following Request for Comment (RFC) specifications:

- RFC 1771 — *A Border Gateway Protocol 4 (BGP-4*)
- RFC 1745 — *BGP4/IDRP for IP (OSPF Interaction)*
- RFC 1965 — *Autonomous System Confederations for BGP*
- RFC 1966 — *BGP Route Reflection, an Alternative to Full Mesh IBGP*
- RFC 1997 — *BGP Communities Attributes*
- RFC 1998 — *An Application of the BGP Community Attribute in Multi-home Routing*
- RFC 2439 — *BGP Route Flap Damping*
- RFC 2385 — *Protection of BGP Sessions via the TCP MD5 Signature Options*

Configuring Basic BGP Connectivity

BGP provides a means to organize the connectivity of BGP neighbors and peer groups. The following are basic BGP connectivity configuration tasks:

- Configuring a BGP neighbor
- Advertising networks in an AS

Configuring a BGP Neighbor

Follow these steps to configure a BGP neighbor:

1. To find the AS configured for BGP, use the **show running-config** command in Privileged EXEC mode, as shown below:

   ```
   MOT#show running-config
   ```

2. To enter Router BGP Configuration mode, use the **router bgp** command in Global Configuration mode, as shown below:

   ```
   MOT(config)#router bgp <n>
   ```

   where:

n is the Autonomous System (AS) to which the neighbor belongs; valid values are 1 to 65535.

3. To add an entry to the BGP neighbor table, use the **neighbor remote-as** command in Router BGP Configuration mode, as shown below. The BGP neighbor table identifies a router as a BGP peer and maps its IP address to a specific AS.

   MOT(config-bgp)#**neighbor** {*ip-address* | *peer-group*} **remote-as** *number*

   where:

   ip-address is the neighbor IP address.

   peer-group is the name of the BGP peer group.

   number is the AS to which the neighbor belongs.

4. To associate a textual description with a BGP neighbor, use the **neighbor description** command in Router BGP Configuration mode, as shown below. <>

   MOT(config-bgp)#**neighbor** {<*ip-address*> | <*name*>} **description** *text*

   where:

   ip-address is the IP address of the neighbor.

   name is the name of the BGP peer group.

   text is up to 80 characters of text that describes the neighbor.

Example

The following commands configure Routers Miami with Routers Chicago, Boston, and New York as neighbors (as shown in Figure 12-2):

```
MOT(config-bgp)#router bgp 100
MOT(config-bgp)#neighbor 172.30.20.2 remote-as 100
MOT(config-bgp)#neighbor 172.30.20.2 description peer_New York
MOT(config-bgp)#neighbor 172.40.20.2 remote-as 100
MOT(config-bgp)#neighbor 172.40.20.2 description peer_Chicago
MOT(config-bgp)#neighbor 192.50.30.2 remote-as 300
MOT(config-bgp)#neighbor 192.50.30.2 description peer_Boston
MOT(config-bgp)#network 120.20.0.0
```

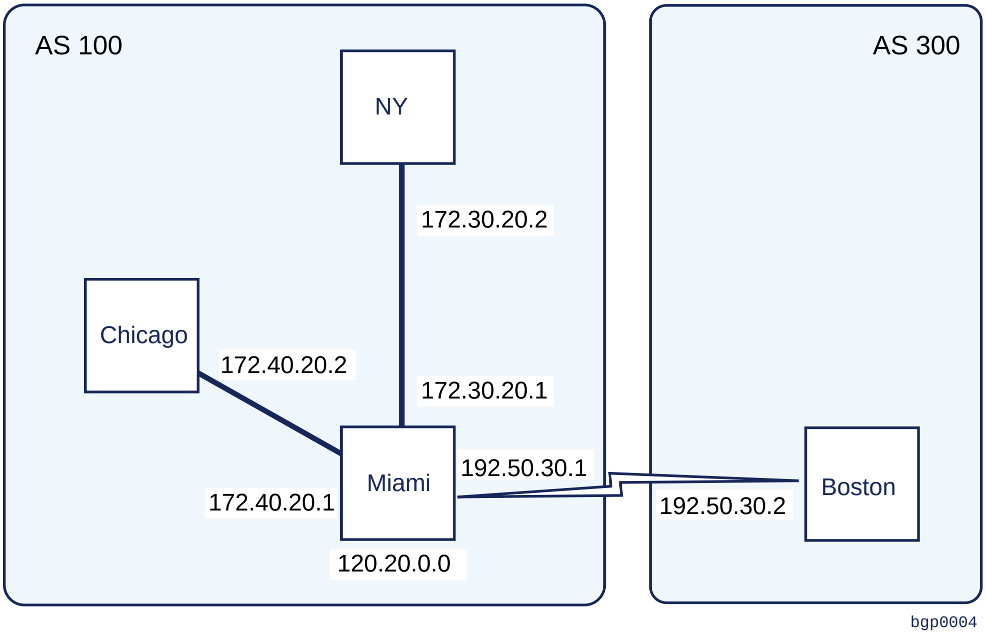


Figure 12-2 Configuring BGP Neighbors

Advertising Networks in an AS

1. To inform BGP peers in other ASs about the networks, advertise them using the **network** command in Router BGP Configuration mode, as shown below. The **network** command specifies the networks that an AS originates.

   `MOT(config-bgp)#`**network** <*network-number*> [*mask* <*network-mask*>]

   where:

   network-number is the network that BGP advertises.

   mask is the keyword for specifying network or subnetwork mask.

   network-mask is the network mask.

Example

Figure 12-3 shows how Routers Miami, Chicago, and Los Angeles advertise networks in their ASs.

The following commands configure Router Miami:

```
MOT(config-bgp)#router bgp 100
MOT(config-bgp)#neighbor 2.2.2.2 remote-as 100
MOT(config-bgp)#network 120.60.0.0
```

The next commands configure Router Chicago:

```
MOT(config-bgp)#router bgp 300
MOT(config-bgp)#neighbor 3.3.3.2 remote-as 100
MOT(config-bgp)#network 162.24.0.0
```

These commands configure Router Los Angeles:

```
MOT(config-bgp)#router bgp 400
MOT(config-bgp)#neighbor 2.2.2.1 remote-as 100
MOT(config-bgp)#neighbor 3.3.3.1 remote-as 300
MOT(config-bgp)#network 162.56.0.0
```

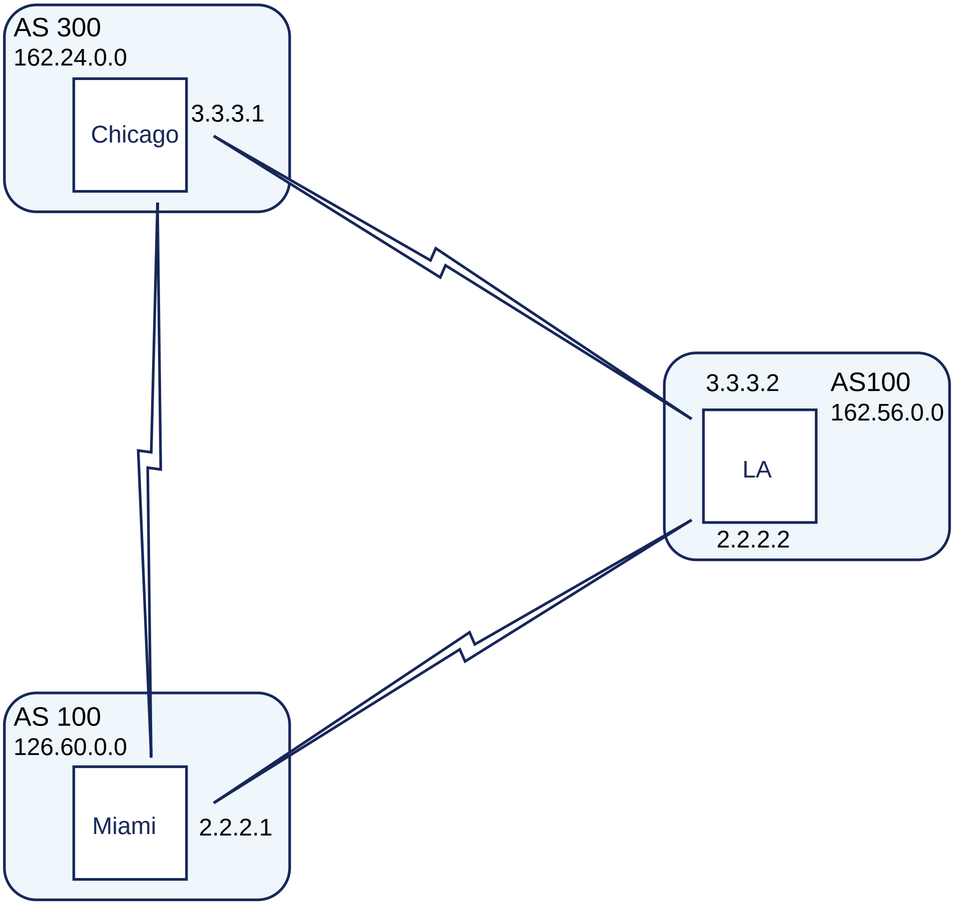


Figure 12-3 Advertising Networks in an AS

Configuring Advanced BGP Connectivity

The following are advanced BGP connectivity configuration tasks:

- Configuring BGP Peer Groups
- Configuring a Routing Domain Confederation
- Configuring a Route Reflector

- Restoring Route Reflection from a Route Reflection Client
- Configuring Route-flap Dampening
- Shutting Down a Neighbor or Peer Group
- Enabling MD5 Authentication Between Peers
- Setting the Minimum Interval for Sending BGP Routing Updates to Neighbors or Peer Groups
- Enabling EBGP Multihop for Neighbor and Peer Groups
- Controlling the Number of Prefixes Received from a Neighbor
- Configuring Next Hop Processing

Configuring BGP Peer Groups

Routing policies are usually defined by route maps, filter lists, and distribution lists. You can define a BGP peer group that assigns the same set of routing policies to a group of BGP peers (or neighbors). You can also use peer groups to override configuration options for incoming updates.

Figure 12-4 shows two peer groups. The first peer group contains the routers in AS 100. The second peer group contains Routers Philadelphia, Trenton, and Boston.

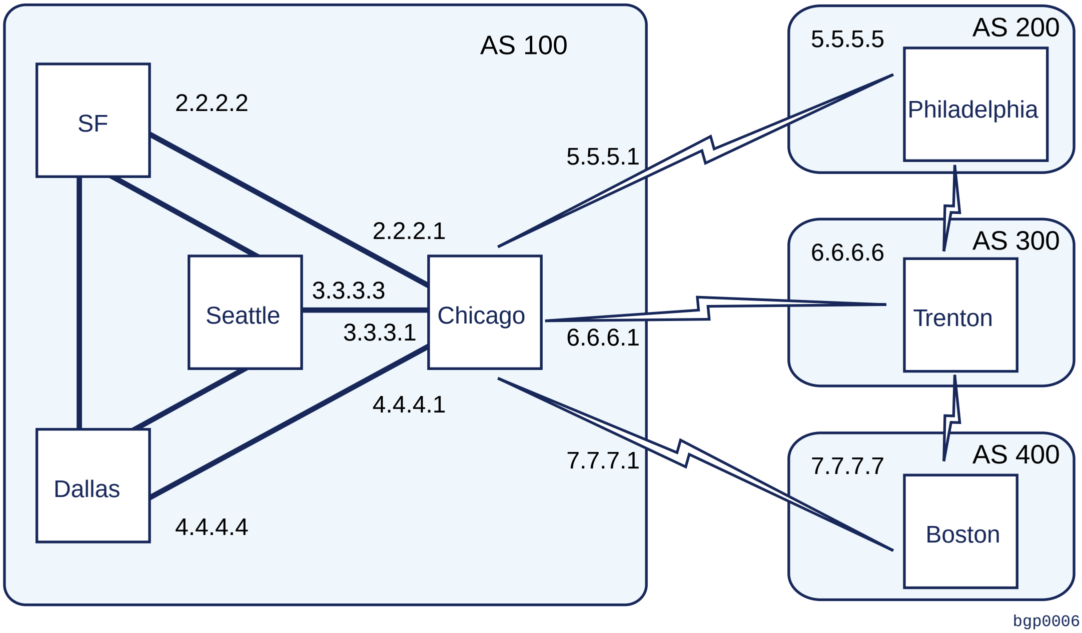


Figure 12-4 BGP Peer Groups

You can create a BGP peer group or you can configure a BGP neighbor to be a member of a BGP peer group using the **neighbor peer-group** command.

1. To create a BGP peer group, use the **neighbor peer-group** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**neighbor** <*name*> **peer-group**

   where:

   name is the name you assign to the peer group.

2. To assign a BGP neighbor as a member of a BGP peer group, use the **neighbor peer-group** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`neighbor <*ip-address*> peer-group <*name*>

   where:

   ip-address is the IP address of the neighbor to be assigned to the peer group.

name is the name you assign to the peer group.

Examples

The commands in the following example configure a BGP peer group on Router Chicago and apply it to Routers San Francisco, Dallas, and Seattle (as shown in Figure 12-4):

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor PACIFIC peer-group
MOT(config-bgp)#neighbor PACIFIC remote-as 100
MOT(config-bgp)#neighbor 2.2.2.2 peer-group PACIFIC
MOT(config-bgp)#neighbor 3.3.3.3 peer-group PACIFIC
MOT(config-bgp)#neighbor 4.4.4.4 peer-group PACIFIC
```

The commands in the following example configure a BGP peer group on Router Chicago and apply it to Routers Philadelphia, Trenton, and Boston (as shown in Figure 12-4):

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor ATLANTIC peer-group
MOT(config-bgp)#neighbor 5.5.5.5 peer-group ATLANTIC
MOT(config-bgp)#neighbor 5.5.5.5 remote-as 200
MOT(config-bgp)#neighbor 6.6.6.6 peer-group ATLANTIC
MOT(config-bgp)#neighbor 6.6.6.6 remote-as 300
MOT(config-bgp)#neighbor 7.7.7.7 peer-group ATLANTIC
MOT(config-bgp)#neighbor 7.7.7.7 remote-as 400
```

Configuring a Routing Domain Confederation

You can reduce the IBGP mesh inside an AS by creating a BGP confederation. In Figure 12-5, AS 400 consists of ten BGP neighbors that, without confederations, require that the routers be fully meshed. Each of the ten routers run IBGP with the other nine routers and connect to an external AS. Using a confederation, you reduce the number of peers required with AS 400. Each AS within AS 400 (AS 100, AS 200, and AS 300) must be fully meshed and IBGP run between its members. ASs outside AS 400 recognize the confederation as one AS, that is, AS 400.

Although the peers in different ASs within the same confederation have EBGP sessions, they exchange routing information as if they were IBGP peers. Specifically, the next-hop and local preference information is preserved. This lets you retain a single IGP for all the ASs in the confederation. To the outside, the confederation looks like a single AS.

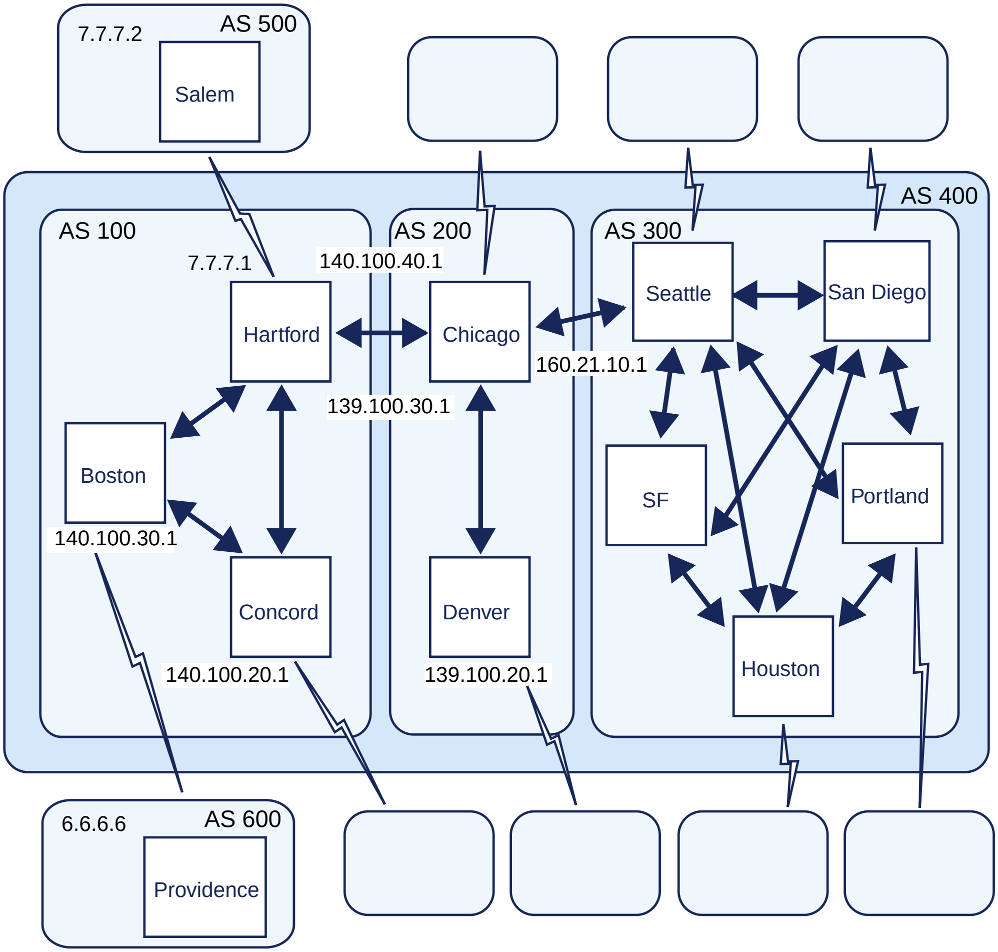


Figure 12-5 Configuring a Routing Domain Confederation

1. To specify a BGP confederation identifier, use the **bgp confederation identifier** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**bgp confederation identifier** <*num*>

   where:

num is the AS number that internally includes multiple ASs.

2. To configure an AS to be a member of the confederation, use the **bgp confederation peers** command in Router BGP Configuration mode, as shown below:

   MOT(config-bgp)#**bgp confederation peers** <*num*>

   where:

   num specifies the AS that is a member of the confederation.

Example 1

The commands in this example configure Router Hartford (as shown in Figure 12-5) and specify that the Router Hartford belongs to Confederation 400. They also specify that Router Hartford belongs to a confederation containing AS 200 and AS 300 as its peer ASs, specify that connections with confederation peers in AS 100 and AS 200, and specify an external peer in AS 500.

```
MOT(config)#router bgp 100
MOT(config-bgp)#bgp confederation identifier 400
MOT(config-bgp)#bgp confederation peers 200 300
MOT(config-bgp)#neighbor 140.100.30.1 remote-as 100
MOT(config-bgp)#neighbor 140.100.20.1 remote-as 100
MOT(config-bgp)#neighbor 139.100.30.1 remote-as 200
MOT(config-bgp)#neighbor 7.7.7.2 remote-as 500
```

Example 2

The commands in this example configure Router Chicago (as shown in Figure 12-5):

```
MOT(config)#router bgp 200
MOT(config-bgp)#bgp confederation identifier 400
MOT(config-bgp)#bgp confederation peers 100 300
MOT(config-bgp)#neighbor 140.100.40.1 remote-as 100
MOT(config-bgp)#neighbor 139.100.20.1 remote-as 200
MOT(config-bgp)#neighbor 160.21.10.1 remote-as 300
```

Configuring a Route Reflector

A BGP speaker cannot advertise a route to an IBGP neighbor if that BGP speaker originally heard the route from another IBGP speaker. The result of the this rule requires a full mesh of IBGP sessions within an AS to fully distribute routes via IBGP. If an AS has many BGP speakers, the number of peer connections can become very large.

A *route reflector* alleviates this problem. A route reflector is a BGP speaker that learns routes from an IBGP neighbor and advertises the routes to other IBGP neighbors. A *route reflector client* is a router within the same AS that depends on a router reflector to readvertise its routes to the entire AS and to learn about routes from the rest of the AS.

Figure 12-6 shows how a route reflector works. Without Router Los Angeles as a route reflector, the network requires a full IBGP mesh and that Router New York is a peer of Router Boston.

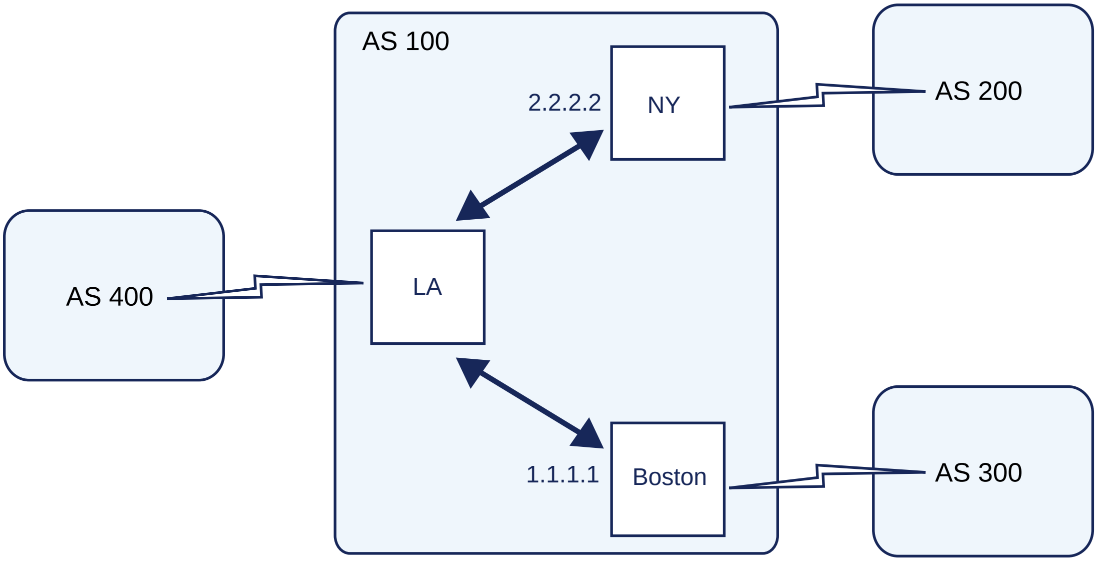


Figure 12-6 A BGP Route Reflector

Figure 12-7 shows that an AS can have more than one route reflector. Each route reflector considers other router reflectors as non-clients. You can configure multiple route reflectors per cluster and multiple clusters per AS.

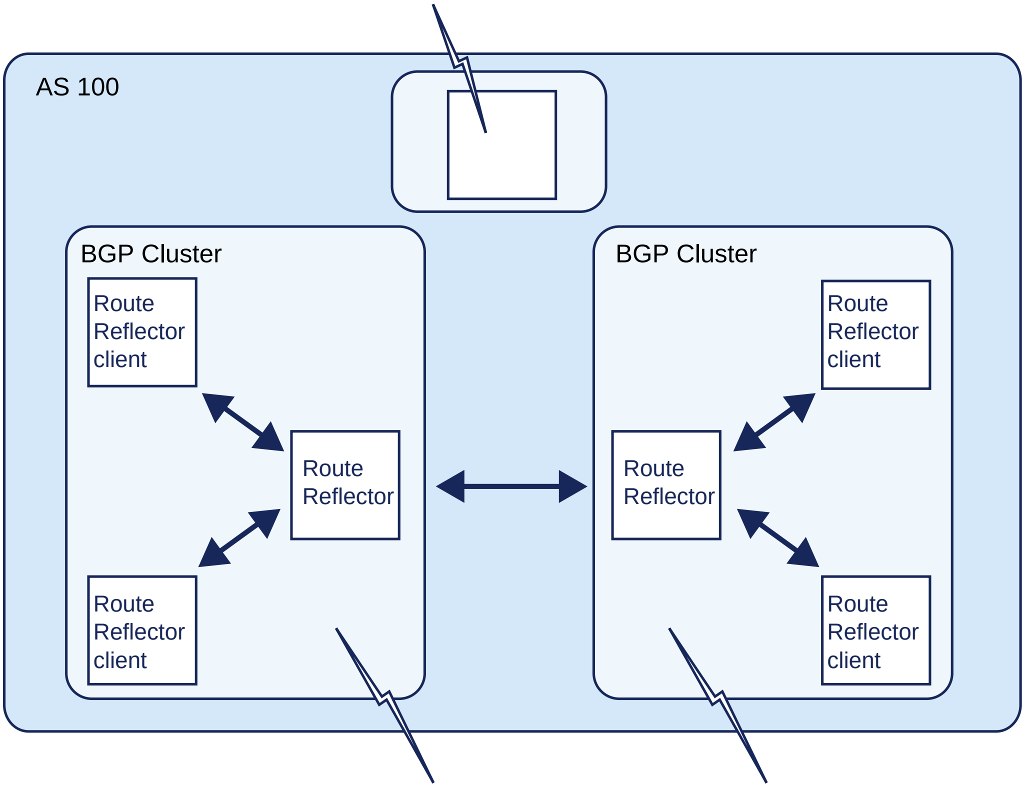


Figure 12-7 Multiple Route Reflectors

To specify route-reflector-clients for a route reflector, use the **neighbor route-reflector-client** command in Router BGP Configuration mode, as shown below:

`MOT(config-bgp)#`**neighbor** [<*ip-address*> |<*name*>] **route-reflector-client**

where:

ip-address is the IP address of the BGP neighbor that is the route reflector client.

name is the BGP neighbor peer group name that is the route reflector client.

Example

The commands in the following example configure Routers New York and Boston as route reflector clients (as shown in Figure 12-6):

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 1.1.1.1 remote-as 100
MOT(config-bgp)#neighbor 1.1.1.1 route-reflector-client
MOT(config-bgp)#neighbor 2.2.2.2 remote-as 100
MOT(config-bgp)#neighbor 2.2.2.2 route-reflector-client
```

Configuring a Cluster-ID

A route reflector and its clients form a *cluster*. Usually a cluster of clients has a single route reflector. The cluster is identified by the router ID of the route reflector.

If the cluster has more than one route reflector, use the **bgp cluster-id** command in Router BGP Configuration mode to configure the cluster ID:

`MOT(config-bgp)#`**bgp cluster-id** <*num*>

where:

num is the number of the cluster ID; valid entries are 0 – 4294967295.

Example

In the following example, the local router is one of the route reflectors serving the cluster. The example configures the local router with the cluster ID to identify the cluster.

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 156.80.20.12 route-reflector-client
MOT(config-bgp)#bgp cluster-id 40000
```

Restoring Route Reflection from a Route Reflection Client

By default, the clients of a route reflector are not required to be fully meshed and the routes from a client are reflected to other clients. However, if the clients are fully meshed, route reflection with the same set of clients is not required. Also, if client-to-client reflection is enabled, the clients of a route reflector cannot be members of a peer group.

In Figure 12-8, the local router is a route reflector. The four neighbors are fully meshed, so client-to-client reflection is disabled.

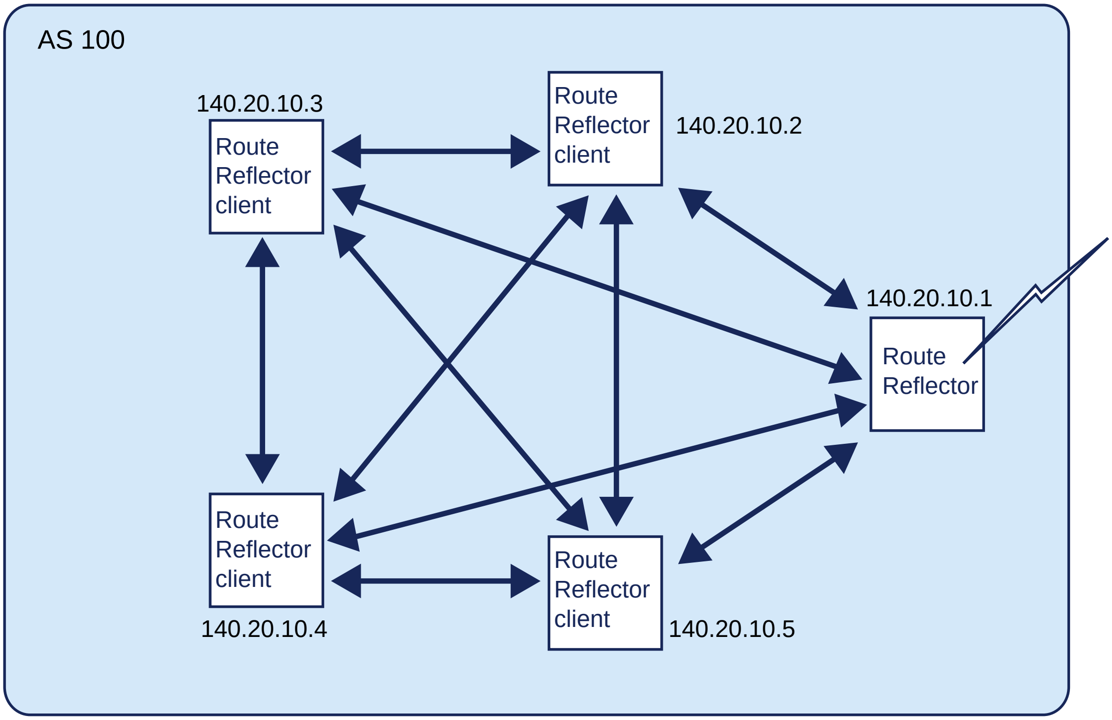


Figure 12-8 Disabling Client-to-Client Reflection

To disable client-to-client route reflection, use the **no bgp client-to-client reflection** command in Router BGP Configuration mode, as shown below:

```
MOT(config-bgp)#no bgp client-to-client reflection
```

Example

The commands in the following example show four configured route-reflector-clients for a router acting as a route reflector (as shown in Figure 12-8). The **no bgp client-to-client reflection** command disables client-to-client reflection because the clients are fully meshed.

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 140.20.10.2 route-reflector-client
MOT(config-bgp)#neighbor 140.20.10.3 route-reflector-client
MOT(config-bgp)#neighbor 140.20.10.4 route-reflector-client
MOT(config-bgp)#neighbor 140.20.10.5 route-reflector-client
MOT(config-bgp)#no bgp client-to-client reflection
```

Configuring Route Flap Dampening

The BSR supports two types of route flap dampening.

- Global
- Policy-based

Route flapping occurs when a link constantly fluctuates between being available and unavailable. When a link changes its availability, the upstream neighbor sends an update message to all its neighbors. These routes are advertised globally. This process continues until the underlying problem is fixed.

Route flap dampening is a mechanism for minimizing instability caused by route flapping. A penalty value for a route is increased by 1000 if the route flaps and is decreased by half after 15 minutes. Once the penalty exceeds the suppress limit of 2000, the route is no longer advertised to neighbors. (The route is *damped*.) When the penalty for a damped route falls below the reuse limit of 750, the route is again available.

Global Route Flap Dampening

1. To enable global route flap dampening with default values on all BGP routes, use the **bgp dampening** command in Router BGP Configuration mode, as shown below:

   ```
   MOT(config-bgp)#bgp dampening
   ```

2. To configure individual route flap dampening parameters, use the **bgp dampening** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**bgp dampening** [<*half-life*> <*reuse*> <*suppress*> <*max-suppress-time*>]

   where:

   half-life is the half-life period in minutes; valid values are 1 to 45; default is 15.

   reuse is the reuse penalty limit below which dampened routes become available again; valid values are 1 to 20000; default is 750.

   suppress is the penalty limit above which a flapping route is suppressed; valid values are 1 to 20000; default is 2000.

   max-suppress-time is the maximum suppression time in minutes; valid values are 1 to 255.

Example

The following command enables global route flap dampening:

```
MOT(config-bgp)#bgp dampening 5 1000 1500 15
```

Policy-based Route Flap Dampening

To filter specific routes for route flap dampening in a route map, use the **set dampening** route map command in Router BGP Configuration mode, as shown below:

`MOT(config-bgp)#`**set dampening** <*half-life*> <*reuse*> <*suppress*> <*max-suppress-time*>

where:

half-life is the half-life period in minutes in the range 1-45. The default is 15.

reuse is the reuse limit in the range 1-20000. The default is 750.

suppress is the suppress limit in the range 1-20000. The default is 2000.

max-suppress-time is the maximum suppression time in minutes in the range 1-255. The default is four times the half-life.

Example

The following commands create the route map:

```
MOT(config)#router bgp 100
MOT(config-bgp)#bgp dampening route-map dallas
MOT(config-bgp)#route-map dallas permit 10
MOT(config-bgp)#ip as-path access-list 1
MOT(config-bgp)#set dampening 5 1000 1500 15
```

These commands specify AS path access list 1 as the filter to determine the permitted ASs.

```
MOT(config)#ip as-path access-list 1 deny ^300
MOT(config)#ip as-path access-list 1 permit any
```

Clearing Route Flap Dampening

1. To clear all route dampening information, use the **clear ip bgp dampening** command in Privileged EXEC mode, as shown below:

   MOT#**clear ip bgp dampening**

2. To unsuppress a suppressed route, use the **clear ip bgp dampening** command in Router BGP Configuration mode, as shown below:

   MOT(config-bgp)#**clear ip bgp dampening** <*address*> <*mask*>

   where:

   address is the network IP address.

   mask is the network mask applied to the address.

Example

The following example clears route dampening information about the route to network 170.0.0.0 and unsuppresses its suppressed routes. If you do not specify the address and mask arguments, the **clear ip bgp dampening** command clears route dampening information for the entire BGP routing table.

```
MOT(config-bgp)#clear ip bgp dampening 170.0.0.0 255.255.0.0
```

Shutting Down a Neighbor or Peer Group

1. To terminate any active session for a specified BGP neighbor or peer group and remove all associated routing information, use the **neighbor shutdown** command in Router BGP Configuration mode, as shown below:

   ```
   MOT(config-bgp)#neighbor {<ip-address> | <name>} shutdown
   ```

   where:

   ip-address is the IP address of the neighbor.

   name is the name of the peer group.

Note: In the case of a peer group, use of the **neighbor shutdown** command may suddenly terminate a large number of peering sessions.

2. To view a summary of BGP neighbors and peer-group connections, use the **show ip bgp summary** command in Privileged EXEC mode as shown below. Disabled neighbors have an *Idle* status and *Admin* entry.

   ```
   MOT(config-bgp)#show ip bgp summary
   ```

Examples

The following example terminates the active session for the neighbor 156.40.20.23:

```
MOT(config-bgp)#neighbor 156.40.20.23 shutdown
```

The following example terminates all peering sessions for the peer group PACIFIC:

```
MOT(config-bgp)#neighbor PACIFIC shutdown
```

Enabling Message Digest 5 Authentication Between Peers

You can enable Message Digest 5 (MD5) authentication between two BGP peers, causing each segment sent on the TCP connection between them to be verified. You must configure the same password on both BGP peers; otherwise, the connection between them is not made. The authentication feature uses the MD5 algorithm command that causes the generation and checking of the MD5 digest on every segment sent on the TCP connection. Configuring a password for a neighbor terminates an existing session and establishes a new one. If you specify a BGP peer group using the *name* argument, all the members of the peer group inherit the characteristic configured with this command.

To enable MD5 authentication, use the **neighbor password** command in Router BGP Configuration mode, as shown below:

MOT(config-bgp)#**neighbor** {<*ip-address*> | <*name*>} **password** <*string*>

where:

ip-address is the IP address of the BGP-speaking neighbor.

name is the name of the BGP peer group.

string is a case-sensitive password of up to 80 alphanumeric characters. The first character cannot be a number.

Example

The commands in the following example enable the authentication feature between a router and the BGP neighbor at 122.35.3.1. The password that must also be configured for the neighbor is *mypassword*.

```
MOT(config)#router bgp 109
MOT(config-bgp)#neighbor 122.35.3.1 password mypassword
```

Setting the Routing Updates Interval

To set the minimum interval between the sending of BGP routing updates to neighbors or peer groups, use the **neighbor advertisement-interval** command in Router BGP Configuration mode. Lower values for the advertisement interval cause route changes to be reported more quickly. However, this may cause the routers to use more bandwidth.

MOT(config-bgp)#**neighbor** {<*ip-address*> | <*name*>} **advertisement-interval** <*seconds*>

where:

ip-address is the IP address of the neighbor.

name is the name of the BGP peer group.

seconds is the advertisement interval time in seconds; valid values are 0 to 600.

Example

The commands in the following example set the minimum time between sending BGP routing updates to 4 seconds:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 3.3.3.3 advertisement-interval 4
```

Enabling EBGP Multihop for Neighbor and Peer Groups

Normally, EBGP neighbors are directly connected. When EBGP neighbors do not connect directly, use the **neighbor ebgp-multihop** command to specify that the neighbor is more than one hop away.

MOT(config-bgp)#**neighbor** {<*ip-address*> | <*name*>} **ebgp-multihop** [<*ttl*>]

where:

ip-address is the IP address of the BGP neighbor.

name is the name of the BGP peer group.

ttl is the time-to-live in the range 1 – 255 hops.

Example

The commands in this example configure Router Miami, as shown in Figure 12-9:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 1.1.1.1 remote-as 200
MOT(config-bgp)#neighbor 1.1.1.1 ebgp-multihop
```

Figure 12-9 shows Router Miami is configured with Router Washington as an external peer. Because Router Miami and Router Washington are connected together via Router Boston, rather than by a direct link, the **neighbor ebgp-multihop** command is used.

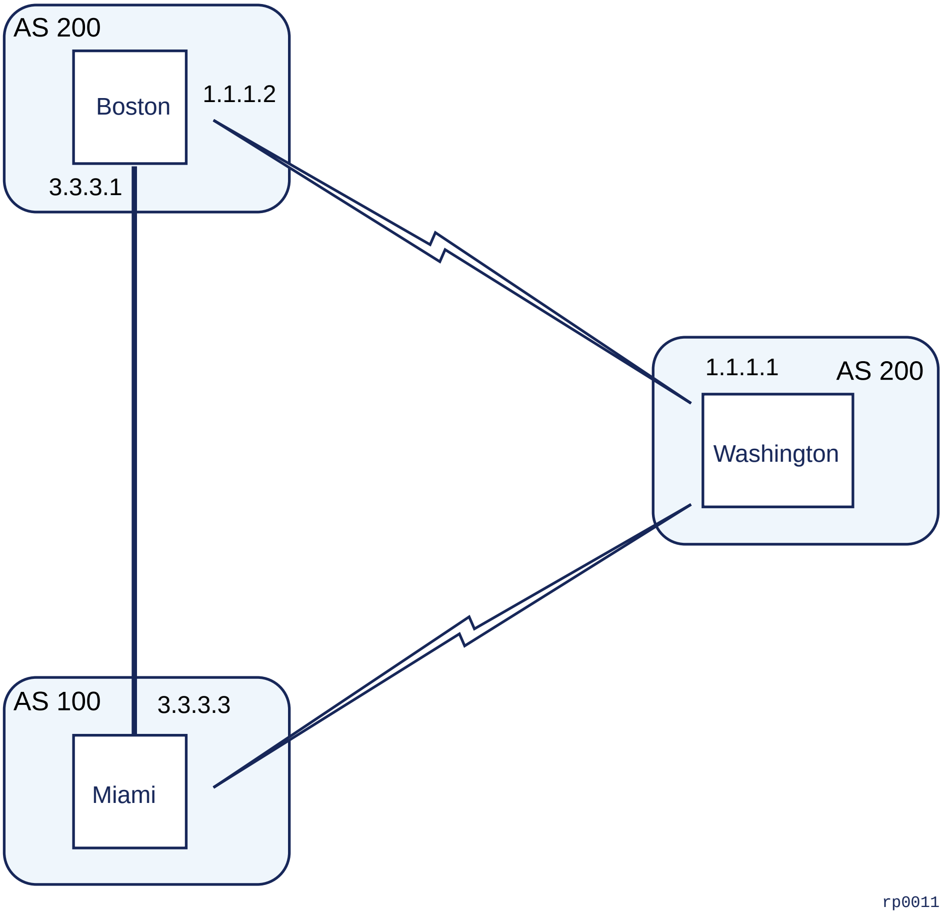


Figure 12-9 Using EBGP-Multihop

Controlling the Number of Prefixes

To control the number of prefixes received from a neighbor, use the **neighbor maximum-prefix** command in Router BGP Configuration mode, as shown below:

MOT(config-bgp)#**neighbor** {<*ip-address*> | <*name*>} **maximum-prefix** [<*num*> <*threshold*>] [**warning-only**]

where:

ip-address is the IP address of the neighbor.

name is the name of the BGP peer group.

num is the maximum number of prefixes allowed from this neighbor.

threshold is the percent of the *maximum* at which the router generates a warning message; valid entries are 1 to 100; default is 75 percent.

warning-only indicates generate a warning message only instead of shutting down the peer.

Example

The commands in the following example set the maximum prefix to 900:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 1.1.1.1 maximum-prefix 900
```

Configuring Next Hop Processing

The BGP next hop attribute is the IP address of the next hop. The next hop is usually the IP address of a neighbor you specified with the **neighbor remote-as** command. You can also use the neighbor **next-hop-self** command to specify the router itself.

To configure next hop processing, use the **neighbor remote-as** command in Router BGP Configuration mode, as shown below:

MOT(config-bgp)#**neighbor** {<*ip-address*> | <*name*>} **remote-as** <*num*>

where:

ip-address is the IP address of the neighbor.

name is the name of the BGP peer group.

num is the AS number of the neighbor AS.

In the network shown in Figure 12-10, Router San Francisco advertises network 172.56.0.0 to Router Miami with a next hop attribute of 172.56.20.1. Router Miami advertises network 120.80.0.0 to Router San Francisco with a next hop attribute of 172.56.20.2.

In BGP, the next hop of EBGP-learned routes is carried without modification in IBGP. Because of this BGP rule, Router Miami advertises 172.56.0.0 to its IBGP peer (Router New York) with a next hop attribute of 172.56.20.1. As a result, according to Router New York, the next hop to reach 172.56.0.0 is 172.56.20.1 instead of 130.60.20.1 via an IGP. Router New York drops packets destined for 171.56.0.0, if this next hop 172.56.20.1 is not reachable via IGP.

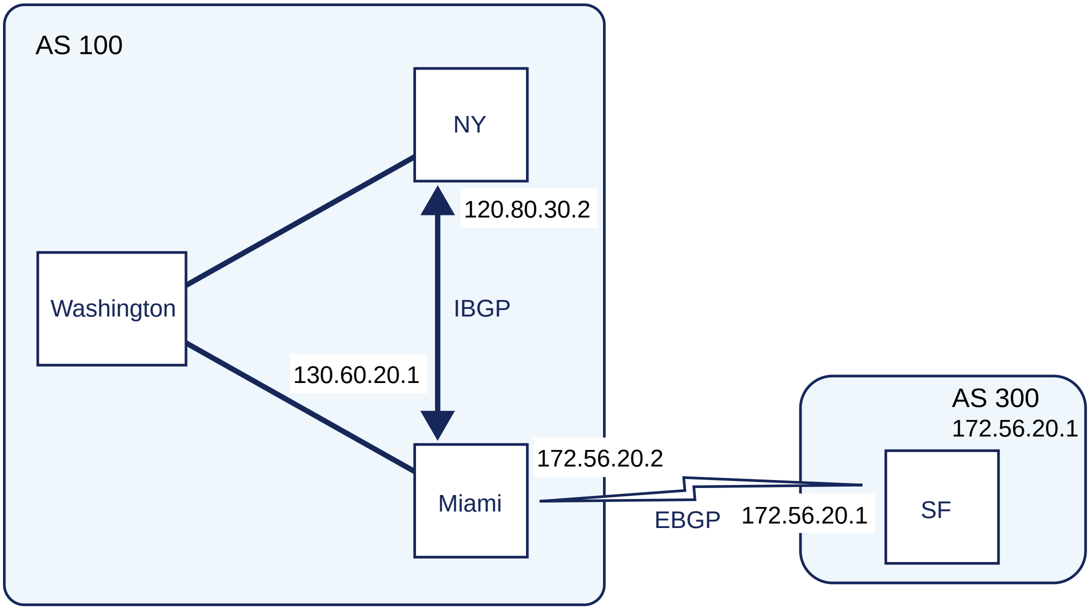


Figure 12-10 Configuring Next Hop Processing

The following commands configure Router Miami:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 172.56.20.1 remote-as 300
MOT(config-bgp)#neighbor 120.80.30.2 remote-as 100
MOT(config-bgp)#network 130.60.0.0
```

The following commands configure Router New York:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 130.60.20.1 remote-as 300
```

The following commands configure Router San Francisco:

```
MOT(config)#router bgp 300
MOT(config-bgp)#neighbor 172.56.20.2 remote-as 100
MOT(config-bgp)#network 172.56.0.0
```

Configuring Next Hop Processing

To configure next hop processing, use the **neighbor next-hop-self** command in Router BGP Configuration mode, as shown below:

MOT(config-bgp)#**neighbo**r {<*ip-address*> | <*name*>} **next-hop-self**

where:

ip-address is the IP address of the neighbor.

name is the name of the neighbor peer group.

The network shown in Figure 12-11 shows a situation that may require a different IP address for the next hop. Routers Miami, Washington, and San Francisco use Frame Relay as a common medium. Router San Francisco advertises 172.24.0.0 to Router Washington with a next hop of 172.56.10.1. Routing fails because Router Washington does not have a direct PVC connection to Router San Francisco and cannot reach the next hop. Use the **neighbor next-hop-self** command to cause Router San Francisco to advertise 172.24.0.0 with the next hop attribute set to 172.56.10.2.

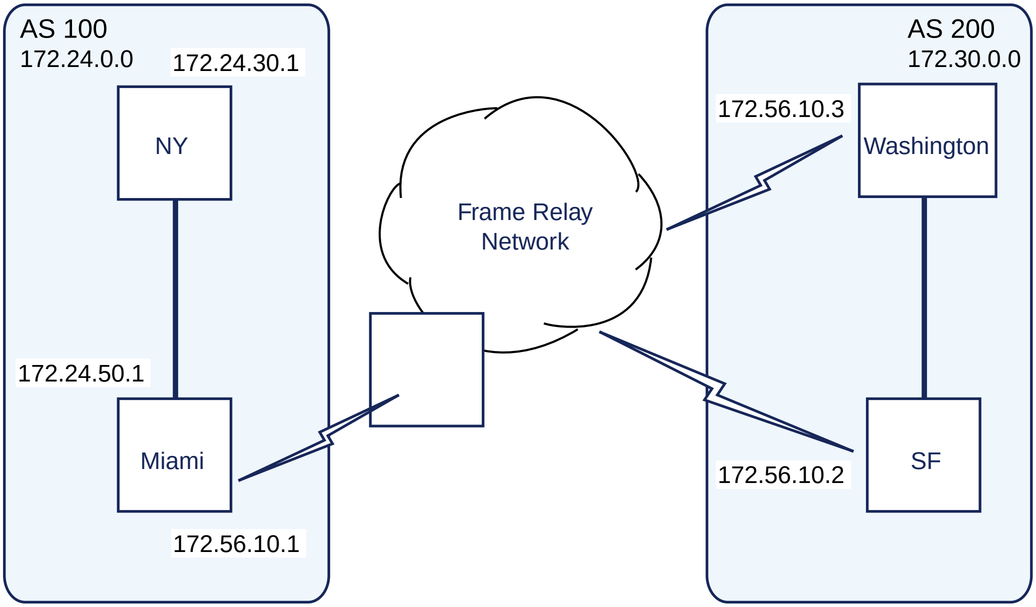


Figure 12-11 Using the neighbor next-hop-self Command

Example

The following commands configure Router San Francisco:

```
MOT(config)#router bgp 200
MOT(config-bgp)#neighbor 172.56.10.1 remote-as 200
MOT(config-bgp)#neighbor 172.56.10.1 next-hop-self
```

Configuring Global BGP Tasks

The following are global configuration tasks that apply to overall BGP routing and not to a single BGP peer:

- Resetting BGP Connections
- Configuring BGP Soft Reconfiguration
- Enabling and Disabling Synchronization

- Configuring BGP Administrative Weights
- Adjusting BGP Timers
- Setting the Administrative Distance for a Route
- Disabling Route Summarization
- Configuring Aggregate Addresses
- Assigning an Interface to BGP Session
- Configuring a Default Route
- Redistribution

Resetting BGP Connections

Once you define two routers to be BGP neighbors, they form a BGP connection and exchange routing information. If you subsequently change a BGP filter, weight, route map, distance, or timer, or make a similar configuration change, you must reset BGP connections for the configuration change to take effect.

1. To reset BGP connections, use the **clear ip bgp** command in Privileged EXEC mode, as shown below:

   `MOT>`**clear ip bgp** {***** | <*ip-address*> | <*name*>} [**soft** [**in** | **out**]]

   where:

   * resets all BGP connections.

   ip-address is the BGP neighbor address.

   name is the BGP neighbor peer group name.

   soft in initiates inbound soft reconfiguration.

   soft out initiates outbound soft reconfiguration.

2. To reset flap statistics information for a BGP neighbor, use the **clear ip bgp flap-statistics** command in Privileged EXEC mode, as shown below:

   `MOT>`**clear ip bgp** {<*ip-address*> | <*name*>} **flap-statistics**

   where:

   ip-address is the BGP neighbor address.

   name is the BGP peer group name.

Configuring BGP Soft Reconfiguration

To make a change in routing policy, you must clear a BGP session. This causes cache invalidation, which may have a large impact on the operation of your networks. *Inbound soft reconfiguration* needs inbound updates from a neighbor and enables the new inbound policy to take effect. *Outbound soft reconfiguration* sends a new set of updates to a neighbor and causes the new local outbound policy to take effect without resetting the BGP session. You can configure the BSR to store received updates, a requirement for inbound BGP soft reconfiguration. Outbound reconfiguration does not require enabling of inbound soft reconfiguration.

1. To reset a BGP neighbor IP address or peer group using a software reconfiguration of the inbound route update, use the **clear ip bgp soft-reconfiguration inbound** command in Privileged EXEC mode, as shown below:

   ```
   MOT(config-bgp)#clear ip bgp soft-reconfiguration [<ip-address> |
   <name>] inbound
   ```

   where:

   ip-address – the IP address of the neighbor.

   name – the name of the BGP peer group.

2. To reset a BGP neighbor IP address or peer group using a software reconfiguration of the outbound route update, use the **clear ip bgp soft-reconfiguration outbound** command in Privileged EXEC mode, as shown below:

   ```
   MOT(config-bgp)#clear ip bgp soft-reconfiguration [<ip-address> |
   <name>] outbound
   ```

   where:

   ip-address – the IP address of the neighbor.

   name – the name of the BGP peer group.

Enabling and Disabling Synchronization

Synchronization is a feature of BGP that prevents a BGP speaker from advertising a route before all routers within an AS have learned the route. Without synchronization, traffic may be dropped as a result of intermediate non-BGP routers not having learned routes when the AS provides transit service to other ASs.

Usually, a BGP speaker does not advertise a route to an external neighbor unless that route is local or exists in the IGP. This feature allows routers within an AS to learn the route before BGP makes it available to other ASs.The **no synchronization** command disables synchronization and allows the BGP router to advertise a network route without waiting to learn it via IGP.

Figure 12-12 shows a situation that demonstrates the value of synchronization. Router Los Angeles sends updates about network 192.56.0.0 to Router Boston. Router New York receives updates about network 192.56.0.0 from Router Boston via IBGP. Router New York wants to reach network 192.56.0.0 and sends traffic to Router Albany. If Router Boston does not distribute network 192.56.0.0. into an IGP, Router Albany cannot know that network 192.56.0.0 exists. Router Albany drops the packets from Router New York destined for network 192.56.0.0. In addition, Router New York advertises to AS 200 that it can reach 192.56.0.0 before Router Albany learns about the network via IGP. This means that traffic coming from Router Chicago to Router Albany through Router New York with a destination of 192.56.0.0 is dropped.

Synchronization solves this problem by not allowing BGP to advertise a route before all routers within the same AS have learned about the route. In Figure 12-12, Router New York waits to hear about network 192.56.0.0 before it sends an update to Router Chicago.

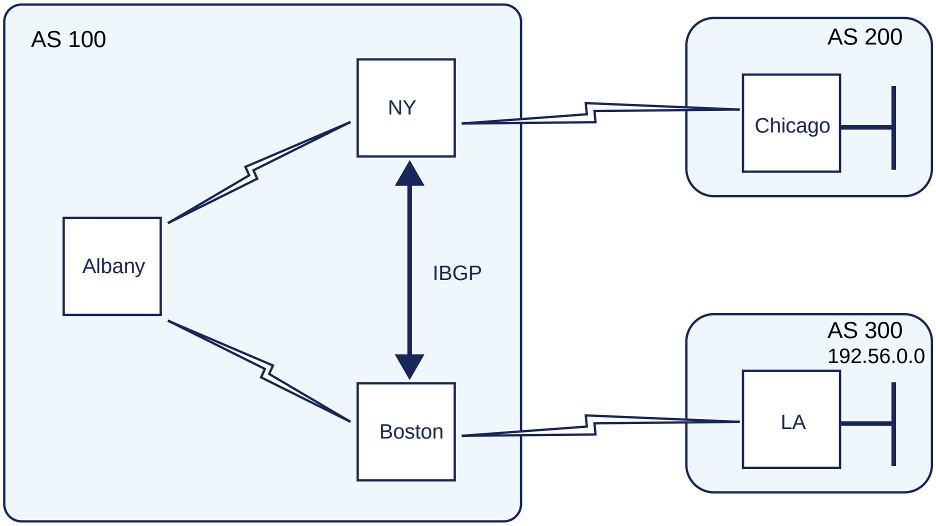


Figure 12-12 Synchronization

If your AS does not pass traffic from one AS to another or if all the transit routers in your AS run BGP, use the **no synchronization** command in Router BGP Configuration mode to disable synchronization, as shown below:

```
MOT(config-bgp)#no synchronization
```

Example

The commands in the following example configure a router with synchronization disabled:

```
MOT(config)#router bgp 100
MOT(config-bgp)#network 192.24.0.0.
MOT(config-bgp)#neighbor 2.2.2.2 remote-as 100
MOT(config-bgp)#neighbor 1.1.1.1 remote-as 300
MOT(config-bgp)#no synchronization
```

Configuring BGP Administrative Weights

You can assign a weight to a neighbor connection if more than one route exists for the same destination. A weight indicates a preference for a particular route; a higher weight indicates a preferred route. Initially, all routes learned from the neighbor have the assigned weight. The BSR chooses the route with the highest weight as the preferred route if multiple routes exist for a particular network.

In Figure 12-13, Routers Boston and New York learn about network 160.80.0.0 from AS200. Router Boston and New York propagate the update to Router Los Angeles. Router Los Angeles has two routes for reaching 160.80.0.0 and must determine the appropriate route. On Router Los Angeles, if you set the weight of updates coming from Router Boston to be higher than the updates coming from Router New York, Router Los Angeles uses Router Boston as the next hop to reach network 160.80.0.0.

Use the following commands to assign a weight to a neighbor connection:

- **neighbor weight**
- **route-map**
- **access-list**

The weights assigned with the **match as-path** and **set weight route map** commands override the weights assigned using the **neighbor weight** and **neighbor filter-list** commands.

To change the weight attribute of all route updates received from a specific AS, use the **neighbor weight** command in Router BGP Configuration mode, as shown below:

`MOT(config-bgp)#`**neighbor** {<*ip-address*> | <*name*>} **weight** <*num*>

where:

ip-address is the IP address of the neighbor.

name is the name of the BGP peer group.

num is the assigned weight in the range 0 – 65535.

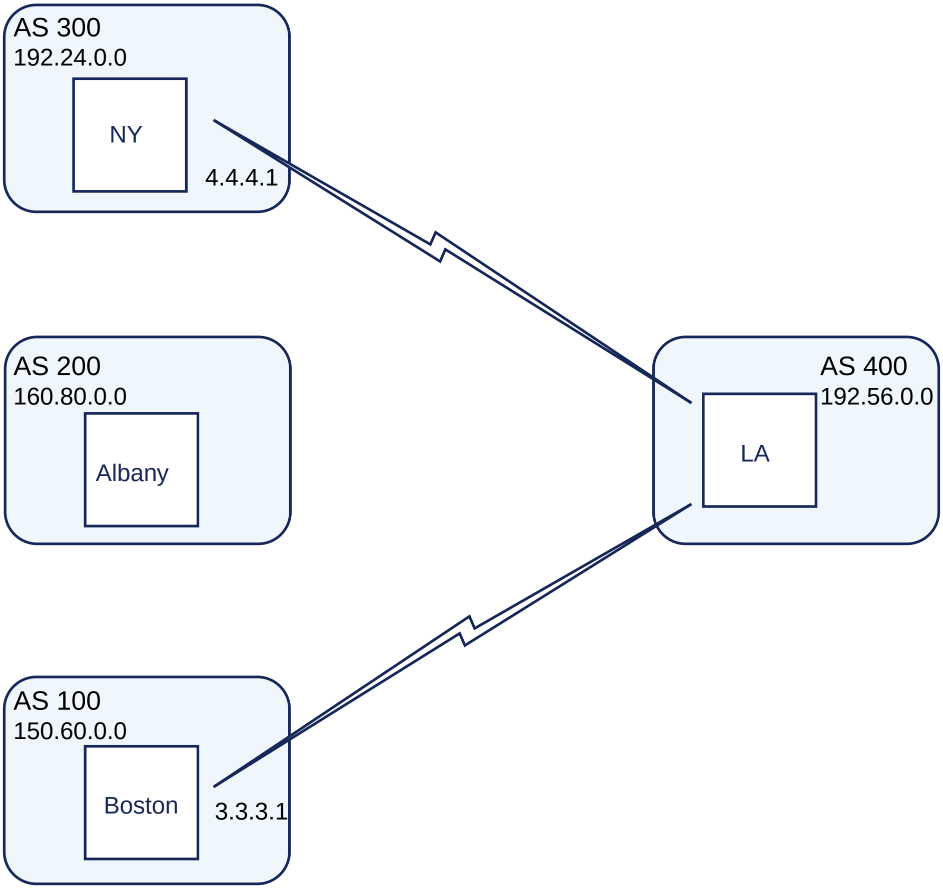


Figure 12-13 Assigning a Weight to a Neighbor Connection

Example

For example, the commands in the following example configure Router Los Angeles (as shown in Figure 12-13) using the **neighbor weight** command. This configuration assigns the weight attribute of 1000 to all route updates received from AS 100 and assigns 500 to the weight attribute of all route updates from AS 300. This causes Router Los Angeles to send traffic through Router Boston to destinations reachable via both the ASs.

```
MOT(config)#router bgp 400
MOT(config-bgp)#neighbor 3.3.3.1 remote-as 100
MOT(config-bgp)#neighbor 3.3.3.1 weight 1000
MOT(config-bgp)#neighbor 4.4.4.1 remote-as 300
MOT(config-bgp)#neighbor 4.4.4.1 weight 500
```

Using a Route Map

The commands in the following example configure Router Los Angeles using a route map. In the commands to configure Router Los Angeles, Instance 10 or route map 10 assigns a weight of 1000 to any updates from AS 100. Instance 20 assigns a weight of 500 to updates from any other AS.

Example

```
MOT(config)#router bgp 400
MOT(config-bgp)#neighbor 3.3.3.1 remote-as 100
MOT(config-bgp)#neighbor 3.3.3.1 route-map 10 in
MOT(config-bgp)#neighbor 4.4.4.1 remote-as 300
MOT(config-bgp)#neighbor 4.4.4.1 route-map 10 in
MOT(config-bgp)#exit
MOT(config)#ip as-path access-list 1 permit ^100
MOT(config)#route-map 10 permit 10
MOT(config)#match as-path 1
MOT(config)#set weight 1000
MOT(config)#route-map 10 permit 20
MOT(config)#set weight 500
```

Using an AS Path Access List

The commands in the following example configure Router Los Angeles using an AS path access list. Filter List 1 assigns a weight attribute of 1000 to updates received from neighbors from AS 100. Access List 1 permits any update whose AS-path attribute begins with 100 (specified by “^”). The same is true for Access List 2 regarding AS 300. Filter List 2 assigns a weight attribute of 500 to updates received from AS 300.

Example

```
MOT(config)#router bgp 400
MOT(config-bgp)#neighbor 3.3.3.1 remote-as 100
MOT(config-bgp)#neighbor 3.3.3.1 filter-list 1 weight 1000
MOT(config-bgp)#neighbor 4.4.4.1 remote-as 300
MOT(config-bgp)#neighbor 4.4.4.1 filter-list 2 weight 500
MOT(config-bgp)#exit
MOT(config)#ip as-path access-list 1 permit ^100
MOT(config)#ip as-path access-list 2 permit ^300
```

Adjusting BGP Timers

BGP supports the following two commands that set the frequency of keepalive and holdtime timers:

- **timers bgp** — globally sets the keepalive timers for BGP
- **neighbor timers** — sets the keepalive timers for a BGP peer or peer group

Keepalive messages are exchanged between BGP peers or peer groups to monitor the health of the link between them. If a BGP peer does not receive a keepalive message, it waits for a configured holdtime before the ailing BGP peer is declared dead by its waiting peer.

To configure the keepalive frequency and holdtime interval globally for BGP, use the **timers bgp** command in Router BGP Configuration mode, as shown below:

`MOT(config-bgp)#`**timers bgp** <*keepalive*> <*holdtime*>

where:

> *keepalive* – is the frequency in seconds, that the BSR sends keepalive messages to its peers; default is 60 seconds.

holdtime – is the interval in seconds, after which, not receiving a keepalive or any other BGP message, the BSR declares a BGP peer dead; default is 180 seconds.

To configure the keepalive frequency and holdtime interval for a BGP peer or peer-group, use the **neighbor timers** command in Router BGP Configuration mode, as shown below:

`MOT(config-bgp)#`**neighbor** [<*ip-address*> | <*name*>] **timers** <*keepalive*> <*holdtime*>

where:

ip-address is the IP address of the BGP peer.

name is the name of the BGP peer group.

keepalive is the frequency, in seconds, that the BSR sends keepalive messages to its peers. The default is 60 seconds.

holdtime is the interval a BSR waits to receive a keepalive message before it declares a BGP peer dead. The default is 180 seconds.

Example

The commands in the following example, configure the keepalive frequency and holdtime interval for BGP on a BSR and configure the keepalive frequency and holdtime interval for a BGP peer:

```
MOT(config)#router bgp 100
MOT(config-bgp)#timers bgp 80 200
MOT(config-bgp)#neighbor 192.56.20.2 timers 80 200
```

Setting the Administrative Distance for a Route

An administrative distance is a rating of the trustworthiness of a routing information source, such as a router or group of routers. The administrative distance is an integer between 0 and 255 with a higher value indicating a lower trust rating. An administrative distance of 255 denotes a routing information source that cannot be trusted and should be ignored. Routes with distances of 255 are not installed in the routing table.

You can change an administrative distance if you know that another protocol provides a better route than that learned via EBGP or if you want IBGP to show preference for internal routes.

Note: Changing the administrative distance of BGP internal routes is dangerous and is not recommended. It can cause the accumulation of routing table inconsistencies that can break routing within an AS and between ASs.

To set the following three administrative distance types, use the **distance bgp** command:

- external — for BGP external routes learned from a neighbor external to the AS.
- internal — for BGP internal routes learned from another BGP router within the same AS.
- local — for local BGP routes that are networks listed with the **network** command, often as back doors for that router or for networks that are redistributed from another process.

To set the external, internal, and local administrative distances for a BGP router, use the **distance bgp** command in Router BGP Configuration mode, as shown below:

```
MOT(config-bgp)#distance bgp [<external-distance> <internal-distance>
<local-distance>]
```

where:

external-distance is the administrative distance for routes external to the AS; range is 1 to 255; default is 20.

internal-distance is the distance for routes internal to the AS; range is 1 to 255; default is 200.

local-distance is the administrative distance for local routes; range is 1 to 255; default is 200.

Example

In the following example, internal routes are preferable to those learned through the IGP. The administrative distance values are set accordingly: 20 for routes external and internal to AS 100 and 200 for local routes.

```
MOT(config)#router bgp 100
MOT(config-bgp)#network 160.20.0.0
MOT(config-bgp)#neighbor 156.30.10.1 remote-as 100
MOT(config-bgp)#neighbor 131.65.1.2 remote-as 200
MOT(config-bgp)#distance bgp 20 20 200
```

Disabling Route Summarization

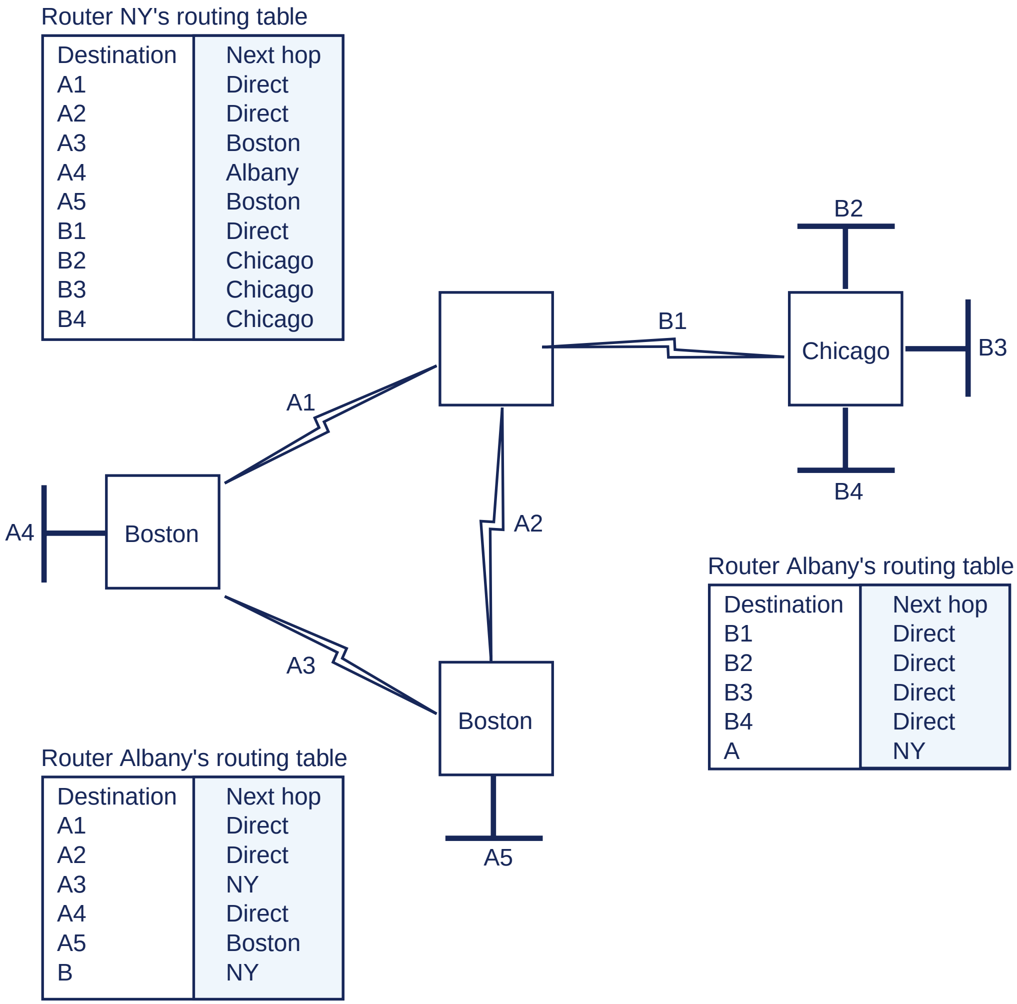
Route summarization condenses routing information. Without summarization, each router in a network must retain a route to every subnet in the network. With summarization, routers can reduce some sets of routes to a single advertisement, reducing both the load on the router and the perceived complexity of the network. The importance of route summarization increases with network size.

The reduction in route propagation and routing information overhead is significant. For example, without summarization, each router in a network with 1,000 subnets must contain 1,000 routes. With summarization in a Class B network with eight bits of subnet address space, each router must know all of the routes for each subnet in its network number. This is 250 routes, assuming that 1,000 subnets fall into four major networks of 250 routes each. In addition, the router must know one route for each of the other three networks for a total of 253 routes. This represents a nearly 75 percent reduction in the size of the routing table.

In Figure 12-14, Router Albany maintains one route for all destination networks beginning with B, and Router Chicago maintains one route for all destination networks beginning with A. This is the essence of route summarization. Router New York tracks all routes because it exists on the boundary between A and B.

To disable automatic network summarization of routes, use the **no auto-summary** command in Router BGP Configuration mode, as shown below:

```
MOT(config-bgp)#no auto-summary
```



Router NY's routing table

Destination	Next hop
A1	Direct
A2	Direct
A3	Boston
A4	Albany
A5	Boston
B1	Direct
B2	Chicago
B3	Chicago
B4	Chicago

Router Albany's routing table

Destination	Next hop
B1	Direct
B2	Direct
B3	Direct
B4	Direct
A	NY

Router Albany's routing table

Destination	Next hop
A1	Direct
A2	Direct
A3	NY
A4	Direct
A5	Boston
B	NY

Figure 12-14 Route Summarization

Example

The commands in the following example disable automatic network summarization for AS 100:

```
MOT(config)#router bgp 100
MOT(config-bgp)#no auto-summary
```

Configuring Aggregate Addresses

Using CIDR addressing, you can combine routes so that multiple routes are advertised as a single route. CIDR replaces the concept of classes (such as Class A, Class B, and Class C) with the concept of IP prefixes. An IP prefix is a network address that indicates the number of bits that comprise the network number.

1. To combine multiple routes, use the **aggregate** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**aggregate** <*ip-address*> <*address-mask*>

   where:

   ip-address is the aggregate IP address.

   address-mask is the aggregate IP mask.

2. To identify the route map for selecting routes to be aggregated, use the **aggregate advertise-map** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**aggregate** <*ip-address*> <*address-mask*> **advertise-map** <*name*>

   where:

   ip-address is the aggregate IP address.

   address-mask is the aggregate IP mask.

   name is the route map name.

3. To generate autonomous system path information for the aggregate IP address based on the routes selected for aggregaton, use the **aggregate as-set** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**aggregate** <*ip-address*> <*address-mask*> **as-set**

   where:

   ip-address is the aggregate IP address.

   address-mask is the aggregate IP mask.

4. To identify the route map for manipulating the attributes of the aggregate route, use the **aggregate attribute-map** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**aggregate** <*ip-address*> <*address-mask*> **attribute-map** <*name*>

   where:

   ip-address is the aggregate IP address.

   address-mask is the aggregate IP mask.

   name is the route map name.

5. To filter all more specific routes from updates, use the **aggregate summary-only** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**aggregate** <*ip-address*> <*address-mask*> **summary-only**

   where:

   ip-address is the aggregate IP address.

   address-mask is the aggregate IP mask.

6. To specify the route map for selectively blocking routes, use the **aggregate suppress-map** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**aggregate** <*ip-address*> <*address-mask*> **suppress-map** <*name*>

   where:

   ip-address is the aggregate IP address.

   address-mask is the aggregate IP mask.

   name is the route map name.

Example

Network 200.10.0.0 is an illegal Class C network address. This address becomes legal when it is represented in CIDR notation as 200.10.0.0/16. The /16 specifies that the subnet mask consists of 16 bits (counting from left to right). Thus, the CIDR address, 200.10.0.0/16, is the same as 200.10.0.0 with a network mask of 255.255.0.0.

```
MOT(config)#router bgp 100
MOT(config-bgp)#aggregate-address 200.10.0.0 255.255.0.0
```

Assigning an Interface to BGP Session

To allow an BGP session to use any operational interface for TCP connections, use the **neighbor update-source** command in Router BGP Configuration mode, as shown below. You specify an IP address or peer-group and the interface. This feature is often used in conjunction with loopback interfaces. Loopback interfaces are often used by IBGP peers. The advantage of using loopback interfaces is that they eliminate operational status and negotiated address dependencies that result from using the IP address of a physical interface on the router to configure BGP. Loopback interfaces are rarely used between EBGP peers because they are usually directly connected and depend on a specific interface for connectivity.

1. To assign an Ethernet interface to the BGP session, use the **neighbor update-source** command in Router BGP Configuration mode, as shown below:

   MOT(config-bgp)#**neighbor** {<*ip-address*> | <*name*>} **update-source** *ethernet* <*slot*> {/} <*port*>

   where:

   ip-address is the IP address of the BGP neighbor.

   name is the name of the BGP peer group.

   slot is the interface slot number.

   port is the interface port number.

2. To assign a loopback interface to the BGP session, use the **neighbor update-source** command in Router BGP Configuration mode, as shown below:

   MOT(config-bgp)#**neighbor** {<*ip-address*> | <*name*>} **update-source** *loopback* <*num*>

   where:

ip-address is the IP address of the BGP neighbor.

name is the name of the BGP peer group.

loopback indicates the interface type.

num is the loopback number; valid values are 1 to 16.

Note: The loopback interface is unrelated to the IP loopback address 127.x.x.x.

Example

Figure 12-15 shows a network that can benefit from the use of a loopback interface, because an alternate path exists. Routers New York and Albany are running IBGP within AS 100. Router New York specifies the IP address of the loopback interface (140.10.0.1) of Router Albany in the **neighbor remote-as** command. Router Albany is configured to include the **neighbor update-source** command so that the source of BGP TCP connections for the specified neighbor is the IP address of the loopback interface instead on the IP address of a physical interface. This eliminates the dependency on the physical interface. If a cable break occurs on S0, the connection can still be maintained via S1, provided IP connectivity still exists to the loopback.

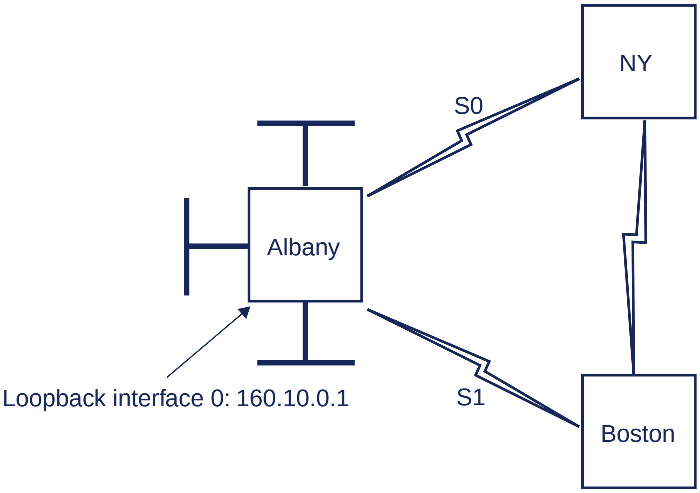


Figure 12-15 Using a Loopback Interface

The following commands configure a loopback interface for Router Albany, enter the following:

```
MOT(config)#interface loopback 1
MOT(config-if)#ip address 140.10.0.1 255.255.255.255
MOT(config-if)#exit
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 192.56.42.10 update-source loopback 1
```

Configuring a Default Route

A default route in a router IP forwarding table is used by the router if a routing entry for a destination does not exist. By convention, a default route is represented by the network mask combination 0.0.0.0/0.0.0.0. Any AS advertising the default route represents itself as the *gateway of last resort* to other systems.

To target the default route to a specific BGP neighbor so that only that router receives the default advertisement, use the **neighbor default-originate route-map** command in Router BGP Configuration mode, as shown below:

MOT(config-bgp)#**neighbor** {<*ip-address*> | <*name*>} **default-originate route-map** [<*map-name*>]

where:

ip-address is the IP address of the neighbor.

name is the name of the BGP peer group.

map-name is name of the route map. The route map allows route 0.0.0.0/0.0.0.0 to be injected conditionally.

It is important to control defaults in BGP, because a BGP neighbor, in an attempt to advertise a default route to a specific peer, may send the default to all of its neighbors.

Example

In Figure 12-16, Router Boston originates the default route 0.0.0.0/0.0.0.0 toward Router Miami only. Router Chicago does not receive the default route.

The following commands configure Router Boston:

```
MOT(config)#router bgp 100
MOT(config-bgp)#network 150.20.30.0 255.255.255.0
MOT(config-bgp)#neighbor 150.20.20.1 remote-as 200
MOT(config-bgp)#neighbor 150.20.20.1 default-originate
```

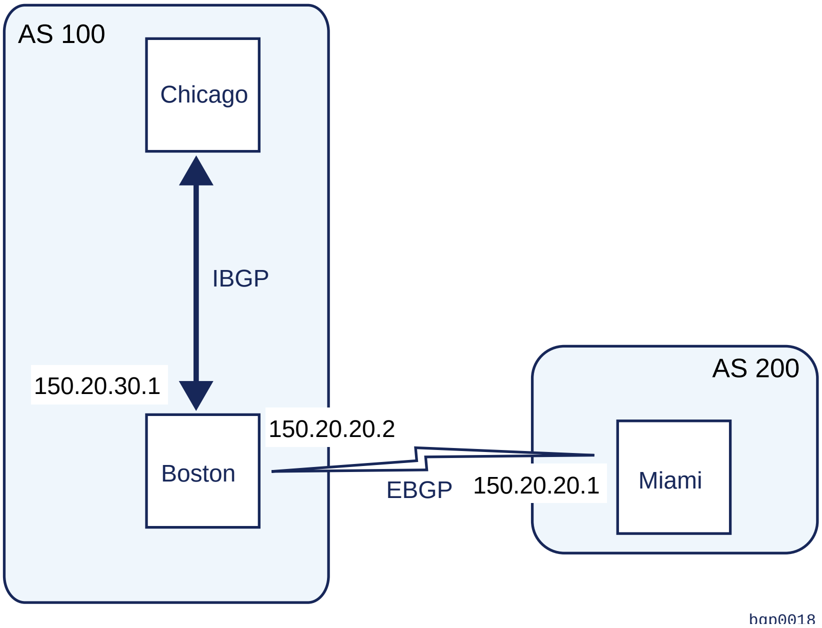


Figure 12-16 Dynamically Configuring a Default Route

Configuring BGP Update Flows

BGP update messages are exchanged between BGP peers to determine how a BGP router updates route entries in its routing table. Use BGP commands to modify the information in BGP updates sent out by a router to one or more of its peers.

To control the flow of BGP updates, configure the following:

- Filtering of sent and received updates
- BGP path selection options

Configuring BGP Path Selection Algorithm

BGP selects the best possible path for a route and installs it in its route table. If only one route exists for a specific destination, BGP selects that route, because, by definition, it is the best route. If multiple routes exist, BGP uses the BGP Path Selection Algorithm to select the best path.

BGP Path Selection Algorithm

The BGP path selection process uses the following sequential criteria to select a path:

1. If the next hop is inaccessible, BGP does not consider the route. For this reason, it is important to have an IGP route to the next hop.

2. If synchronization is enabled, the path is internal, and the route is not in an IGP. BGP ignores the route.

3. BGP uses the route that was locally originated, using either the **network** or **aggregate-address** command, or through redistribution from an IGP.

4. BGP uses the path with the largest weight, a value ranging from 0 to 65535. The administrative *weight* is local to the router. BSR originating paths have a of weight 32768 by default; other paths, from peers, have a default weight of 0. To configure specific neighbors as *preferred* for most traffic, use the **neighbor weight** command to assign a higher weight to all routes learned from that neighbor. You can also assign weights based on AS path access lists. A given weight becomes the weight of the route, if the AS path is permitted by the access list. Any number of weight filters are allowed.

5. BGP uses the route with the largest local preference. Define a particular path as more preferable or less preferable than other paths by changing the default local preference value of 100. To assign a different default local preference value, use the **bgp default local-preference** command.

6. BGP uses the route with the shortest AS path.

7. BGP uses the route with the lowest origin type. IGP is lower than EGP, and EGP is lower than INCOMPLETE.

8. BGP uses the route with the lowest MED. The comparison is made only if the neighboring AS is the same, except when the **bgp always-compare-med** command is enabled.

9. BGP uses the route with the lowest IGP metric to the BGP nexthop.
10. BGP prefers EBGP over IBGP. All confederation paths are considered IBGP.
11. If the best route and a new route are both external and **maximum-paths *n*** is enabled, BGP inserts the new route into the IP routing table as an alternate path. EBGP multipath load sharing can occur at this point. The forwarding table holds 1 - 2 paths.
12. BGP prefers the path with the lowest IP address specified by the BGP router ID.

Configuring the Local Preference

When multiple paths exist to the same destination, the local preference specifies the preferred path. The preferred path is the one with the higher preference value. To configure the local preference of a BGP path, perform one of the following steps.

1. To set the default local preference attribute in BGP updates, use the **bgp default local-preference** command in Router BGP Configuration mode, as shown below:

   `MOT(config-bgp)#`**bgp default local-preference** <*preference-value*>

   where:

   preference-value – the local preference number in the range 0 – 4294967295

2. Use a route map to set the local preference attribute

Example: Setting the Local Preference

This example configures Routers New York and Boston so that AS 100 receives updates for network 156.10.0.0 from AS 200 and AS 400 (as shown in Figure 12-17).

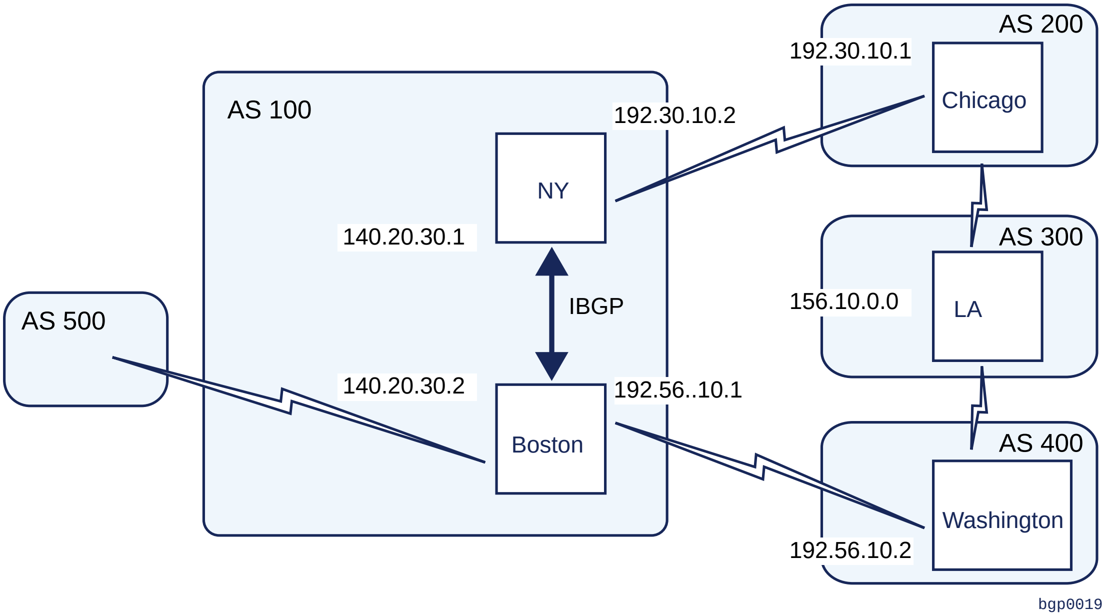


Figure 12-17 Configuring the Local Preference Attribute

Router New York sets the local preference for all updates from AS 200 to 125. Router Boston sets the local preference for all updates from AS 400 to 200. Because Router New York and Router Boston exchange local preference information within AS 100, they recognize that updates regarding network 156.10.0.0 have higher local preference when they come to AS 100 from AS 400 than from AS 200.

The following commands configure Router New York:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 140.20.30.2 remote-as 100
MOT(config-bgp)#neighbor 192.30.10.1 remote-as 200
MOT(config-bgp)#bgp default local-preference 125
```

The following commands configure Router Boston:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 140.20.30.1 remote-as 100
MOT(config-bgp)#neighbor 192.56.10.2 remote-as 400
MOT(config-bgp)#bgp default local-preference 200
```

Example: Using a Route Map to Set the Local Preference

A route map setting the local preference allows more flexibility in determining updates from a specific AS. In the previous example, all updates received by Router Boston are set to a local preference of 200 (including updates from AS 500).

Use a route map to specifically assign a local preference for updates from AS 400. In this example, all local preference attributes from updates coming from AS 400 are set to 200.

The following commands configure Router Boston:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 192.56.10.2 remote-as 400
MOT(config-bgp)#neighbor 192.56.10.2 route-map 10 in
MOT(config-bgp)#neighbor 140.20.30.1 remote-as 100
```

The following commands specify that the local preference attribute for updates coming from AS 400 are set to 200:

```
MOT(config-bgp)#neighbor route-map 10 permit 10
MOT(config-bgp)#set local-preference 200
```

Configuring the Origin Attribute

The Origin attribute indicates the route origin and is one of the following values:

- IGP – indicates that the route was learned via an IGP and, therefore, is interior to the originating AS.
- EGP – indicates that the route was learned via EGP.
- Incomplete – indicates that the origin of the route is unknown. It was learned from something other than IGP or EGP. Incomplete origin occurs when a route is distributed into BGP. This value most often appears for static routes.

The BSR assigns origin as described in Table 12-1.

Table 12-1 BSR Origin Assignment

BGP Route Entry Type	Origin Code
Redistributed	INCOMPLETE
Network	IGP
Peer-based default (0/0)	IGP

Example

In Figure 12-18, from Router Boston, the route for reaching 192.56.0.0 has an AS-path of 300 with an origin attribute of IGP. From Router Boston, the route for reaching 175.40.30.0 has an origin attribute of IGP. From Router Los Angeles, the route for reaching 150.20.0.0 has an AS-path of 100 with an origin attribute of IGP. For Router Los Angeles, the route for reaching 175.40.0.0 has an AS-path of 100 with an origin attribute of Incomplete. Route 175.40.0.0 is a redistributed route.

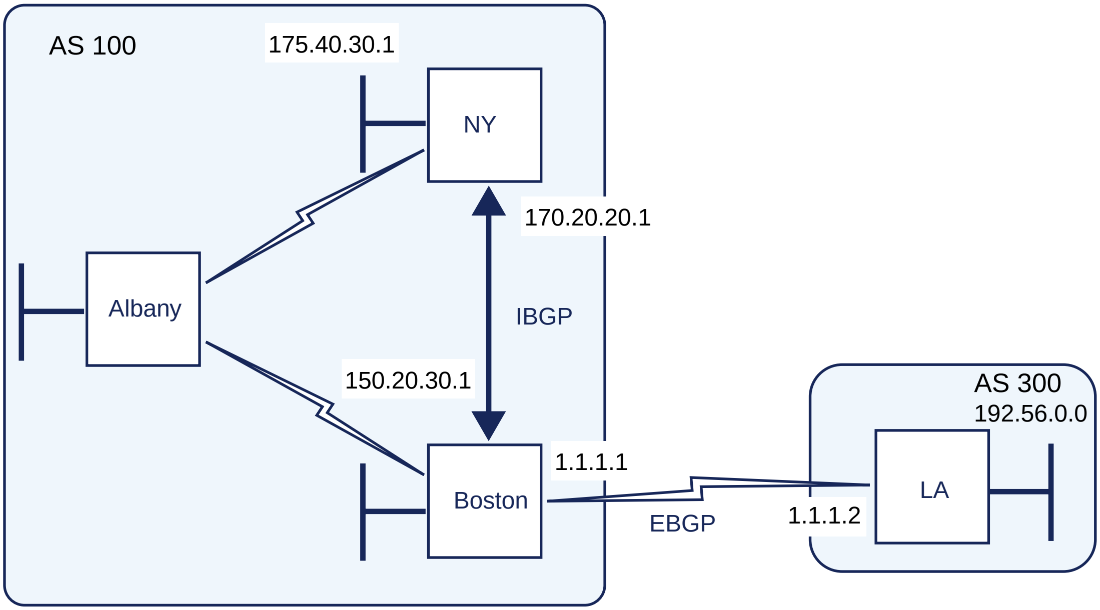


Figure 12-18 Configuring the Origin Attributed

The following commands configure Router Boston:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 170.20.20.1 remote-as 100
MOT(config-bgp)#neighbor 1.1.1.2 remote-as 300
MOT(config-bgp)#network 150.20.0.0
MOT(config-bgp)#redistribute static
MOT(config-bgp)#exit
MOT(config)#ip route 175.40.0.0 255.255.0.0
```

The following command configure Router New York:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 150.20.30.1 remote-as 100
MOT(config-bgp)#network 175.40.30.0
```

The following commands configure Router Los Angeles:

```
MOT(config)#router bgp 300
MOT(config-bgp)#neighbor 1.1.1.1 remote-as 100
MOT(config-bgp)#network 192.56.0.0
```

Configuring the AS-path Attribute

When a BGP route passes through an AS, BGP prepends its AS number to the route. The AS_path attribute contains the list of ASs that a route has gone through.

In Figure 12-19, Router Boston advertises network 150.60.0.0 in AS 400 with an AS-path of 100. When the BGP route arrives in AS 400, Router Los Angeles adds its AS number. When the BGP route arrives at Router New York, its AS_path attribute contains AS numbers 400 and 100.

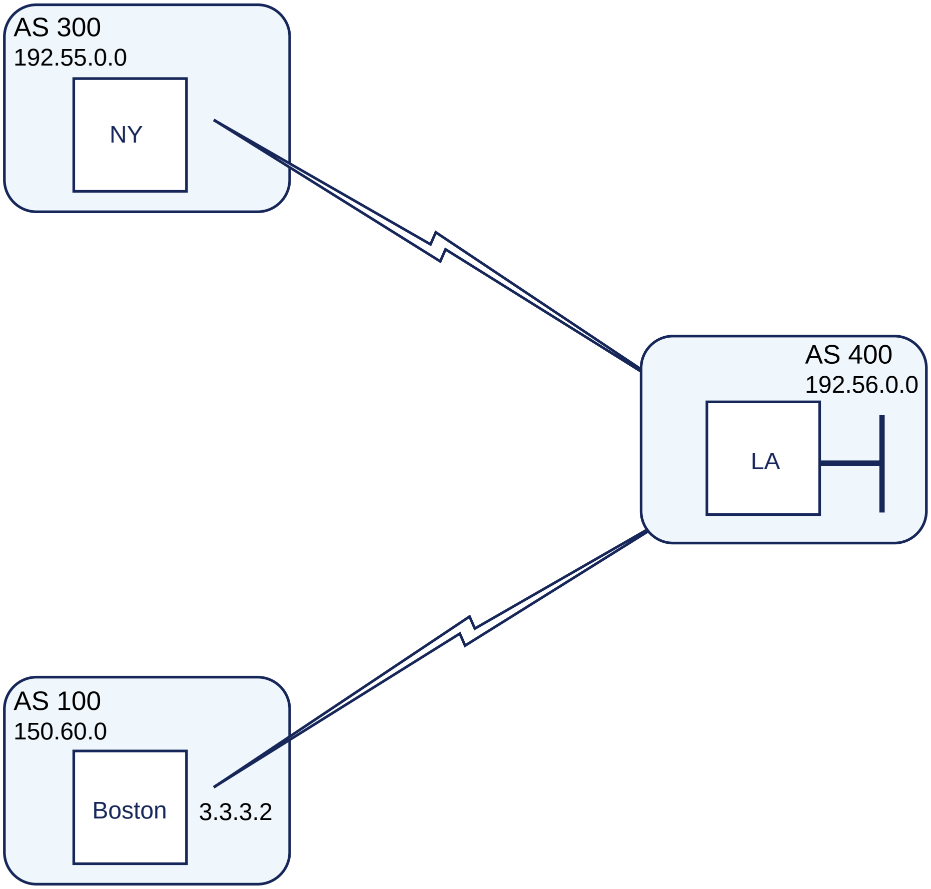


Figure 12-19 The AS-path Attribute

Configuring the MED Attribute

The MED attribute carries a metric expressing a degree of preference for a particular route. By default, the BSR compares MED attributes for paths from external neighbors only that are in the same AS.

If two ASs are connected in more than one place, you can change this value so that a router chooses the optimal link to reach a specific prefix in or behind that AS. Unlike the Local Preference attribute, the MED attribute is exchanged between ASs. When BGP sends an update to another AS, the MED attribute is reset to 0.

To compare MED attributes from different neighbors in different ASs, use the **bgp always-compare-med** command.

Example

In Figure 12-20, Routers Los Angeles, New York, and Boston send updates regarding network 155.30.0 to AS 200. Router Chicago only compares the MED attributes of routes coming from Routers New York and Boston, because, by default, BGP only compares MED attributes of routes coming from external neighbors that are in the same AS. Router Chicago ignores the MED attribute coming from Router Los Angeles even though it is smaller.

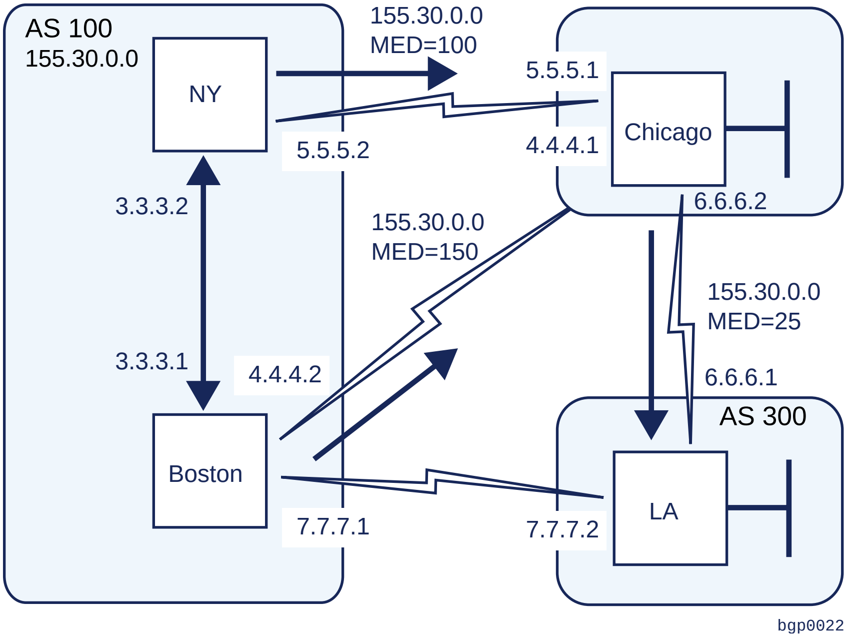


Figure 12-20 Configuring the MED

The following commands configure Router Chicago:

```
MOT(config)#router bgp 200
MOT(config-bgp)#neighbor 5.5.5.2 remote-as 100
MOT(config-bgp)#neighbor 4.4.4.2 remote-as 100
MOT(config-bgp)#neighbor 6.6.6.1 remote-as 300
```

The following commands configure Router Los Angeles:

```
MOT(config)#router bgp 300
MOT(config-bgp)#neighbor 6.6.6.2 remote-as 200
MOT(config-bgp)#neighbor 6.6.6.2 route-map 10 out
MOT(config-bgp)#neighbor 7.7.7.1 remote-as 100
MOT(config-bgp)#exit
MOT(config)#route-map 10 permit 10
MOT(config)#set metric 25
```

The following commands configure Router New York:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 5.5.5.1 remote-as 200
MOT(config-bgp)#neighbor 5.5.5.1 route-map 10 out
MOT(config-bgp)#neighbor 3.3.3.1 remote-as 100
MOT(config-bgp)#exit
MOT(config)#route-map 10 permit 10
MOT(config)#set metric 100
```

The following commands configure Router Boston:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 4.4.4.1 remote-as 200
MOT(config-bgp)#neighbor 4.4.4.1 route-map 10 out
MOT(config-bgp)#neighbor 3.3.3.2 remote-as 100
MOT(config-bgp)#neighbor 7.7.7.2 remote-as 300
MOT(config-bgp)#exit
MOT(config)#route-map 10 permit 10
MOT(config)#set metric 150
```

In the following commands, the **bgp always-compare-med** command changes the configuration of Router Chicago so that it considers the MED attribute from Router Los Angeles:

```
MOT(config)#router bgp 200
MOT(config-bgp)#neighbor 5.5.5.2 remote-as 100
MOT(config-bgp)#neighbor 4.4.4.2 remote-as 100
MOT(config-bgp)#neighbor 6.6.6.1 remote-as 300
MOT(config-bgp)#bgp always-compare-med
```

Configuring the Community Attribute

A community is a group of destinations that share a common policy. You can define the communities a destination belongs to. This determines how routes are advertised. Use a route map to set the community attribute. BGP defines the following well-known communities:

- *no-export* advertises a route to IBGP peers only (peers within the local AS).
- *no advertise* does not advertise a route to any peer.
- *local* advertises a route only to peers in the same subconfederation.

Examples

The following example sets the value of the community attribute:

```
MOT(config)#route map 20 permit 10
MOT(config-bgp)#match ip address 2
MOT(config-bgp)#set community no-advertise
MOT(config-bgp)#exit
MOT(config)#route-map 20 permit 20
MOT(config-bgp)#match as-path 2
MOT(config-bgp)#set community 300 additive
MOT(config-bgp)#exit
```

The following example uses the **send community** command to send the community attribute to a neighbor:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 5.5.5.5 remote-as 400
MOT(config-bgp)#neighbor 5.5.5.5 send-community
MOT(config-bgp)#neighbor 5.5.5.5 route-map 20 out
MOT(config-bgp)#exit
```

Configuring Routing Policy

Route maps define the conditions for redistributing routes from one routing protocol to another (for example, between BGP and OSPF) and for advertising and learning routes from one router to another. A route map consists of a set of **route-map** commands, **match** statements defining conditions that a route must meet and **set** statements defining the conditions that apply to a route.

To define a route map, use the **route-map** command in Global Configuration mode, as shown below:

MOT(config)#**route-map** <*name*> [**permit** | **deny**] <*sequence-number*>

where:

name uniquely identifies a route map.

permit specifies consider the route for further operation.

deny specifies do not consider the route for further operation.

sequence-number uniquely identifies an instance of the route map. Instances with lower sequence numbers are parsed first.

Match and Set Statements

Match statements define the conditions that must be met by a route. Each instance may contain multiple match statements.If all match statements within a given instance match for a given route, the route meets the conditions of the instance. Therefore, the ordering of match statements within an instance does not matter. If an instance has no match statements, all routes meet the conditions of the instance (unless they are denied by an instance with a lower sequence-number).

Set statements define the conditions that are applied to the route. If the match conditions of a given instance are met by a route, all set statements within the instance are applied to the route. Therefore, the ordering of set statements within an instance does not matter since either all or none are applied.

If an instance has no set statements and all the match statements in the instance match, nothing is set for the route. The route is simply redistributed, advertised, or learned as is (depending on where the route map is applied).

Table 12-1 shows **match** commands for creating route maps.

Table 12-1 match Commands

Command	Description
match as-path	Matches a BGP AS_path access list.
match community	Matches a BGP community list.

Table 12-1 match Commands

Command	Description
match ip address	Matches an IP access list.
match ip next-hop	Matches the next-hop ip address.
match metric	Matches a routing metric value. For BGP, this is the MED.
match ip route-src	Matches neighbor IP address

Table 12-2 shows **set** commands for creating route maps.

Table 12-2 set Commands

Command	Description
set as-path prepend	Modifies an AS path.
set comm_list	Removes selected communities.
set community	Sets the BGP community attribute.
set ip next-hop	Sets the next-hop attribute of a route.
set local-preference	Set the local preference value.
set metric	Set the metric. For BGP, this is the MED.
set origin	Set the BGP origin.
set weight	Set weight of the route.

Example

The commands in the following example, executed from Global Configuration mode, create the route map, locpref, the AS path access list 1, and apply the route map to a BGP neighbor. They create the route map, locpref, which sets the local preference for BGP updates. The route map also uses an AS path access list to permit any update whose AS path attribute begins and ends with 400. This sets the local preference to 50 for all updates originating from AS 400.

```
MOT(config)#route-map locpref permit 10
MOT(config)#match as-path 1
MOT(config)#set local-preference 50
MOT(config)#route-map locpref permit 20
MOT(config)#exit
```

The following commands, executed from Global Configuration mode, create AS_path access list 1:

```
MOT(config-bgp)#ip as-path access-list 1 permit ^400
```

These commands, also executed from Global Configuration mode, apply the route map to a BGP neighbor:

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 160.20.30.4 route-map locpref in
MOT(config-bgp)#exit
```

Handling Access Lists

An access list is a sequential collection of permit and deny conditions. The BSR tests conditions one-by-one against conditions in an access list. The BSR supports the following two types of access-lists:

- IP access list
- AS path access list

Configuring an Access List

To define an extended IP access list entry, use the **access-list** command in Global Configuration mode, as shown below:

`MOT(config)#`**access-list** <*access-list-number*> {**permit** | **deny**} **ip** {<*source-address*> <*source-address-wildcard*> | **any**} {<*destination-address*> <*destination-address-wildcard*> | **any**}

where:

access-list-number is the number of the access list.

source-address is the source IP address.

source-address-mask is the network wildcard bits of the source address.

destination-address is the destination IP address.

destination-address-mask is the network wildcard bits of the destination address.

any is the abbreviation for 0.0.0.0 address and 255.255.255.255 wildcard to match against any IP address.

Permit and deny conditions in an IP access list apply to IP addresses. Use the **neighbor distribute-list** command to apply an access list to a BGP neighbor.

Example

This sample configuration filters BGP updates from a BGP neighbor. The following commands, executed in Global Configuration mode, configure standard Access List 4 by specifying its permit and deny conditions. Access list 4 prohibits the propagation of networks specified in the deny statements (10.0.0.0, 162.15.0.0, and 180.10.0.0) and permits all others.

```
MOT(config)#access-list 4 deny 10.0.0.0 0.255.255.255
MOT(config)#access-list 4 deny 162.15.0.0 0.0.255.255
MOT(config)#access-list 4 deny 180.10.0.0 0.0.255.255
MOT(config)#access-list 4 permit any
```

Note: Any type of list always has an assumed *deny all* entry as the last statement. If there are no matches at the end, the route or match (depending on the type of list and/or how it is used) is denied.

The following commands, also executed in Global Configuration mode, enable BGP, specify an AS, and apply Access List 4 to a neighbor. The example instructs the router to pass all network information received from the BGP neighbor 156.30.10.22 through access list 4.

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 156.30.10.22 distribute-list 4 in
```

Configuring an AS Path Access List

The permit and deny conditions in an AS path access list apply to AS paths. The **neighbor filter-list** command applies an AS path access list for inbound and outbound updates to a BGP neighbor. The **match as_path** command adds a match clause to a route map. To define an AS path access list, use the **ip as-path access-list** command in Router BGP Configuration mode, as shown below:

MOT(config-bgp)#**ip as-path access-list** <*access-list-number*> {**permit** | **deny**} <*path-expression*>

where:

access-list-number is the access list number.

path-expression is a valid path regular expression.

Example

The commands in the following example configure a router with two AS path access lists. Routes that pass AS path access list 1 are sent to one destination. Routes that pass AS path access list 2 are accepted from another destination. The commands, executed in Global Configuration mode, specify permit and deny conditions for AS path access lists 1 and 2.

```
MOT(config-bgp)#ip as-path access-list 1 permit _200
MOT(config-bgp)#ip as-path access-list 1 permit ^100
MOT(config-bgp)#ip as-path access-list 2 deny _690
MOT(config-bgp)#ip as-path access-list 2 permit.*
```

The next command, executed in Global Configuration mode, enables BGP and specifies an AS. The next commands, executed in Router BGP Configuration mode, define two neighbor peers, and assign the AS path list to one of the neighbor BGP peers. This indicates that outbound routes have the conditions defined in AS path access list 1 applied to it.

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 156.30.10.22 remote-as 200
MOT(config-bgp)#neighbor 160.25.15.10 remote-as 300
MOT(config-bgp)#neighbor 156.30.10.22 filter-list 1 out
```

Creating a Community List

Peers exchange the BGP community attributes when they exchange reachability information with each other. A community is a group of destinations that share a common policy. Based on community, you can control the routing information a BGP speaker accepts, prefers, or distributes to other neighbors.

You can use the following predefined well-known community attributes with the **set-community** command in a route map:

- no-export
- no-advertise
- no-sub-confed export

In addition, you can define a community number to advertise to a specific community number. All destinations belong to the general Internet community by default.

Use the **no export** keyword to disallow advertising to EBGP peers. This is useful in a network that uses IBGP heavily but does not want to share its internal routing entries with its EBGP peers. Use the **no-advertise** keyword to prevent routes from being propagated beyond the local router, even to IBGP peers.

Figure 12-21 details creating a route map based on the network in which Router Boston sets the value of the local preference attribute based on the value of the community attribute. Any route that has a community attribute of 100 matches community list 1 and has its local preference set to 50. Any route that has a community attribute of 200 matches community list 2 and has its local preference set to 25. All other routes do not have their local preference attributes changed, because all routes are members of the internet community.

1. To create a community list that globally accepts or rejects all advertisements, use the **ip community-list** command in Global Configuration mode, as shown below:

   `MOT(config)#`**ip community-list** <*community-list-number*> {**permit** | **deny**} <*community-numbers*>

   where:

   community-list-number is a number that identifies a community list.

   permit indicates accept the advertisements.

   deny indicates reject the advertisements.

   community-numbers are one or more community numbers.

2. To create a community list that accepts or rejects advertisements with a *local AS* community, use the **ip community-list local-as** command in Global Configuration mode as shown below:

   `MOT(config)#`**ip community-list** <*community-list-number*> {**permit** | **deny}** **local-as**

   where:

community-list-number identifies the community list.

permit indicates accept the advertisements.

deny indicates reject the advertisements.

local-as indicates the well-known community Local-AS.

3. To create a community list that accepts or rejects advertisements with a well-known community on the Internet, use the **ip community-list internet** command in Global Configuration mode as shown below:

`MOT(config)#`**ip community-list internet** <*community-list-number*> {**permit** | **deny} internet**

where:

community-list-number identifies the community list.

permit indicates accept the advertisements.

deny indicates reject the advertisements.

internet is the name of the Internet community.

4. To create a community list that accepts or rejects advertisements with "*No-Advertise*" community, use the **ip community-list no-advertise** command in Global Configuration mode as shown below:

`MOT(config)#`**ip community-list** <*community-list-number*> {**permit** | **deny} no-advertise**

where:

community-list-number identifies a community list.

permit indicates accept the advertisements.

deny indicates reject the advertisements.

no-advertise is the name of a well-known community.

5. To create a community list that accepts or rejects advertisements from an AS, or to advertise a route to IBGP peers only, use the **ip community-list no-export** command in Global Configuration mode as shown below:

`MOT(config)#`**ip community-list** <*community-list-number*> {**permit** | **deny**} **no-export**

where:

community-list-number is the route map name.

permit indicates accept the advertisements.

deny indicates reject the advertisements.

no-export is the name of a well-known community.

6. To filter routes based on a community list, use the **community-list** command in Router BGP Configuration mode, as shown below:

`MOT(config-bgp)#`**ip community-list** <*community-list-number*> {**permit** | **deny**} {<*community-numbers*> | **no-export** | **no-advertise** | **local-as** | **internet**}

where:

community-list-number identifies a community list.

community-numbers is a number that identifies a community.

no-export is the name of a well-known community.

no-advertise is the name of a well-known community.

local-as indicates the well-known community Local-AS.

internet is the name of the Internet community.

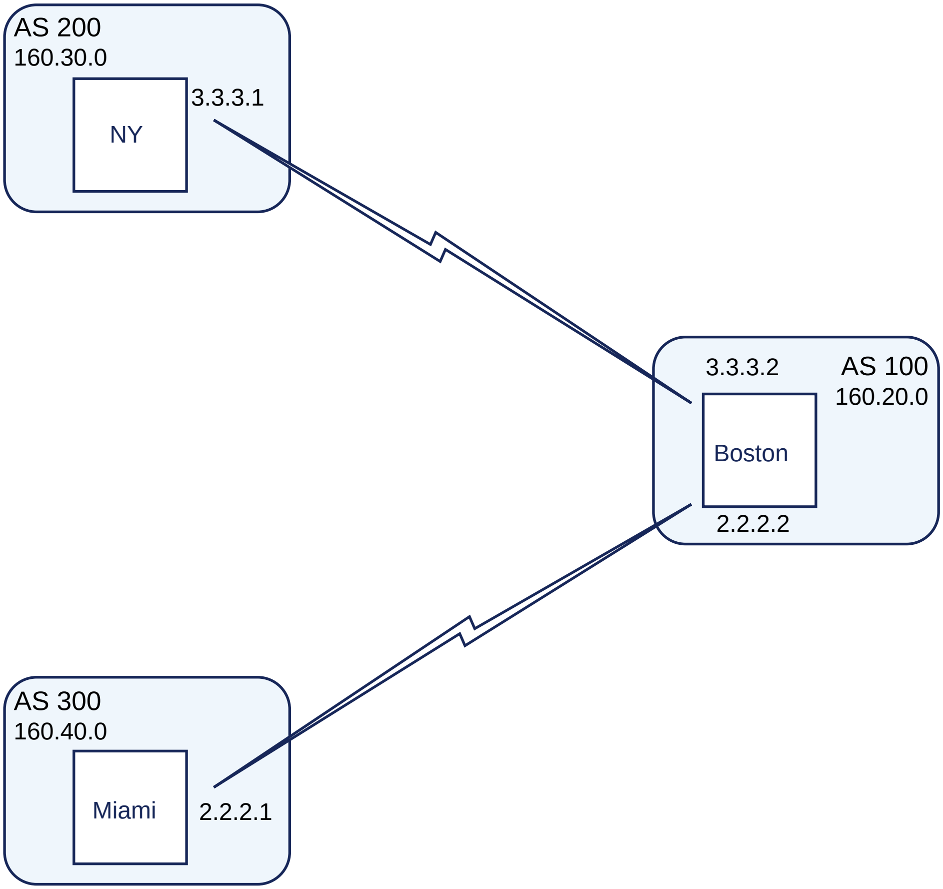


Figure 12-21 Using a Community List

Example

This example uses a community list to filter routes based on the local preference. The following commands, executed in Global Configuration mode, define a community list. Specify community list 1 to permit routes from AS 100 and community list 2 to permit routes from AS 200, as shown in Illustration 12-21.

```
MOT(config)#ip community-list 1 permit 100
MOT(config)#ip community-list 2 permit 200
```

The next commands, also executed in Global Configuration mode, define the first instance of the route map with the appropriate match and set clauses and specify route map 10, instance 10. This permits the route to be accepted and its local preference to be set to 50. The last command indicates that the route is part of the communities defined in Community List 1.

```
MOT(config)#route-map 10 permit 10
MOT(config)#match community 1
MOT(config)#set local preference 50
```

The following commands define the second instance of the route map, route map 10 to permit the route to be accepted and its local preference to be set to 25 if the route is part of the communities defined in Community List 2.

```
MOT(config)#route-map 10 permit 20
MOT(config)#match community 2
MOT(config)#set local preference 25
```

The following commands, executed from Global Configuration mode, enable BGP and specify an AS. As indicated in Figure 12-21, they specify the AS for Router Boston in AS 100. They specify the AS of the BGP neighbors, New York and Miami, to which the route map applies, and apply the route map 10 for all incoming routes from router New York and Miami.

```
MOT(config)#router bgp 100
MOT(config-bgp)#neighbor 3.3.3.1 remote-as 200
MOT(config-bgp)#neighbor 3.3.3.1 route-map 10 in
MOT(config-bgp)#neighbor 2.2.2.1 remote-as 300
MOT(config-bgp)#neighbor 2.2.2.1 route-map 10 in
```

It is assumed that the New York and Miami routers set their outgoing routes (to Boston) to belong to communities 200 and 300, respectively.

Redistributing Routes into BGP

Each routing protocol uses different metrics to transfer routes. Some protocols use hop count metrics, while others use bandwidth and delay attributes to define metrics. When a specific route is redistributed from one routing protocol or domain into another, a common metric must be applied by the receiving protocol. Routes are redistributed to advertise networks on another routing protocol. Figure 12-22 shows Router New York redistributes the routes learned through OSPF protocol from Routers Boston and Los Angeles into BGP.

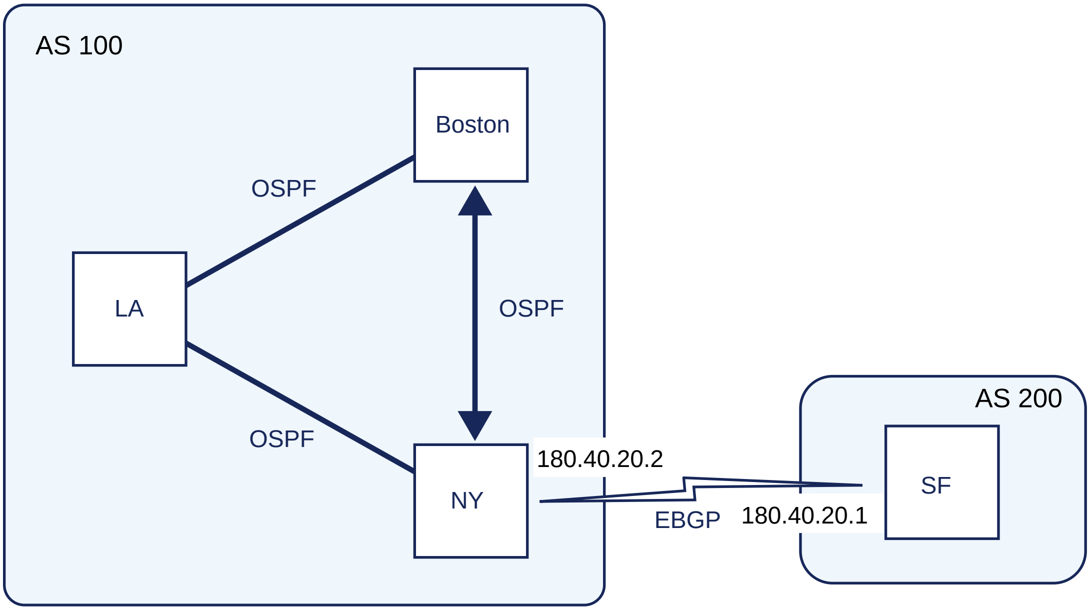


Figure 12-22 Redistributing Routes Learned from OSPF

Follow these steps to redistribute routes into BGP:

1. Enter the BGP routing process in which the routes are to be redistributed, as shown below:

`MOT(config)#`**router bgp** <*n*>

where:

n is the Autonomous System (AS) number.

2. Choose from one or more of the following options to redistribute routes from a specified protocol:

   - Use the **redistribute ospf** command in Router Configuration mode to redistribute OSPF routes into BGP, as shown below:

     `MOT(config-bgp)#`**redistribute ospf** {[**external** | **internal**] | **metric** <*n*> | **route-map** <*map-name*> | <*cr*>}

     where:

     The **external** argument is used to redistribute external OSPF routes.

     The **internal** argument is used to redistribute internal OSPF routes.

     metric <*n*> is the redistribution metric number for OSPF routes.

     route-map <*map-name*> is the OSPF route-map name.

     cr is a command return that redistributes of all OSPF routes.

   - Use the **redistribute connected** command in Router Configuration mode to redistribute connected routes into BGP, as shown below:

     `MOT(config-bgp)#`**redistribute connected** {**metric** <*n*> | **route-map** <*map-name*> | <*cr*>}

     where:

     metric <*n*> is the redistribution metric number for connected routes.

     route-map <*map-name*> is the route-map name for the connected route.

     cr is a command return that redistributes of all connected routes.

   - Use the **redistribute isis** command in Router Configuration mode to redistribute IS-IS routes into BGP, as shown below:

     `MOT(config-bgp)#`**redistribute isis** {**match** [**level-1** | **level-1-2** | **level-2**] | **metric** <*n*> | **route-map** <*map-name*> | weight <*n*> | <*cr*>}

     where:

     The **match** argument is used to choose level 1 ISIS routes only, level 1 and 2 ISIS routes, or level 2 ISIS routes to their destination only.

     metric <*n*> is the redistribution metric number for ISIS routes.

     route-map <*map-name*> is the route-map name for the ISIS route.

weight <*n*> sets the network weight value from 0 to 65535 for redistributing RIP routes into BGP.

cr is a command return that redistributes of all ISIS routes.

- Use the **redistribute rip** command in Router Configuration mode to redistribute RIP routes into BGP, as shown below:

  `MOT(config-bgp)#`**redistribute rip** {**metric** <*n*> | **route-map** <*map-name*> | **weight** <*n*> | <*cr*>}

  where:

  metric <*n*> is the redistribution metric number for RIP routes.

  route-map <*map-name*> is the route-map name for the RIP route.

  weight <*n*> sets the network weight value from 0 to 65535 for redistributing RIP routes into BGP.

  cr is a command return that redistributes of all RIP routes into BGP.

Assigning a Default Metric Value for Redistributed Routes

The default metric function is used to eliminate the need for separate metric definitions for each routing protocol redistribution.

Follow these steps to assign a default metric value for all routes redistributed into BGP:

1. Use the **router bgp** command to enter the BGP routing process in Global Configuration mode, as shown below:

   `MOT(config)#`**router bgp**

2. Use the **default-metric** command in Router Configuration mode to force a routing protocol to use the same metric value for all distributed routes from other routing protocols, as shown below:

   `MOT(config-bgp)#`**default-metric** <*n*>

   where:

   n is the default metric value for all routes that are redistributed into BGP.

Monitoring BGP

Use these **show** commands to monitor BGP:

- **show ip bgp**
- **show ip bgp cidr-only**
- **show ip bgp neighbors**
- **show ip bgp paths**
- **show ip bgp peer-group**
- **show ip bgp summary**
- **show ip bgp regexp**
- **show ip as-path-access-list**
- **show ip community list**
- **show ip protocols** [**summary**]

Use the **show ip bgp community-list** command to display the routes that are permitted by a BGP community list.

```
MOT#show ip bgp community-list <list-num>
```

13

Configuring VRRP

Overview

This chapter describes how to configure the Virtual Router Redundancy Protocol (VRRP) for the BSR 64000™ system using the command line interface (CLI). For further information on the CLI commands described in this chapter, refer to the *BSR 64000 Command Reference Guide*. This chapter discusses the following topics:

- About VRRP
- Initial VRRP Tasks
- Managing VRRP on the BSR
- Gathering Virtual Router Information

About VRRP

VRRP dynamically assigns responsibility for one or more virtual routers to VRRP routers on a LAN. This allows several routers on a multiaccess link to use the same virtual IP address. A VRRP router runs the VRRP protocol in conjunction with one or more other routers attached to a LAN. One router is elected master; the others act as backups in case the master router fails. The election process provides dynamic fail-over in the forwarding responsibility if the master becomes unavailable. This allows any of the virtual router IP addresses on the LAN to be used as the default first-hop router by end-hosts. VRRP provides a higher-availability default path without requiring configuration of dynamic routing or router discovery protocols on every end host.

Note: If you configure a virtual IP on a router with the same IP on a physical interface, then it is assigned a priority of 255 and will always be master unless it fails.

Initial VRRP Tasks

You must complete the following tasks to configure VRRP on the BSR:

- Enabling VRRP
- Creating a Virtual Router
- Configuring a Virtual IP Address
- Specifying Authentication String
- Configuring Primary IP Address
- Enabling a Virtual Router
- Configuring Authentication Type

Enabling VRRP

To enable VRRP on all interfaces that are configured to run VRRP, use the **ip vrrp** command in Global Configuration mode, as shown below. This command enables all interfaces so that multiple virtual routers can be enabled or disabled at one time. Use the **no ip vrrp** command to disable VRRP on all interfaces. VRRP is enabled by default.

```
MOT(config)#ip vrrp
```

Creating a Virtual Router

To create a virtual router, use the **ip vrrp** command in Interface Configuration mode, as shown below. Each virtual router selects its own master and backups, independent of other virtual routers. Each virtual router has a unique virtual MAC address and virtual IP address. Use the **no ip vrrp** command to delete a virtual router.

```
MOT(config-if)#ip vrrp <number>
```

where:

number is the VRID; valid values are 1 to 255.

Example

The following example configures a system to participate in two virtual routers, 1 and 2, on the Ethernet interface 7/0:

```
interface ethernet 7/0
ip vrrp 1
ip vrrp 2
```

Configuring a Virtual IP Address

To configure a virtual IP address or addresses, use the **ip vrrp address** command in Interface Configuration mode, as shown below. VRRP is not enabled for the virtual router until you specify at least one IP address. If you specify one or more IP addresses, those addresses are used as the designated IP address or addresses by associated routers. Use the **no ip vrrp address** to remove the virtual IP address or addresses.

`MOT(config-if)#`**ip vrrp** <*number*>> **address** <*ip-address*> [...<*ip-address*>]

where:

number is the VRID; valid values are 1 to 255.

ip-address is the virtual IP address.

Example

The following example configures the system to use 198.112.190.1 and 20.20.20.1 as the virtual IP addresses of virtual router 1 on the configured Ethernet interface on slot 1, port 0:

```
interface ethernet 1/0
ip vrrp 1 address 198.112.190.1 20.20.20.1
```

Specifying Authentication String

To specify the authentication key for use with the authentication type, text only, use the **ip vrrp authentication key** in Interface Configuration mode, as shown below. This sets a simple text key in VRRP messages. The unencrypted authentication string is transmitted for authentication type simple text in all VRRP protocol messages.

Note: You must configure the same authentication string on all routers associated with a virtual router. An Authentication mismatch does not prevent a router from taking over as the designated master, however, it may cause VRRP to work incorrectly. This can result in lack of communication between virtual routers.

Use the **no ip vrrp authentication key** command to remove the specified authentication string for use with the authentication type, text only.

`MOT(config-if)#`**ip vrrp** <*num:1,255*> **authentication key** <*bounded-string:1,8*>

*num:1,255*the identification of a configured virtual router

*bounded-string*a string of up to 8 characters

Example

The following example shows how to specify the authentication string, text only. The system is configured to use *mot* as the authentication key to operate between other VRRP routers for virtual router 1 on the configured interface.

```
interface ethernet 1/0
ip vrrp 1 authentication key mot
```

Configuring Primary IP Address

To configure the the primary IP address for a virtual router, use the **ip vrrp primary-ip** command in Interface Configuration mode, as shown below. Use the **no ip vrrp primary-ip** command to reset the primary IP address to the smallest value among all real interface addresses for the interface.

`MOT(config-if)#`**ip vrrp** <*num:1,255*> **primary-ip** <*ip-address*>

*num:1,255*the identification of a configured virtual router

*ip-address*IP address to be set as source of outgoing IP packet

Example

This examples sets the primary IP address to 20.20.20.100.

```
interface ethernet 1/0
ip address 10.10.10.100 255.255.255.0
ip address 20.20.20.100 255.255.255.0 secondary
ip vrrp 1 primary-ip 20.20.20.100
```

Enabling a Virtual Router

To enable a virtual router on a configured interface, use the **ip vrrp enable** command in Interface Configuration mode, as shown below. This brings up a specific VRRP router on the interface when the router is enabled. The command brings the VRRP router to either backup or master when the router is enabled, if at least one IP address is configured for the virtual router. Use the **no vrrp enable** command to disable a virtual router on a configured interface.

`MOT(config-if)#`**ip vrrp** <*number*> **enable**

where:

number is the VRID; valid values are 1 to 255.

Example

This example disables the configured virtual router 1 on the interface Ethernet slot 1, port 0

```
interface ethernet 1/0
no ip vrrp 1 enable
```

Configuring Authentication Type

To specify the authentication type for the virtual router on the configured interface, use the **ip vrrp authentication type** command in Interface Configuration mode, as shown below. Use the **no ip vrrp authenticate type** command to remove the specified type of authentication for the virtual router on the configured interface.

`MOT(config-if)#`**ip vrrp** <*number*> **authentication type** {**text**}

where:

number is the VRID; valid values are 1 to 255.

textauthentication type can be simple text

Note: You must configure the same authentication type on all routers associated with a virtual router. An Authentication mismatch does not prevent a router from taking over as the designated master, however, it may cause VRRP to work incorrectly. This can result in lack of communication between virtual routers.

Example

This example configures the system to use simple text authentication when exchanging protocol messages among VRRP routers for virtual router 1. It reverts to no authentication when exchanging protocol messages among VRRP routers for virtual router 2 on the configured interface Ethernet slot 7, port 0:

```
interface ethernet 7/0
ip vrrp 1 authentication type text
no ip vrrp 2 authentication type
```

Managing VRRP on the BSR

The following sections are used to manage VRRP on the BSR:

- Specifying Priority
- Pre-empting a Master
- Specifying Advertisement Interval
- Clearing Statistic Counters

Specifying Priority

To specify the priority of the router to act as master for a virtual router, use the **ip vrrp priority** command in Interface Configuration mode, as shown below. Use the **ip vrrp priority** command to select a master when multiple routers are associated with the same virtual router. If two routers have the same priority, the system compares their primary IP addresses. The router with the higher IP address value takes precedence. A priority of 255 is reserved for VRRP routers that own the virtual IP address. During configuration, the system automatically sets the priority to 255 for the router owning that IP address. This value cannot be changed. The system reserves the value 0 for the master to indicate the relinquishing of responsibility of the virtual router. This value cannot be changed. The default value is 100.

Use the **no** form of this command to restore the default priority value of the router.

`MOT(config-if)#`**ip vrrp** *<number>* **priority** *<priority:1,254>*

where:

number is the VRID; valid values are 1 to 255.

priority is the priority value for the virtual router; valid entries are 1 to 254.

Note: If you set up a virtual IP address on a router with the same IP address on a physical interface, it receives priority 255 and is always master, unless it fails.

Example

This example shows that the system is configured with a priority of 150 for virtual router 1 on the configured interface 1/0.

```
interface ethernet 1/0
ip vrrp 1 priority 150
```

Pre-empting a Master

To configure a higher priority backup that can pre-empt a lower priority master, use the **ip vrrp preempt** command in Interface Configuration mode, as shown below. If virtual routers have IP addresses that do not belong to any router interface, use the **ip vrrp preempt** command to specify which router may pre-empt. Use the **no ip vrrp preempt** command to disable pre-emption of a lower priority master by a higher priority backup.

```
MOT(config-if)#ip vrrp <number> preempt
```

where:

number is the VRID; valid values are 1 to 255.

Note: If the router owns the IP address or addresses associated with the virtual router, the master always pre-empts, regardless of this command setting.

Example

This example configures the system to pre-empt the current master on the configured interface Ethernet slot 1, port 0 for VR1, which has been configured with a higher priority. This command assumes the current master does not own the IP addresses of virtual router 1.

```
interface ethernet 1/0
ip vrrp 1 preempt
```

Specifying Advertisement Interval

To specify the VRRP advertisement messages time interval, use the **ip vrrp timer** command in Interface Configuration mode, as shown below. Use the **no ip vrrp timer** command to restore the default of 1 second.

```
MOT(config-if)#ip vrrp <number> timer <interval>
```

where:

number is the VRID; valid values are 1 to 255.

interval is the advertisement time interval in seconds; valid entries are 1 to 255.

Example

This example configures the system to send VRRP advertisements every three seconds for the virtual router on the configured interface Ethernet 1/0, if the router is configured as the master for the virtual router 1. If not, this interval is the factor that determines the router configured as backup for virtual router 1.

```
interface 1/0
ip vrrp 1 timer 3
```

Clearing Statistic Counters

To reset all statistic counters for all virtual routers, use the **clear ip vrrp** command in any mode except User EXEC mode. This resets the statistic counters of all virtual routers on all interfaces or on specific interfaces with a specific Virtual Router ID (VRID).

`MOT(config)#`**clear ip vrrp** [***** | **ethernet** <*slot*> {/} <*port*> | **ethernet** <*slot*> {/} <*port*> **vrid** <*number*>]

where:

slot is the interface slot number or numbers.

port is the interface port number or numbers.

number is the VRID; valid values are 1 to 255.

Examples

The following example resets all statistic counters for all routers on all interfaces:

```
clear ip vrrp *
```

This example resets statistic counters for all virtual routers on the Ethernet interface on slot 7, port 0:

```
clear ip vrrp ethernet 7/0
```

This example resets statistic counters for the virtual router with the VRID 5 on the Ethernet interface on slot 7, port 0:

```
clear ip vrrp ethernet 7/0 vrid 5
```

Gathering Virtual Router Information

- Monitoring Critical Link State
- Monitoring Virtual Router Information
- Monitoring Ethernet Virtual Routers
- Obtaining Summary Information

Monitoring Critical Link State

To configure one or more ip addresses for a virtual router to monitor as critical link states, use the **ip vrrp verify-availability** command in Interface Configuration mode, as shown below. Use this command to configure a virtual router to monitor a link state of another interface. If one or more IP addresses are configured for monitoring and all monitored links are down, the virtual router is brought down automatically. The master relinquishes responsibility by sending an advertisement 0. If at least one monitored link comes back up, the associated virtual router is brought back up automatically. Use the **no ip vrrp verify-availability** command to delete one or more designated IP address from the virtual router.

MOT(config-if)#**ip vrrp** <*num:1,255*> **verify-availability** <*address*> [...<*ip-address*>]

where:

number is the VRID; valid values are 1 to 255.

ip-address is the IP address this router monitors.

Example

In this example, the system is configured to monitor 198.112.190.11 and 20.20.20.11 for their availability. If both links are down, then VRRP 1 is brought down automatically.

```
interface ethernet 1/0
ip vrrp 1 verify-availability 198.112.190.11 20.20.20.11
```

Monitoring Virtual Router Information

To displays detailed information on virtual routers that are configured for VRRP, use the **show ip vrrp** command in all modes except User EXEC mode. This command verifies router virtual status.

MOT(config-if)#**show ip vrrp**

For example:

The **show ip vrrp** command below results in the following display:

```
VRRP Global Statistic:
    Recv: 0 checksum errors, 0 bad version
          60621 bad vrid, 0 bad packet size

Interface ethernet 7/0, VRID 1:
   Status:  vrrp is enabled, in state backup, priority is
   100
      advert interval is 1 sec, preempt mode is on
      use no authentication, up since 16:46:27 ago
      last state change 3:13:14 ago
      primary ip is 10.10.10.202, total virtual ip
      address(es) is 1
            virtual ip address(es): 10.10.10.19
   Advertisement Recv: 56615 total, 0 mismatched interval
      0 bad vrrp type, 0 bad packet len, 0 as master
      0 mismatched ip ttl, 0 mismatched addrlist
      0 bad authentication type, 0 mismatched authentication
type
      0 failed authentication, 0 zero priority
   Advertisement Sent: 4 total, 0 zero priority
   Become master: 3 times


Interface ethernet 7/2, VRID 3:
   Status:  vrrp is enabled, in state master, priority is 255
   advert interval is 1 sec, preempt mode is on
   use no authentication, up since 16:46:6 ago
   last state change 16:46:6 ago
   primary ip is 10.10.20.20, total virtual ip
   address(es) is 1
            virtual ip address(es): 10.10.20.20
Advertisement Recv: 126 total, 0 mismatched interval
   0 bad vrrp type, 0 bad packet len, 126 as master
   0 mismatched ip ttl, 0 mismatched addrlist
```

```
 0 bad authentication type, 0 mismatched authentication
 type
         0 failed authentication, 0 zero priority
Advertisement Sent: 60367 total, 0 zero priority
Become master: 1 times
```

Monitoring Ethernet Virtual Routers

To display detailed information on all of the virtual routers with an Ethernet interface, use the **show ip vrrp ethernet** command in User EXEC mode, as shown below.

MOT#**show ip vrrp ethernet** <*slot*> / <*port*> [<*number*>]

where:

slot is the interface slot number.

port is the interface port number.

number is the VRID; valid values are 1 to 255.

Examples

The following are examples of the **show ip vrrp ethernet** command and their displays.

Example 1:

MOT(config-if)# show ip vrrp ethernet 7/0

```
Interface ethernet 7/0, VRID 1:
   Status:  vrrp is enabled, in state backup, priority is
    100
       advert interval is 1 sec, preempt mode is on
       use no authentication, up since 16:49:20 ago
       last state change 3:16:7 ago
       primary ip is 10.10.10.202, total virtual ip
       address(es) is 1
            virtual ip address(es): 10.10.10.19
   Advertisement Recv: 56778 total, 0 mismatched interval
            0 bad vrrp type, 0 bad packet len, 0 as master
            0 mismatched ip ttl, 0 mismatched addrlist
            0 bad authentication type, 0 mismatched
             authentication type
            0 failed authentication, 0 zero priority
   Advertisement Sent: 4 total, 0 zero priority
   Become master: 3 times

Interface ethernet 7/0, VRID 3 :
   Status:  vrrp is enabled, in state master, priority is
   255
        advert interval is 1 sec, preempt mode is on
        use no authentication, up since 16:48:59 ago
        last state change 16:48:59 ago
        primary ip is 10.10.20.20, total virtual ip
        address(es) is 1
             virtual ip address(es): 10.10.20.20
   Advertisement Recv: 126 total, 0 mismatched interval
        0 bad vrrp type, 0 bad packet len, 126 as master
        0 mismatched ip ttl, 0 mismatched addrlist
        0 bad authentication type, 0 mismatched
        authentication type
0 failed authentication, 0 zero priority
   Advertisement Sent: 60540 total, 0 zero priority
   Become master: 1 times
```

Example 2:

```
MOT(config-if)#show ip vrrp ethernet 7/0 vrid 3
```

```
Interface ethernet 7/0, VRID 3:
   Status:  vrrp is enabled, in state master, priority is 255
            advert interval is 1 sec, preempt mode is on
            use no authentication, up since 16:49:48 ago
            last state change 16:49:48 ago
            primary ip is 10.10.20.20, total virtual ip
          address(es) is 1
            virtual ip address(es): 10.10.20.20
   Advertisement Recv: 126 total, 0 mismatched interval
            0 bad vrrp type, 0 bad packet len, 126 as
             master
            0 mismatched ip ttl, 0 mismatched addrlist
            0 bad authentication type, 0 mismatched
                 authentication type
            0 failed authentication, 0 zero priority

   Advertisement Sent: 60589 total, 0 zero priority
   Become master: 1 times
```

Obtaining Summary Information

To show summary information on all VRRP routers configured on all interfaces on the router, use the **show ip vrrp summary** command in User EXEC mode, as shown below.

```
MOT#show ip vrrp summary
```

Example

The example below shows summary information:

```
MOT(config-if)# show ip vrrp summary
```

```
Global vrrp configuration is enabled, total vrrp configured: 2

Interface VRID Enble  State Pri Tmr Primary IP Addr Virtual IP Addr StateChg
----------- ------ -------- ------- --- ----- --------------------- ------------------- -----------
ethernet  7/0 1 true backup 100   1  10.10.10.202   10.10.10.19 13:15:46
ethernet  7/0 3  true master  255   1    10.10.20.20         10.10.20.20
16:48:38
```

14

Configuring Packet Over SONET

Overview

This chapter describes how to configure the Packet Over SONET (POS) interface for the BSR 64000™ using the command line interface (CLI). For further information on the CLI commands described in this chapter, refer to the *BSR 64000 Command Reference Guide*.

This chapter discusses the following topics:

- About SONET/SDH
- POS Interface Configuration Tasks
- Configuring SONET
- Configuring SONET Alarms
- Changing the POS Signal Rate
- Specifying the POS Loopback Mode Type
- Gathering POS Network Information

Table 14-1 describes the SONET transmission types and rates that are available for the BSR 64000.

Table 14-1 SONET Transmission Types and Rates

Transmission Type	Rate
OC-1	51.84 Mbps
OC-3	155.52 Mbps
OC-12	622.08 Mbps

About SONET/SDH

The Synchronous Optical Network (SONET) standard provides for data transmission over fiber optic cable and high-bandwidth utilization and efficiency over Internet links. The SONET standard defines industry interface standards at the physical layer of the OSI seven-layer model. This standard defines a hierarchy of interface rates that allow data streams at different rates to be multiplexed. SONET establishes Optical Carrier (OC) levels from 51.8 Mbps to 2.48 Gbps. Prior rate standards used by different countries specified rates that were not compatible for multiplexing. Synchronous Digital Hierarchy (SDH), the international equivalent of SONET, defines a standard rate of transmission at 155.52 Mbps. With the implementation of SONET/SDH, communication carriers throughout the world can interconnect existing digital carrier and fiber optic systems.

Specifications

The BSR POS interface supports the following Request for Comment (RFC) specifications:

- RFC 1619, *PPP over SONET/SDH*
- RFC 1662, *PPP in HDLC-like Framing*
- RFC 2558 SONET/SDH

POS Features

The BSR supports the following POS features:

- Payload scrambling
- Clock source configuration
- Maximum transmission unit size configuration
- SONET/SDH framing
- Cyclic redundancy check
- Alarm Reporting
- Automatic Protection Switching

POS Interface Configuration Tasks

The following POS module configuration tasks are mandatory:

- Configuring the POS Interface
- Configuring PPP
- Configuring the Network Clock Source for SONET

Configuring the POS Interface

Follow these steps to configure the POS interface:

1. To configure the physical POS interface, use the **interface pos** command in Global Configuration mode, as shown below:

   `BSR(config)#`**interface pos** <*slot*>/<*interface*>

   where:

   slot is the POS module slot on the BSR 64000 chassis.

   interface is the line POS interface on the POS module.

2. To enter the IP address and subnet mask for the POS interface, use the **ip-address** command in Interface Configuration mode, as shown below:

   `MOT(config-if)#`**ip-address** {<*ip-address*> <*subnet-mask*>}

   where:

   ip-address is the IP address of the POS interface

   subnet-mask is the subnet mask of the POS interface

3. To optionally set a secondary IP address and subnet mask for the POS interface, use the **ip-address secondary** command in Interface Configuration mode, as shown below:

   `MOT(config-if)#`**ip-address** {<*ip-address*> <*subnet-mask*>} **secondary**

   where:

   ip-address is the IP address of the POS interface

   subnet-mask is the subnet mask of the POS interface

Configuring PPP

Table 14-2 describes the PPP features and commands that are available on the POS module:

Table 14-2 PPP Commands

Command	Description	Default	Value
ppp mtu	Maximum transmission unit packet size allowed on the POS interface	**1500** bytes	**72** to **1500** bytes
ppp negotiation-count	Number of attempts before PPP connection is dropped	**10** attempts	**1** to **100** attempts
ppp timeout ncp	Maximum wait time for network layer to negotiate before PPP connection is dropped	**10** seconds	**1** to **2147483** seconds
ppp timeout retry	Maximum wait time for a negotiation response before PPP connection is dropped	**10** seconds	**0** to **2147483**

Follow these steps to configure PPP on the POS interface:

1. To set the size of the MTU packet allowed on the POS interface, use the **ppp mtu** command in Interface Configuration mode, as shown below:

   `MOT(config-if)#`**ppp mtu** <*bytes*>

   where:

   bytes is the number of bytes from permitted for the MTU; value is between 72 to 1500; default is 1500 bytes.

2. To set the number of attempts for successful negotiation before terminating the PPP link on the POS interface, use the **ppp negotiation-count** command in Interface configuration mode, as shown below:

   `MOT(config-if)#`**ppp negotiation-count** <*n*>

   where:

n is the number of permitted negotiation attempts from 1 to 100.

3. To set the maximum PPP wait time for the network layer to negotiate before disconnecting the PPP link on the POS interface when there is no activity on the link, use the **ppp timeout ncp** command in Interface Configuration mode, as shown in the following example.

```
MOT(config-if)#ppp timeout ncp <seconds>
```

where:

seconds is the number of seconds of inactivity before the PPP link is disconnected; default is 10 seconds.

4. To set the maximum PPP wait time for a negotiation response before disconnecting the PPP link on the POS interface when there is no activity on the link, use the **ppp timeout** command in Interface Configuration mode, as shown in the following example.

```
MOT(config-if)#ppp timeout retry <seconds>
```

where:

seconds is the number of seconds of inactivity before the PPP link is disconnected; default is 10 seconds.

Configuring the Network Clock Source for SONET

The network clock source is set for the SONET link to avoid undesirable conditions related to timing synchronization such as jitter and wander. Jitter refers to the short-term instabilities in network signal timing. Wander refers to long-term random variations of the significant instances of a digital signal from their ideal position in time. The internal clock on the POS module is enabled by default so that it references the network clock source for the BSR 64000.

Table 14-3 describes the network clocking commands that are available on the BSR and the POS module:

Table 14-3 Network Clocking Commands

Command	Description	Default	Value
network-clock-select [**1** | **2**] **bits e1**	Enables the E1 Building Integrated Timing Supply (BITS) network clocking to be derived from the central office (CO) BITS source.	none	**1** sets the priority of the clocking source to primary or **2** sets the priority of the clocking source to secondary. Input **a** or **b** port on SRM I/O module.
network-clock-select [**1** | **2**] **bits t1**	Enables the T1 Building Integrated Timing Supply (BITS) network clocking to be derived from the central office (CO) BITS source.	none	**1** sets the priority of the clocking source to primary or **2** sets the priority of the clocking source to secondary. Input **a** or **b** port on SRM I/O module.
network-clock-select pos	Enables the network clocking to be derived from a specific POS module and interface.	none	slot and interface number
pos internal-clock	Set the clock source on the POS interface	**internal**	**internal** or **recovered**

Setting the Primary BITS Network Clocking Source

Follow these steps to set the primary network clocking source:

1. Use the **network-clock-select 1 bits** command in Global Configuration mode to identify the primary network clock and define the network clocking port on the SRM I/O module from which the T1 or E1 BITS signal is derived, as shown in the following example:

   BSR(config)#**network-clock-select 1 bits** [**t1** {**esf-b8zs** | **sf-d4** | **slc96** | **t1dm**} | **e1** {**pcm31-crc** | **pcm31-hdb3** | **pcm31-nocrc**}] [**a** | **b**]

   where:

   1 is the priority assigned to the primary network clock.

t1 specifies a T1 BITS signal.

esf-b8zs is ESF framing with B8ZS line coding.

sf-d4 is SF-D4 framing with AMI line coding.

slc96 is SLC96 framing with AMI line coding.

t1dm is T1DM framing with AMI line coding

e1 specifies an E1 BITS signal.

pcm31-crc is PCM-31 framing with AMI line coding and CRC Multiframe support.

pcm31-hdb3 is PCM-31 framing with HDB3 line coding and CRC Multiframe support.

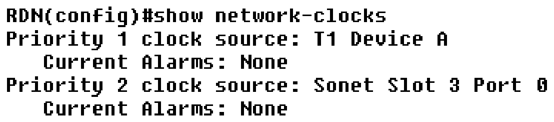
pcm31-nocrc is PCM-31 framing with AMI line coding and no CRC Multiframe support.

a specifies the Input A port on the SRM I/O module.

b specifies the Input B port on the SRM I/O module.

2. To verify the primary network clock source information that you have configured, use the **show network-clocks** command in Global Configuration mode, as shown in the following example:

   `BSR(config)#`**show network-clocks**



```
RDN(config)#show network-clocks
Priority 1 clock source: T1 Device A
   Current Alarms: None
Priority 2 clock source: Sonet Slot 3 Port 0
   Current Alarms: None
```

Figure 14-1 show network-clocks Command Output

Setting the Secondary BITS Network Clocking Source

Follow these steps to set the secondary network clocking source:

1. Use the **network-clock-select 2 bits** command in Global Configuration mode to identify the secondary network clock and define the network clocking port on the SRM I/O module from which the T1 or E1 BITS signal is derived, as shown in the following example:

BSR(config)#**network-clock-select 2 bits** [**t1** {**esf-b8zs** | **sf-d4** | **slc96** | **t1dm**} | **e1** {**pcm31-crc** | **pcm31-hdb3** | **pcm31-nocrc**}] [**a** | **b**]

where:

2 is the priority assigned to the secondary network clock.

t1 specifies a T1 BITS signal.

esf-b8zs is ESF framing with B8ZS line coding.

sf-d4 is SF-D4 framing with AMI line coding.

slc96 is SLC96 framing with AMI line coding.

t1dm is T1DM framing with AMI line coding

e1 specifies an E1 BITS signal.

pcm31-crc is PCM-31 framing with AMI line coding and CRC Multiframe support.

pcm31-hdb3 is PCM-31 framing with HDB3 line coding and CRC Multiframe support.

pcm31-nocrc is PCM-31 framing with AMI line coding and no CRC Multiframe support.

a specifies the Input A port on the SRM I/O module.

b specifies the Input B port on the SRM I/O module.

2. To verify the secondary network clock source information that you have configured, use the **show network-clocks** command in Global Configuration mode, as shown in the following example:

   BSR(config)#**show network-clocks**

Deriving the Network Clocking Source from a POS Interface

Follow these steps to derive network clocking from a POS interface:

1. Issue the **network-clock-select pos** command to enable the network timing to be derived from the clocking recovered from a specified POS module interface, in Global Configuration mode as shown in the following example:

   BSR(config)#**network-clock-select** [**1** | **2**] **pos** <*slot*>/<*interface*>

where:

1 is the priority assigned to the primary network clock.

2 is the priority assigned to the secondary network clock.

slot is the POS slot on the BSR.

interface is the POS interface from which the clocking is recovered.

2. To verify the network clock source information that you have configured, use the **show network-clocks** command in Global Configuration mode, as shown in the following example:

```
BSR(config)#show network-clocks
```

Setting Clock Recovery from the Received SONET Signal

Follow these steps to set clock recovery from the received SONET signal:

1. Enter the POS interface from which the clock is to be recovered, using the **interface pos** command in Global Configuration mode, as shown in the following example:

```
BSR(config)#interface pos <slot>/<interface>
```

where:

slot is the POS module slot on the BSR 64000 chassis.

interface is the line POS interface on the POS module.

2. To use the recovered clock from the received SONET signal, use the **no pos internal-clock** command in Interface Configuration mode, as shown below:

Note: Internal clock mode is enabled by default for the POS module to gather its timing from the network clocks that are configured on the SRM. If the internal clocking function is disabled, clock recovery is taken from the received SONET signal.

```
BSR(config-if)#no pos internal-clock
```

To return to the default, use the **pos internal-clock** command in Interface Configuration mode, as shown below:

```
BSR(config-if)#pos internal-clock
```

Configuring SONET

The following sections describe how to set a variety of SONET commands:

- Optionally Disabling SONET Payload Scrambling
- Changing the SONET Framing Type
- Changing the CRC Function on the POS Interface
- Defining SONET Frame Overhead Bytes

Table 14-4 describes the SONET features and commands that are available on the POS module:

Table 14-4 SONET Commands

Command	Description	Default	Value
pos scramble	Scrambling algorithm used for the SONET payload	enabled	disabled or enabled
pos framing	POS interface framing type	**sonet**	**sdh** or **sonet**
pos crc	Number of bits used for CRC checking	**16**	**16** or **32** check digits per frame. Also the crc can be optionally set for **big-endian** byte order.
pos flag c2	The c2 byte in the SONET frame is the path signal identifier.	0xCF	Hexadecimal value for the protocol encapsulation. For example, PPP is 0xCF.
pos flag c2-exp	Path signal identifier expected from the far end SONET equipment	0xCF	Hexadecimal value for the protocol encapsulation.
pos flag j0	Defines hexadecimal single byte message	0x1	Hexadecimal value.

Table 14-4 SONET Commands

Command	Description	Default	Value
pos flag j0 16byte	Defines 16 byte section trace message	undefined	Text string containing the 16 byte sequence.
pos flag j1 16byte	Defines 16 byte path trace message	undefined	Text string containing the 16 byte sequence.
pos flag j1 64byte	Defines 64 byte path trace message	undefined	Text string containing the 64 byte sequence.
pos flag j1	Defines hexadecimal path single byte message	0x0	Hexadecimal value.
pos flag s1	Identifies the timing source for SONET frame synchronization	0x0	Hexadecimal value.

Optionally Disabling SONET Payload Scrambling

SONET optical interface signals use binary line encoding, and therefore must be scrambled to assure an adequate number of transitions of zeros to ones, and ones to zeros.

If the remote SONET device does not support scrambling, payload scrambling may need to be disabled.

Payload scrambling is enabled by default on the POS interface. To disable payload scrambling, use the **no pos scramble** command in Interface Configuration mode, as shown below:

```
BSR(config-if)#no pos scramble
```

Note: When payload scrambling is enabled, both sides of the connection must be using the same scrambling algorithm.

To return to the default, which is payload scrambling, use the **pos scramble** command in Interface Configuration mode, as shown below:

```
BSR(config-if)#pos scramble
```

Changing the SONET Framing Type

The BSR framing type default is SONET. Follow these steps to change the framing type:

1. To set the framing type for SDH, use the **pos framing sdh** command in Interface Configuration mode, as shown in the following example.

   ```
   MOT(config-if)#pos framing sdh
   ```

2. To set the framing type back to SONET, use the **pos framing sonet** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#pos framing sonet
   ```

Changing the CRC Function on the POS Interface

Cyclic Redundancy Check (CRC) error-checking uses a calculated numeric value to detect errors in transmitted data. The values 16 and 32 indicate the number of check digits per frame for calculating the frame check sequence (FCS). Both the sender and receiver must use the same setting. The default is 16. To set the number of bits used for CRC on the POS interface, use the **crc** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#crc [16 | 32] <big-endian>
```

where:

16 is the bit size.

32 is the bit size.

big-endian is the byte ordering. If nothing is specified “little-endian” is assumed.

Defining SONET Frame Overhead Bytes

SONET overhead bytes identify information in the SONET frame. Refer to the GR-253-CORE document for STS Path Signal Label Assignments and the hexadecimal values required for the SONET frame overhead bytes.

Follow these steps to specify flags for the SONET frame overhead bytes:

1. The c2 byte of the SONET frame is a path signal identifier. To set the hexadecimal value for the c2 byte, use the **pos flag c2** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#pos flag c2 <hexnum>
   ```

   where:

   hexnum is the hexadecimal value for the c2 byte.

   For example:

   - PPP encapsulation is **0xCF**
   - HDLC encapsulation is **0x16**
   - ATM encapsulation is **0x13**
   - Unequipped is **0x00**
   - Equipped non-specific is **0x01**

2. The c2 byte is the path signal label expected from the far end SONET equipment. To set the hexadecimal value for the c2 byte for the SONET frame payload for the PPP or HDLC protocols, use the **pos flag c2-exp** command in Interface Configuration mode as shown below:

   ```
   MOT(config-if)#pos flag c2-exp <hexnum>
   ```

   where:

   hexnum is the hexadecimal value for the c2 byte.

3. If the received c2 byte differs from the expected c2 byte, a payload mismatch occurs. To view a payload mismatch for the c2 byte, use the **show controllers pos** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#show controllers pos
   ```

4. To configure the j0 byte as a hexadecimal section trace byte, use the **pos flag j0** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#pos flag j0 <hexnum>
   ```

   where:

   hexnum is the hexadecimal value for the j0 byte.

5. To configure the j0 byte as a 16 byte section trace message, use the **pos flag j0 16byte** command in Interface Configuration mode, as shown below:

   ```
   MOT(config-if)#pos flag j0 16byte <string>
   ```

   where:

   string is a text string containing the byte *15* sequence.

Note: The first byte contains an automatically generated CRC, the remaining 15 bytes are user defined.

6. To configure the j1 byte as a 16 byte path message, use the **pos flag j1 16byte** command in Interface Configuration mode as shown below:

   ```
   MOT(config-if)#pos flag j1 16byte <string>
   ```

   where:

   string is a text string containing the byte *15* sequence.

Note: The first byte contains an automatically generated CRC, the remaining 15 bytes are user defined.

7. To configure the j1 byte as a 64 byte path message, use the **pos flag j1 64byte** command in Interface Configuration mode as shown below:

   ```
   MOT(config-if)#pos flag j1 64byte <string>
   ```

   where:

string is a text string containing the 62 byte sequence. The remaining 2 bytes are automatically set to CR/LF for framing purposes.

8. To configure the j1 byte as a single hex byte, use the **pos flag j1** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#pos flag j1 <hexnum>
```

where:

hexnum is the hexadecimal value for the j1 byte.

9. The s1 byte identifies the timing source for SONET frame synchronization. To set the hexadecimal value for the s1 byte, use the **pos flag s1** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#pos flag s1 <hexnum>
```

where:

hexnum is the hexadecimal value for the s1 byte.

Note: Only the lower four bits of the s1 byte are actually configured.

Configuring SONET Alarms

The following sections discuss SONET alarm thresholds, alarm reports, and alarm indicators:

- Setting Alarm Thresholds
- Setting Alarm Reporting
- Setting the Line Alarm Indication Signal

Table 14-5 describes the SONET alarms that can be configured on the POS module:

Table 14-5 SONET Alarm Commands

Command	Description	Default	Value
pos threshold b1-tca	The BER threshold for the b1 threshold crossing alarm (TCA)	**4** (10E-4 Rate)	**3** to **9** (10E-3 to 10E-9 Rate)
pos threshold b2-tca	The BER threshold for the b2 threshold crossing alarm (TCA)	**4** which is 10E-4 Rate.	**3** to **9** (10E-3 to 10E-9 Rate)
pos threshold b3-tca	The BER threshold for the b3 threshold crossing alarm (TCA)	**4** which is 10E-4 Rate.	**3** to **9** (10E-3 to 10E-9 Rate)
pos report	SONET alarm reporting	No reports are set.	**all**, **b1-tca**, **b2-tca**, **b3-tca**, **lais**, **lrdi**, **pais**, **plop**, **prdi**, **rdool**, **sd-ber**, **sf-ber**, **slof**, **slos**
pos ais-shut	Controls whether an Alarm Indication Signal - Line (AIS-L) is sent to the far end SONET device when the POS interface is placed in an administratively shutdown state	disabled	enabled or disabled

Setting Alarm Thresholds

Use the SONET alarm thresholds discussed in this section to evaluate network performance. Alarm thresholds define the Bit Error Rate (BER) threshold values for specific control bits in SONET frames. The default SONET alarm thresholds are adequate for most POS installations. Refer to Table 14-10 for more information on each alarm threshold.

Follow these steps to set SONET alarm thresholds:

1. To configure the BER threshold for the b1 threshold crossing alarm (TCA), use the **pos threshold b1-tca** command in Interface Configuration mode as shown below:

```
MOT(config-if)#pos threshold b1-tca <3-9>
```

where:

<**3-9**> is the 10E-3 to 10E-9 Rate.

2. To configure the BER threshold for the b2 TCA, use the **pos threshold b2-tca** command in Interface Configuration mode as shown below:

```
MOT(config-if)#pos threshold b2-tca <3-9>
```

where:

<**3-9**> is the 10E-3 to 10E-9 Rate.

3. To configure the BER threshold for the b3 TCA, use the **pos threshold b3-tca** command in Interface Configuration mode as shown below:

```
MOT(config-if)#pos threshold b3-tca <3-9>
```

where:

<**3-9**> is the 10E-3 to 10E-9 Rate.

4. To return to the default setting, use the **no pos threshold** command in POS Interface Configuration mode as shown below:

```
MOT(config-if)#no pos threshold [b1-tca | b2-tca | b3-tca] <rate>
```

where:

rate is the numeric rate of error.

Setting Alarm Reporting

No SONET alarms are reported by default. Refer to Table 14-10 for command option alarm reporting descriptions. To select SONET alarms to be logged to the POS interface console, use the **pos report** command in Interface Configuration mode as shown below:

```
MOT(config-if)#pos report [all | b1-tca | b2-tca | b3-tca | lais | lrdi | pais | plop
| prdi | rdool | sd-ber | sf-ber | slof | slos]
```

To disable logging of selected SONET alarms, use the **no pos report** command in POS Interface Configuration mode as shown below:

```
MOT(config-if)#no pos report [all | b1-tca | b2-tca | b3-tca | lais | lrdi | pais |
plop | prdi | rdool | sd-ber | sf-ber | slof | slos]
```

Table 14-6 describes the SONET reports that you can configure:

Table 14-6 SONET Alarm Report Selections and Descriptions

Alarm Report	Description
all	All possible alarm reporting
b1-tca	B1 BER TCA errors
b2-tca	B2 BER TCA errors
b3-tca	B3 BER TCA errors
lais	Line alarm indication signal
lrdi	Line remote defect indicator
pais	Path alarm indication signal
plop	Path loss of pointer
prdi	Path remote defect indicator
rdool	Remote data out of lock
slof	Section loss of frame error
slos	Section loss of signal error

Setting the Line Alarm Indication Signal

The **pos ais-shut** command controls whether an Alarm Indication Signal - Line (AIS-L) is sent to the far end SONET device when the POS interface is placed in an administrative shutdown state. To configure whether an Alarm Indication Signal-Line (AIS-L) is sent when a POS interface is shut down, use the **pos ais-shut** command in POS Interface Configuration mode as shown below:

Note: If the far end SONET device has Automatic Protection Switching (APS) configured, an APS switch can be forced by using the **pos ais-shut** command.

```
MOT(config-if)#pos ais-shut
```

To disable sending the line alarm indication signal on an administrative shutdown, use the **no pos ais-shut** command in POS Interface Configuration mode as shown below:

```
MOT(config-if)#no pos ais-shut
```

Changing the POS Signal Rate

Table 14-7 describes the SONET signal mode command:

Table 14-7 SONET Signal Mode Command

Command	Description	Default	Value
pos signal mode	POS module SONET signal	**oc3**	**oc12** or **oc3**

To optionally change the POS module SONET signal rate for OC12, use the **pos signal mode oc12** command in Interface Configuration mode as shown below:

```
MOT(config-if)#pos signal mode oc12
```

where:

oc12 is signal rate 622 Mbps.

To return to the default signal rate, use the pos signal mode oc3 command in Interface Configuration mode as shown below:

```
MOT(config-if)#pos signal mode oc3
```

where:

oc3 is signal rate 155 Mbps.

Specifying the POS Loopback Mode Type

You can use POS loopback commands to isolate the fault on an end-to-end circuit, especially when the circuit is down.

Table 14-8 describes the POS loopback mode command:

Table 14-8 SONET Signal Mode Command

Command	Description	Default	Value
pos loop	POS loopback mode type	none	**internal**, **line**, **txrx-line**, **txpos-rxpos**

Follow these steps to specify the POS loopback mode:

1. To specify that the data transmitted out of the SONET framer device is directly looped to the receive side of the SONET framer device, use the **loop internal** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#loop internal
```

2. To specify the line loopback mode, which is used to connect the received SONET network signal directly to the transmitted SONET network signal, use the **loop line** command in Interface Configuration mode, as shown below:

```
MOT(config-if)#loop line
```

Note: When configured in line loop mode, the BSR never receives data from the network. Use the **no loop** command to clear any loopbacks.

3. To specify that the data transmitted out of the SONET transceiver device is directly looped to the receive side of the SONET transceiver device, use the **loop txrx-line** command in Interface Configuration mode as shown below:

```
MOT(config-if)#loop txrx-line
```

4. To specify that any packets sent from the transmit packet FIFO are looped back to the receive packet FIFO, use the **loop txpos-rxpos** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**loop txpos-rxpos**

Gathering POS Network Information

The following sections describe how to use **show** commands to gather POS information for the PPP link, SONET interface and the physical SONET link:

- Displaying PPP Link and Statistics Information
- Displaying POS Interface Information
- Displaying Physical SONET Link and Alarm Information

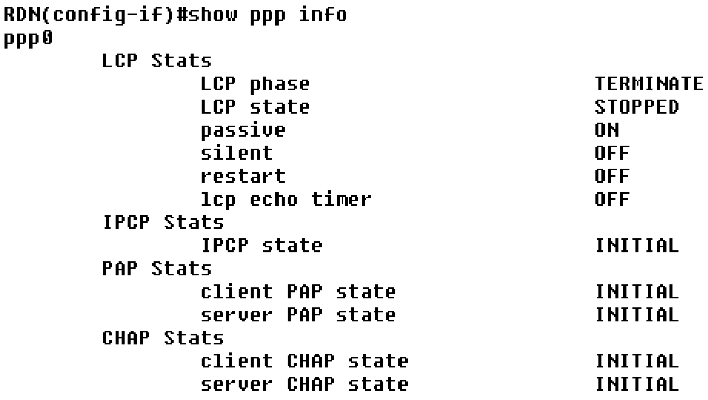
Displaying PPP Link and Statistics Information

Follow these steps to display PPP link and statistics information:

1. To display PPP link information, use the **show ppp info** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**show ppp info**

   Figure 14-2 displays the **show ppp info** command output:



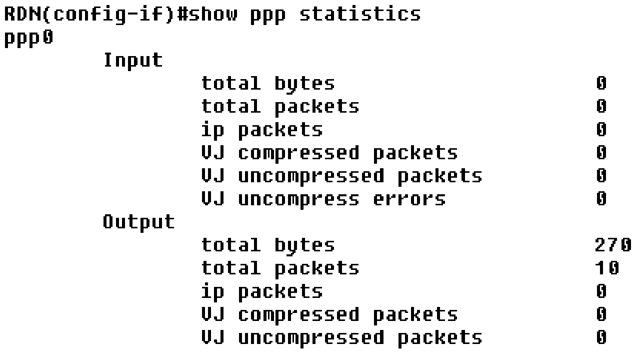
```
RDN(config-if)#show ppp info
ppp0
        LCP Stats
                LCP phase                       TERMINATE
                LCP state                       STOPPED
                passive                         ON
                silent                          OFF
                restart                         OFF
                lcp echo timer                  OFF
        IPCP Stats
                IPCP state                      INITIAL
        PAP Stats
                client PAP state                INITIAL
                server PAP state                INITIAL
        CHAP Stats
                client CHAP state               INITIAL
                server CHAP state               INITIAL
```

Figure 14-2 show ppp info Command Output

2. To display PPP statistics, use the **show ppp statistics** command in Interface Configuration mode, as shown below:

   MOT(config-if)#**show ppp info**

   Figure 14-3 displays the **show ppp statistics** command output:



```
RDN(config-if)#show ppp statistics
ppp0
        Input
                total bytes                     0
                total packets                   0
                ip packets                      0
                VJ compressed packets           0
                VJ uncompressed packets         0
                VJ uncompress errors            0
        Output
                total bytes                     270
                total packets                   10
                ip packets                      0
                VJ compressed packets           0
                VJ uncompressed packets         0
```

Figure 14-3 show ppp statistics Command Output

Displaying POS Interface Information

To display configuration information for the POS interface, use the **show interface pos** command in Interface Configuration mode, as shown below:

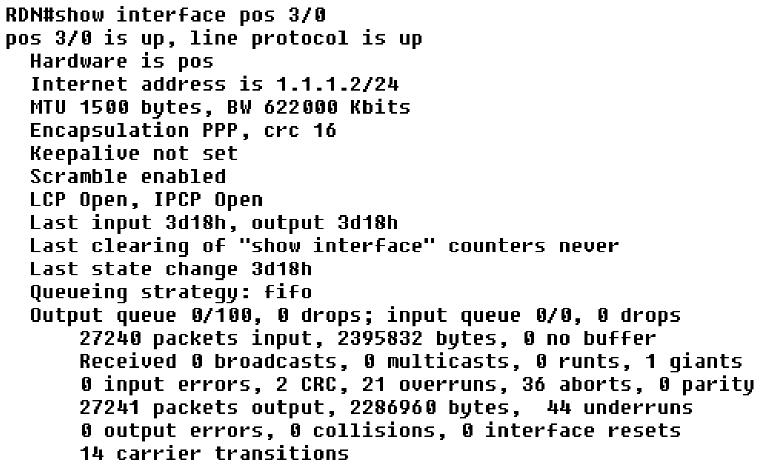
MOT(config-if)#**show interface pos** <*slot*>/<*interface*>

where:

slot is the POS module slot on the BSR 64000 chassis

interface is the line POS interface on the POS module.

Figure 14-4 displays the command output for the POS interface:



```
RDN#show interface pos 3/0
pos 3/0 is up, line protocol is up
  Hardware is pos
  Internet address is 1.1.1.2/24
  MTU 1500 bytes, BW 622000 Kbits
  Encapsulation PPP, crc 16
  Keepalive not set
  Scramble enabled
  LCP Open, IPCP Open
  Last input 3d18h, output 3d18h
  Last clearing of "show interface" counters never
  Last state change 3d18h
  Queueing strategy: fifo
  Output queue 0/100, 0 drops; input queue 0/0, 0 drops
      27240 packets input, 2395832 bytes, 0 no buffer
      Received 0 broadcasts, 0 multicasts, 0 runts, 1 giants
      0 input errors, 2 CRC, 21 overruns, 36 aborts, 0 parity
      27241 packets output, 2286960 bytes,  44 underruns
      0 output errors, 0 collisions, 0 interface resets
      14 carrier transitions
```

Figure 14-4 show interface pos Command Output

Table 14-9 describes the **show interface pos** command output fields.

Table 14-9 show interface pos Output Fields and Descriptions

Field	Description
POS1/0 is up/down	Indicates whether the physical link is currently up or down.
line protocol is up/down	Indicates whether the PPP link or protocol is currently up or down.
Hardware is	Hardware type.
Internet address is	IP address and subnet mask.
MTU	Maximum transmission unit of the interface.
BW	Bandwidth of the interface in KBIT/S per second.
Encapsulation	Encapsulation method assigned to interface which in this instance is PPP with the CRC sequence is set to 16 bytes.
loopback	Indicates whether loopbacks are set.
keepalive	Indicates whether keepalives are set.
scramble	Indicates whether payload scrambling is enabled or disabled.

Table 14-9 show interface pos Output Fields and Descriptions

Field	Description
LCP	Indicates if the Link Control Protocol is open or closed. LCP is used to negotiate PPP configuration parameters.
IPCP	Indicates if the Internet Protocol Control Protocol is open or closed. IPCP is used for transporting IP traffic over a PPP connection.
Last input	Number of hours, minutes, and seconds since the last packet was successfully received by an interface. Indicates a dead interface failure.
(Last) output	Number of hours, minutes, and seconds since the last packet was successfully transmitted by an interface.
Last clearing of “show interface” counters	Describes the last time that the counters were cleared for the show interface command statistics.
Last state change	Number of hours, minutes, and seconds since the last state change of the POS interface.
Queueing strategy	First-in, first-out queuing strategy (other queueing strategies are priority-list, custom-list, and weighted fair).
Output queue, drops input queue, drops	Number of packets in output and input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped because a queue was full.
packets input	Total number of error-free packets received by the system.
bytes (input)	Total number of bytes, including data and MAC encapsulation, in the error-free packets received by the system.
no buffer	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines can cause no input buffer events.
Received <*n*> broadcasts	Where *n* is the total number of broadcast or multicast packets received by the interface.
multicasts	Indicates the number of multicast packets received by the interface.
runts	Number of packets that are discarded because they are smaller than the medium's minimum packet size.
giants	Number of packets discarded because they exceed the medium's maximum packet size.

Table 14-9 show interface pos Output Fields and Descriptions

Field	Description
input errors	Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored, and abort counts. Other input-related errors can also increment the count, so that this sum might not balance with the other counts.
CRC	Cyclic redundancy checksum generated by the originating LAN station or far-end device does not match the checksum calculated from the data received.
overruns	Number of times the serial receiver hardware was unable to transfer received data to a hardware buffer because the input rate exceeded the receiver ability to handle the data.
aborts	Illegal sequence of one bits on the interface.
parity	Report of the parity errors on the interface.
packets output	Total number of messages transmitted by the system.
bytes (output)	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Number of times that the far-end transmitter has been running faster than the near-end router's receiver can handle.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the interface being examined.
collisions	Number of collisions on the interface.
interface resets	Number of times an interface has been completely reset.
carrier transitions	Number of times the carrier detect signal of the interface has changed state.

Displaying Physical SONET Link and Alarm Information

SONET alarm hierarchy rules mandate that only the most severe alarm of an alarm group is reported. Reported alarms are logged to the console.

To display information about the POS module hardware, SONET alarms and BER thresholds, use the **show controllers pos** command in Privileged EXEC mode as shown below:

Note: The Active Defect field in the **show controllers pos** command output shows all alarms currently present on the interface.

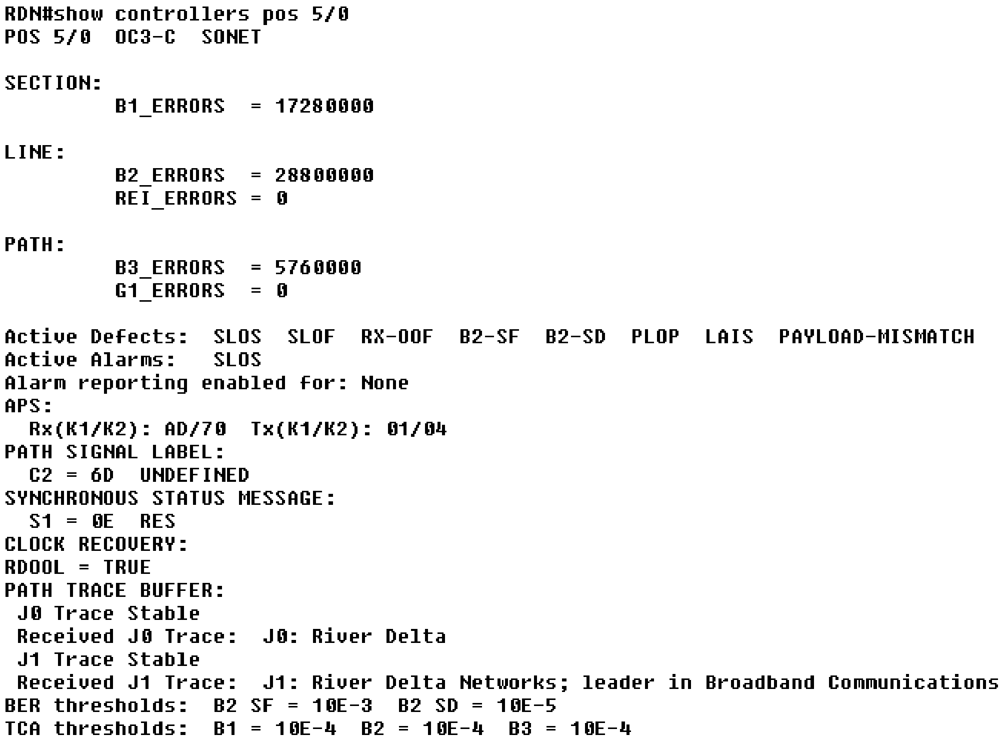
```
MOT#show controllers pos <slot>/<interface>
```

where:

slot is the POS module slot on the BSR 64000 chassis.

interface is the line POS interface on the POS module.

Figure 14-5 displays the **show controllers pos** command output.



```
RDN#show controllers pos 5/0
POS 5/0  OC3-C  SONET

SECTION:
         B1_ERRORS  = 17280000

LINE:
         B2_ERRORS  = 28800000
         REI_ERRORS = 0

PATH:
         B3_ERRORS  = 5760000
         G1_ERRORS  = 0

Active Defects:  SLOS  SLOF  RX-OOF  B2-SF  B2-SD  PLOP  LAIS  PAYLOAD-MISMATCH
Active Alarms:   SLOS
Alarm reporting enabled for: None
APS:
  Rx(K1/K2): AD/70  Tx(K1/K2): 01/04
PATH SIGNAL LABEL:
  C2 = 6D  UNDEFINED
SYNCHRONOUS STATUS MESSAGE:
  S1 = 0E  RES
CLOCK RECOVERY:
RDOOL = TRUE
PATH TRACE BUFFER:
 J0 Trace Stable
 Received J0 Trace:  J0: River Delta
 J1 Trace Stable
 Received J1 Trace:  J1: River Delta Networks; leader in Broadband Communications
BER thresholds:  B2 SF = 10E-3  B2 SD = 10E-5
TCA thresholds:  B1 = 10E-4  B2 = 10E-4  B3 = 10E-4
```

Figure 14-5 show controllers pos Command Output

Table 14-10 defines the SONET Alarms and BER threshold information that is reported on the POS interface console:

Table 14-10 SONET Alarms and Identifications

Alarm	Identification
POS 5/0	POS slot and interface.
SECTION	Errors that occur in the SONET Section. A *section* may be between Customer Premises Equipment (CPE) and SONET Service Provider Equipment (SPE).
SLOS	Section los of signal errors. SLOS is detected when an all-zeros pattern on the incoming SONET signal lasts 19(+-3) microseconds or longer.

Table 14-10 SONET Alarms and Identifications

Alarm	Identification
SLOF	Section loss of frame errors. SLOF is detected when a severely errored frame (SEF) defect on the incoming SONET signal persists for 3 milliseconds.
b1-tca	B1 BER TCA (crossing threshold) For B1, the bit interleaved parity error report is calculated by comparing the BIP-8 code with the BIP-8 code extracted from the B1 byte of the next frame. Differences indicate that section level bit errors have occurred.
LINE	Errors that occur in the SONET Line. A *line* may be between a SONET SPE, regenerators, and another SONET SPE.
L-AIS	Alarm indication signal. Line alarm indication signal is sent by the section terminating equipment (STE) to alert the downstream LTE that a LOS or LOF defect has been detected on the incoming SONET section.
L-RDI	Remote defect indication. Line remote defect indication is reported by the downstream LTE when it detects LOF, LOS, or AIS.
L-REI	Line Remote Error Indicator. Conveys a B2 error count detected by the LTE.
sf-ber	The line is considered in failure.
sd-ber	B2 errors have exceeded the threshold and the line is considered degraded.
b2-tca	B2 BER TCA (crossing threshold) For B2, the bit interleaved parity error report is calculated by comparing the BIP-8/24 code with the BIP-8 code extracted from the B2 byte of the next frame. Differences indicate that line level bit errors have occurred.
PATH	Errors that occur in the SONET Path. A *path* may be between a CPE, SONET SPE, regenerators, and another SONET SPE and CPE.
PLOP	Path loss of pointer is reported as a result of an invalid pointer (H1, H2) or an excess number of new data flag (NDF) enabled indications.
P-AIS	Path alarm indication signal errors. PAIS is sent by line terminating equipment (LTE) to alert the downstream path terminating equipment (PTE) that it has detected a defect on its incoming line signal.

Table 14-10 SONET Alarms and Identifications

Alarm	Identification
P-RDI	Path remote defect indication errors. Path remote defect indication is reported by the downstream PTE when it detects a defect on the incoming signal.
b3-tca	B3 BER TCA (crossing threshold) For B3, the bit interleaved parity error report is calculated by comparing the BIP-8 code with the BIP-8 code extracted from the B3 byte of the next frame. Differences indicate that path level bit errors have occurred.
Active Defects	All currently active SONET defects.
Active Alarms	Current Alarms as enforced by SONET Alarm Hierarchy.
Alarm reporting enabled for	Alarms that were enabled through the use of the **pos report** command in POS interface configuration mode.
Path Signal Label	The value extracted from the SONET path signal label byte (C2).
Synchronous Status Message	Bits 5 through 8 of the S1 Byte which describe the synchronization status of the Near End device.
CLOCK RECOVERY	Displays that the recovered clock is from the received SONET signal.
RDOOL	Receive data out of lock errors. Describes the status of the clock recovery. If a Receive Data Out Of Lock alarm has been detected, this indicates that the clock recovery phased lock loop is unable to lock to the receive stream.
PATH TRACE BUFFER	SONET path trace buffer is used to communicate information regarding the remote hostname, interface name/number and IP address or other user designated parameters. J0 indicates a section and J1indicates a path message.
TCA thresholds	List of TCAs you configured with the **pos threshold** command in Interface Configuration mode.

Index

A

ABR, 11-1, 11-6, 11-9

access control
- using community name, 4-16

access list
- AS-path, 7-6
- IP, 7-6
- name, 9-37
- number, 9-37

access policy
- creating, 4-18

activating
- upstream Forward Error Correction, 6-32

adding
- ARP cache entry, 5-14
- permanent ARP entry, 5-14

address
- IP, 9-32

address range
- specifying, 11-9, 11-10

Address Resolution Protocol
- configuring, 5-14

Address Resolution Protocol (ARP), 6-11

addresses
- Ethernet, 5-14
- Internet, 5-14
- MAC, 5-14

addressing
- multicast, 9-32

advertising
- network, 11-9
- networks, 7-12
- route, 7-1, 11-10

AK
- BPI, 6-49
- grace-time, 6-50
- permanent, 6-50
- temporary, 6-50

application hosting, 1-2

applying
- upstream modulation profile, 6-25

area border router
- area border, 11-1

ARP
- configuring, 5-14
- see host authorization, 6-6

ARP cache
- adding entry, 5-14

ARP entry
- adding permanent, 5-14

ARP table information
- displaying, 5-29

AS
- and stub area, 11-6
- boundary router in, 9-32
- topology, 11-1

AS path
- modifying, 7-4

AS-path
- access list
  - configuring, 7-7

assigning
- cost

        stub area default summary route, 11-7
    OSPF
        area ID, 11-2

associating
    group to MIB view, 4-19, 4-21

authenticaton
    OSPF MD5, 11-21

Authorization Key
    setting, 6-49

autonomous system, 9-32

B

backbone
    network, 1-1
    OSPF, 11-11

backspace key, 2-15

Baseline Privacy
    configuring, 6-49

basic configuration tasks, 3-1

basic tasks
    downstream configuration, 6-13

BGP AS-path access list
    matching, 7-2

BGP community attribute
    setting, 7-4

BGP community list
    displaying routes permitted by, 12-72

BGP origin
    setting, 7-3, 7-4

blocking
    OSPF LSA flooding, 11-22

Bootstrap Routers (BSRs), 8-5

BPI
    configuring, 6-49
    parameters, 6-50
    settings
        viewing, 6-50
    statistics
        viewing, 6-50

broadcast
    message, 9-32

broadcast address
    defining, 5-17
    scheme, 5-17

broadcast packets
    directed, 5-16
    flooded, 5-16
    limited, 5-16

broadcast storms
    avoiding, 5-17

BSR
    configuring, 3-1

buffer
    clearing, 3-16

burst profile, 6-129

C

cable bundling, 6-36

cable headend path loss, 6-22

cable plant, 1-1

Cable Spectrum Group mode, 6-63

Candidate BSRs, 8-5

case
    controlling, 2-16

center frequency
    CM, 6-20

changing
    interface-specific OSPF parameter, 11-19
    OSPF
        distance value, 11-17

channel width
    upstream parameters, 6-20

characters
    transposing, 2-16

chassis information, 3-53

clear ip route, 5-28

clearing
    buffer, 3-16
    dynamic route, 5-20
    routing table entry, 5-20

CLI, 1-2, 2-1, 9-31, 13-1, 14-1

- CM authentication information
  - obtaining, 6-47
- CM information
  - displaying, 6-118
- CMTS interface
  - BPI settings
    - viewing, 6-50
  - BPI statistics
    - viewing, 6-50
- command history
  - buffer size
    - setting, 2-11
  - disabling, 2-12
  - using, 2-11
- command line interface, 1-2, 2-1, 9-31, 13-1, 14-1
- command lines
  - handling, 2-14
- commands
  - area area-id virtual-link router-id, 11-12, 11-13, 11-14
  - area authentication, 11-6
  - area authentication message-digest, 11-6
  - area default-cost, 11-7
  - area nssa, 11-8
  - area-id range address mask, 11-9, 11-10
  - arp, 5-14
  - boot system, 3-42
  - cable downstream carrier-only, 6-19
  - cable downstream frequency, 6-13
  - cable downstream interleave-depth, 6-16
  - cable downstream power-level, 6-17
  - cable downstream rate-limit, 6-14
  - cable downstream shutdown, 6-17
  - cable insertion-interval, 6-44
  - cable modem max-hosts, 6-48
  - cable modem qos dsc, 6-55
  - cable modem qos dsd, 6-57
  - cable modulation-profile, 6-42
  - cable privacy cm-auth reset, 6-53
  - cable privacy cm-tek grace-time, 6-51
  - cable privacy cm-tek life-time, 6-52
  - cable sync-interval, 6-45
  - cable ucd-interval, 6-35
  - cable upstream data-backoff, 6-29
  - cable upstream force-fragmentation, 6-34
  - cable upstream frequency, 6-21
  - cable upstream minislot-size, 6-27
  - cable upstream modulation-profile, 6-25
  - cable upstream range-backoff, 6-28
  - cable upstream range-power-override-continue, 6-29
  - cable upstream rate-limit, 6-25
  - cable upstream scrambler, 6-33
  - clear ip routes, 5-20
  - clear log, 3-16
  - clock set, 3-11
  - clock timezone, 3-11
  - default-information originate, 11-15
  - distance ospf, 11-17
  - hostname, 3-6
  - interface cable, 6-50
  - interface loopback, 11-23
  - ip address, 11-23
  - ip as-path access-list, 7-7
  - ip broadcast-address, 5-17
  - ip community-list, 7-10
  - ip directed-broadcast, 5-17
  - ip ftp password, 3-29
  - ip ftp username, 3-29
  - ip irdp, 5-21
  - ip mask-reply, 5-23
  - ip mtu, 5-17
  - ip ospf authentication-key, 11-19
  - ip ospf cost, 11-19
  - ip ospf dead-interval, 11-19
  - ip ospf hello-interval, 11-19, 11-20
  - ip ospf retransmit-interval, 11-20
  - ip ospf transmit-delay, 11-20, 11-21
  - ip redirects, 5-23
  - ip rip receive version, 9-34
  - ip rip send version, 9-34
  - ip route, 5-18, 5-19
  - ip split-horizon, 9-35
  - ip unreachables, 5-23
  - logging console, 3-14
  - logging trap, 3-12
  - match AS-path, 7-7
  - match as-path, 7-2
  - match community, 7-2
  - match ip address next-hop, 7-3
  - match metric, 7-3

neighbor distribute-list, 7-6
neighbor filter-list, 7-7
network, 9-33
network network-number, 9-33
offset-list, 9-37
ospf database-filter all out, 11-22
output-delay, 9-38
ping, 5-23
recalling, 2-12
redistribute, 7-12
route map, 7-1
route-map, 7-1
router ospf process-id, 11-2
router rip, 9-33
set as-path prepend, 7-4
set community, 7-4
set ip next-hop, 7-4
set local-preference, 7-5
set metric, 7-5
set metric-type, 7-5
set origin, 7-3, 7-4
set-community, 7-9
show, 6-111, 7-18, 14-21
show boot, 3-42
show cable modem, 6-118
show cable modem hosts, 6-49, 6-124
show cable modem svc-flow-id, 6-56
show cable modulation-profile, 6-129
show cable privacy, 6-130
show cable privacy auth, 6-53, 6-130
show cable privacy cm-auth, 6-53
show cable privacy tek, 6-52
show cable qos svc-flow classifier, 6-58
show cable qos svc-flow dynamic stat, 6-55, 6-56, 6-57
show cable qos svc-flow log, 6-57
show cable qos svc-flow phs, 6-60
show cable qos svc-flow statistics, 6-14, 6-26, 6-56, 6-59
show cable qos svc-flow summary, 6-59
show interface cable, 6-18, 6-26, 6-34
show ip, 5-29
show ip access-list, 7-18, 7-19
show ip arp, 5-29
show ip as-path access-list, 7-18, 7-19
show ip bgp, 7-18, 7-19
show ip bgp community, 7-18, 7-19
show ip bgp dampened-paths, 7-18, 7-19
show ip community-list, 7-18, 7-19, 12-72
show ip interface, 5-30
show ip ospf, 11-15
show ip ospf virtual-links, 11-15
show ip prefix-list, 7-18, 7-19
show ip route static, 5-30
show ip routes, 5-30
show ip routes show route-map, 7-18, 7-19
show ip traffic, 5-30
show log, 3-15, 3-16
show route-map, 7-21
show running-config, 6-3, 6-14, 6-26, 6-45, 6-47
show snmp, 4-27
show snmp access, 4-27
show snmp community, 4-27
show snmp contact, 4-27
show snmp context, 4-27
show snmp engineID, 4-27
show snmp group, 4-27
show snmp host, 4-27
show snmp location, 4-27
show snmp packetsize, 4-27
show snmp port, 4-27
show snmp sysname, 4-27
show snmp users, 4-27
show snmp view, 4-27
SNMP, 4-1
snmp-server, 4-1, 4-2, 4-16
snmp-server access, 4-2, 4-21
snmp-server chassis-id, 4-2
snmp-server community, 4-2, 4-17, 4-25
snmp-server contact, 4-2, 4-14
snmp-server context, 4-2, 4-21
snmp-server enable, 4-3, 4-8
snmp-server enable traps, 4-23
snmp-server engine, 4-3, 4-14
snmp-server group, 4-3, 4-11, 4-19
snmp-server host, 4-3, 4-24, 4-25
snmp-server host traps, 4-25
snmp-server host version, 4-25
snmp-server location, 4-3, 4-14
snmp-server packetsize, 4-6, 4-22
snmp-server port, 4-6
snmp-server sysname, 4-7

- snmp-server trap, 4-7
- snmp-server trap rate-limit, 4-26
- snmp-server user, 4-7, 4-19
- snmp-server view, 4-7, 4-12, 4-20
- summary address prefix mask, 11-8
- summary-address, 11-10
- upstream channel width, 6-28
- username, 3-6
- version, 9-34

community name
- defining, 4-16

completing
- partial command name, 2-14

configuration tasks
- basic, 3-1

configured route maps
- displaying, 7-19, 7-20, 7-21

configuring
- AS-path
  - access list, 7-7
- Baseline Privacy, 6-49
- BSR, 3-1
- console logging, 3-14
- FTP access, 3-17, 3-29
- group model, 4-19
- MIB view, 4-20
- modulation interface profile, 6-41
- network identity, 3-1
- Open Shortest Path First, 13-1
- OSPF, 11-1, 11-5, 13-1
- packet size, 4-22
- QoS, 6-54
- Quality of Service, 6-54
- router
  - OSPF-based, 11-1
- SNMP
  - context, 4-21
- SNMP access levels, 4-16
- SNMP server identification, 4-13
- SNMP traps, 4-23
- software
  - RIP version 1, 9-34
  - to advertise single route, 11-10
- stub area, 11-6
- system contact information, 4-14
- system location information, 4-14
- system log parameters, 3-11
- trap destination, 4-23
- trap source, 4-23
- USM security, 4-18
- VACM security, 4-18

console logging
- configuring, 3-14

context-sensitive help, 2-11

controlling
- case, 2-16
- summarization and filtering, 11-8
- upstream input power level, 6-22

CPE, 14-27

creating
- access policy, 4-18
- OSPF
  - routing process, 11-2
- user models, 4-18

Ctrl B, 2-13

Ctrl-A, 2-13, 2-14

Ctrl-B, 2-14

Ctrl-D, 2-15

Ctrl-E, 2-13

Ctrl-F, 2-13

Ctrl-K, 2-15

Ctrl-L, 2-14

Ctrl-N, 2-12

Ctrl-P, 2-12

Ctrl-R, 2-15

Ctrl-T, 2-16

Ctrl-U, 2-15

Ctrl-W, 2-15

D

damping criteria
- alternate half-life period, 7-13, 7-14
- half-life period, 7-13, 7-14
- maximum supression time, 7-13, 7-14
- reuse limit, 7-13, 7-14

suppress limit, 7-13, 7-14

data backoff, 6-20

Data Encryption Standard, 6-49

Data Over Cable Service Interface Specification
standard, 1-1

data packet, 9-31

datagram
fragmentation, 5-2
packet format, 5-2
reassembly, 5-2
routing to remote host, 5-2

decreasing
OSPF
link state database size, 11-10

default power-level
range, 6-22
returning, 6-17

default route
setting, 5-19

default router
definition, 5-18

default synchronization message interval
resettingl, 6-45, 6-46

defining
broadcast address, 5-17
community name, 4-16
MIB view, 4-19
route map, 7-1
route maps, 7-1

delete key, 2-15

deleted service flow log
displaying, 6-57

deleting
entries, 2-15
service flow, 6-56

deny
conditions, 7-6

deny conditions, 7-7

DES, 6-49

destination IP address
specifying, 4-24

determing
number of dropped packets, 6-14

determining
number of dropped packets, 6-26

DHCP relay information option, 6-2

diff, 6-42

differential encoding, 6-42

directed broadcast packets, 5-16

disabling
command history, 2-12
downstream center frequency setting, 6-13
downstream port, 6-17
power adjustment, 6-29
upstream FEC, 6-32
upstream scrambler, 6-33

displaying
ARP table information, 5-29
configured route maps, 7-19, 7-20, 7-21
deleted service flow log, 6-57
DSA message
statistics, 6-55
DSC message
statistics, 6-56
DSD message statistics, 6-57
IP interface state, 5-30
modulation profile group information, 6-129
modulation profile groups, 6-43
routes permitted by BGP community list, 12-72
routing table status, 5-30
service flow
packet classifiers, 6-58
QoS information, 6-59
service flow SID information, 6-59
service flow statistics, 6-56
SFID, 6-56
static routes
status, 5-30
TEK
life-time value, 6-130
TEK grace-time value, 6-130
virtual link information, 11-15

displaying CM information, 6-118

Distance Vector Multicast Routing Protocol
(DVMRP), 8-8 to 8-16

- parent-child links, 8-11
- summary address, 8-15

distance-vector routing, 9-31

distribution hub, 1-1

DOCSIS, 1-1
- standard, 1-1

down arrow key, 2-12

downstream data
- encrypting, 6-49

downstream data transmission rate-limit
- enabling, 6-14

downstream port
- disabling, 6-17

downstream power level
- setting, 6-17

downstream rate-limiting status
- verifying, 6-14

downstream tasks
- basic, 6-13

DSA, 6-54

DSA message
- statistics
  - displaying, 6-55

DSC, 6-54

DSC message
- statistics
  - displaying, 6-56

DSD, 6-54

DSD message statistics
- displaying, 6-57

Dynamic Service Addition, 6-54

Dynamic Service Change, 6-54

Dynamic Service Deletion, 6-54

E

editing
- features, 2-13

EGP, 9-32

enabling
- downstream data transmission rate-limit, 6-14
- ICMP netmask reply, 5-23
- IP redirects, 5-23
- logging process, 3-11
- OSPF, 11-2
  - MD5 authentication, 11-21
- power adjustment, 6-29
- rate-limiting function, 6-14
- SNMP, 4-8
- traps, 4-23
- upstream data transmission rate-limit, 6-25
- upstream FEC, 6-32

encrypting
- downstream data, 6-49
- upstream data, 6-49

encryption, 6-49

entries
- deleting, 2-15

Esc B, 2-14

Esc D, 2-15

Esc F, 2-14

Ethernet
- auto-negotiation, 5-6

Ethernet address, 5-14

Exterior Gateway Protocol, 9-32

F

features
- editing, 2-13

FEC, 6-32
- last code word, 6-43
- upstream parameter, 6-20

field
- next hop, 9-32
- subnet mask, 9-32

File Transfer Protocol
- support for, 1-2

filtering
- routes
  - based on a community list, 7-10

flap lists, 6-95
- troubleshooting, 6-98

flooded broadcast packets, 5-16

forcing
- packet fragmentation, 6-34

Forward Error Correction
- CMTS use of, 6-32
- upstream parameter, 6-20

forwarding
- centralized, 1-1

fragmentation, 5-17

frequency, 6-20

frequency band, 6-63

frequency range
- parameter, 6-12

FTP
- access
  - configuring, 3-17, 3-29
- support for, 1-2

G

generating
- ICMP unreachable messages, 5-23

Global Configuration mode, 2-4, 2-5

grace-time, 6-49

group
- associating to MIB, 4-19, 4-21

group model
- configuring, 4-19

group models
- handling, 4-18

H

handling
- command lines, 2-14
- group models, 4-18

hello interval, 10-11

hello packet, 11-20, 11-22

help
- context-sensitive, 2-11
- obtaining, 2-10

history buffer, 2-12

hop count, 9-32

host
- route, 9-32

host authorization
- dynamic, 6-6
- static, 6-6
- verifying, 6-6

Host-to-Host Transport layer, 5-2

I

ICMP, 5-20

ICMP echo request packets
- sending, 5-23

ICMP netmask reply
- enabling, 5-23

ICMP unreachable messages
- generating, 5-23

IGP, 9-31

incoming metrics, 9-37

individual CM grace-time TEK
- setting, 6-51

initial ranging backoff, 6-20

initiating
- DSD
  - of SFID, 6-57

input power level, 6-20

input power-level range, 6-22

insertion interval
- parameter, 6-12
- setting, 6-44

interactive, multiplayer gaming, 1-1

interface
- type, 9-37

Interface Configuration mode, 2-6

interface number, 9-37

Interior Gateway Protocol, 9-31

interleave depth
- parameter, 6-12
- setting, 6-16

Intermediate System, 10-1

Internet address, 5-14

Internet Control Message Protocol, 5-20

Internet Group Management Protocol (IGMP), 8-16 to 8-20
- host-query messages, 8-17, 8-18
- querier timeout interval, 8-19
- version, 8-18

Internet Service Provider, 11-8

interpacket
- delay
  - RIP update, 9-38

Interval Usage Code, 6-42

IP
- address, 9-32, 11-23
- monitoring, 5-29
- multicast, 11-1
- packet, 5-2
- subnetting, 11-1
- tagging, 11-1

IP interface state
- displaying, 5-30

IP multicast group access, 8-17

IP multicast routing, 8-1

IP summary address, 10-9

IP telephony, 1-1

ip vrrp address, 13-3

IS-IS
- administrative distance, 10-8
- interface password, 10-15
- point-to-point interface, 10-14

ISP, 11-8

iuc, 6-42

K

KEK
- setting, 6-49

KEK BPI information
- displaying, 6-130

Key Encryption Key, 6-49

L

learning
- route, 7-1

left arrow key, 2-13, 2-15

life-time, 6-49

limited broadcast packets, 5-16

line, 14-28

link state
- update packet, 11-20

link state advertisement
- retransmission, 11-20

local preference value
- setting, 7-5

logging process
- enabling, 3-11

logging rates
- restricting, 3-16

loopback, 11-23

LSA
- age, 11-22
- aging process, 11-22
- refreshing, 11-22
- retransmission, 11-20
- transmission interval, 11-21
- Type 5
  - flooding, 11-8
  - translating from Type 7, 11-8
- Type 7
  - importing, 11-8
  - translating into Type 5, 11-8

M

MAC address, 5-14

MAC management messages, 6-54

match tag, 7-5

matching
- BGP AS-path access list, 7-2
- next-hop IP address, 7-3
- routing metric value, 7-3

maximum burst length, 6-42

maximum CPE host number
- specifying, 6-48
- verifying, 6-49, 6-124

message
- broadcast, 9-32

metric
- setting, 7-5

metric type
- setting, 7-5

MIB, 1-2

MIB view
- configuring, 4-20
- defining, 4-19

minislot size, 6-20

modifying
- AS path, 7-4

modulation
- interface profile
  - configuring, 6-41

modulation profile
- content, 6-129
- group
  - viewing, 6-43
- group information
  - displaying, 6-129
- parameter, 6-42
- upstream parameter, 6-20

modulation rate
- parameter, 6-12

monitoring
- IP, 5-29
- routing policy, 7-18
- SNMP, 4-27

MPEG framing format
- parameter, 6-12

multicast
- addressing, 9-32

Multicast Backbone (MBONE), 8-12

multicast group, 8-1

multiple-packet RIP update, 9-38

N

navigating
- command line, 2-13

network
- advertising, 7-12
- local link
  - advertisement, 11-1
- number, 9-33
- OSPF
  - designated router for, 11-20

Network Access layer, 5-2

network identify
- configuring, 3-1

next hop
- field, 9-32

next-hop IP address
- matching, 7-3

nonbackbone area, 11-11

Not So Stubby Area, 11-8

NSSA, 11-8

NSSA ABR, 11-8

number of dropped packets
- determing, 6-14
- determining, 6-26

O

obtaining
- CM authentication information, 6-47
- help, 2-10

offset, 9-37

offset list, 9-37

on-demand
- audio, 1-1
- music, 1-1
- video, 1-1

Open Shortest Path First
- configuring, 13-1

optional packet count
- setting, 5-23

OSPF

  - area ID
    - assigning, 11-2
    - defining, 11-1
  - area parameter
    - configuring, 11-5
  - configuring, 11-1, 13-1
  - designated network router, 11-20
  - distance value
    - changing, 11-17
  - enabling, 11-2
  - interface-specific parameter
    - changing, 11-19
  - link state database
    - decreasing, 11-10
  - LSA flooding
    - blocking, 11-22
  - NSSA area parameter
    - specifying, 11-8
  - password, 11-20
  - route
    - redistributing from other protocol, 11-10
  - routing process
    - creating, 11-2
  - stub area
    - support, 11-6

- OSPF area
  - defining, 11-1
- OSPF router
  - router ID, 11-15
- outgoing interfaces (OIs), 8-25
- outgoing metrics, 9-37
- output power level, 6-22

P

- packet
  - authentication, 11-1
  - data, 9-31
  - RIP Version 1, 9-34
  - size
    - configuring, 4-22
  - specifying
    - cost of sending, 11-19
  - Version 2, 9-34
- packet cable standards, 1-1
- packet fragmentation
  - forcing, 6-34
- parameters
  - BPI, 6-50
- partial command name
  - completing, 2-14
- password, 2-5, 2-13
  - OSPF, 11-20
- path, 14-28
- payload header suppression entries, 6-60
- permanent AK, 6-50
- permanent TEK, 6-50
- permit
  - conditions, 7-6
- permit conditions, 7-7
- PHS entries
  - displaying, 6-60
- physical layer profile characteristics, 6-20
- ping docsis, 6-108
- Policy-Based Routing
  - configuring for, 7-15
  - defining match criteria, 7-15
  - enabling
    - interface, 7-18
    - local, 7-18
  - set commands, 7-16
- port minislot size, 6-20
- power adjustment
  - disabling, 6-29
  - enabling, 6-29
- power level, 6-20
  - parameter, 6-12
- preamble length, 6-43
- Privileged EXEC mode, 2-4, 2-5
- protocol
  - Open Shortest Path First, 13-1
  - OSPF, 13-1
- Protocol-Independent Multicast (PIM), 8-2 to 8-8
  - sparse mode, 8-3

Q

QoS
- configuring, 6-54
- functions, 1-1

Quality of Service
- configuring, 6-54
- functions, 1-1

R

radio frequency channel width, 6-20

range backoff, 6-20

rate-limiting function
- enabling, 6-14

recalling
- commands, 2-12

redirects
- enabling, 5-23

redistributed routes
- specifying address, 11-10
- specifying mask, 11-10

redistributing
- lists, 7-12
- route
  - into OSPF, 11-10
- routes, 7-12

reducing
- routing table size, 5-18

re-enabling
- upstream FEC, 6-32
- upstream Forward Error Correction, 6-32
- upstream scrambler, 6-33

refreshing
- LSA, 11-22

regional fiber network, 1-1

remote hosts, 5-2

removing
- modulation profile, 6-43

rendezvous point (RP), 8-3 to 8-4
- configuration of, 8-7

resetting
- default synchronization message interval, 6-45, 6-46
- features, 2-8
- upstream channel width default, 6-28
- upstream minislot size, 6-28
- upstream range-backoff default, 6-29

restricting
- logging rates, 3-16
- trap rates, 4-26

returning
- default power-level, 6-17

Reverse Path Forwarding (RPF), 8-2

RF output upstream frequency, 6-21
- setting, 6-21

right arrow key, 2-13

RIP
- and distance vector routing, 9-31
- configuring, 9-31
- packet, 9-34
- packets, 9-34
- update
  - interpacket delay, 9-38
  - multiple-packet, 9-38

RiverClear spectrum management, 2-7

route
- advertising, 7-1
- configuring, 5-18
- default external, 11-6
- host, 9-32
- learning, 7-1
- loop-free, 11-1
- next-hop attribute
  - setting, 7-4
- optimum, 11-1
- redistributed, 11-10
- setting specific, 5-18
- specifying, 5-18
- summarization, 11-9
- supernet, 9-32
- tag, 9-32
- tag field, 9-32

route map
- content, 7-1
- defining, 7-1

route maps
- defining, 7-1

router
- ID, 11-23
- OSPF-based
  - configuring, 11-1
- update, 9-31

router discovery packets
- tracing, 5-20

Router Discovery Protocol, 5-20

routers
- internal, 11-1

routes
- redistributing, 7-12

routing
- centralized, 1-1
- datagram
  - to remote host, 5-2
- distance-vector, 9-31

Routing Information Protocol, 9-31

routing metric value
- matching, 7-3

routing policy, 7-1
- monitoring, 7-18

routing table
- information, 5-2
- status
  - displaying, 5-30

routing table entry
- clearing, 5-20

S

scrolling
- line, 2-15
- screen, 2-15

section, 14-27

security access
- system, 3-6

sending
- ICMP echo request packets, 5-23

service flow
- changing, 6-55
- deleting, 6-56
- packet classifiers
  - displaying, 6-58
- SID information
  - displaying, 6-59
- statistics
  - displaying, 6-59

service flow QoS information
- displaying, 6-59

service flow statistics
- displaying, 6-56

service provider equipment
- see SPE

set statements, 7-4

setting
- Authorization Key, 6-49
- BGP community attribute, 7-4
- BGP origin, 7-3, 7-4
- command history buffer size, 2-11
- default route, 5-18, 5-19
- downstream modulation rate, 6-13
- downstream power level, 6-17
- individual CM grace-time TEK, 6-51
- insertion interval, 6-44
- interleave depth, 6-16
- KEK, 6-49
- local preference value, 7-5
- metric, 7-5
- metric type, 7-5
- optional packet count, 5-23
- RF output upstream frequency, 6-21
- route
  - next-hop attribute, 7-4
- specific route, 5-18
- synchronization message interval, 6-45
- TEK, 6-49
- UCD message transmission interval, 6-35
- upstream burst parameters, 6-24, 6-41
- upstream channel width, 6-28
- upstream data-backoff, 6-29
- upstream frequency, 6-21
- upstream minislot size, 6-27
- upstream range-backoff, 6-28

SFID

displaying, 6-55, 6-56

SFID DSD
initiating, 6-57

Shortest Path Tree (SPT), 8-4

show history command, 2-12

show ip ospf memory, 11-31

show ip vrrp ethernet num, 13-12

Simple Network Management Protocol
support for, 1-2

SLAs, 1-1

SNMP
access levels
configuring, 4-16
commands, 4-1
context
configuring, 4-21
enabling, 4-8
monitoring, 4-27
server identification
configuring, 4-13
support for, 1-2
traps
configuring, 4-23

SONET
b1-tca, 14-28
b2-tca, 14-28
b3-tca, 14-29

SPE, 14-27

specific trap types
specifying, 4-25

specific trap versions
specifying, 4-25

specification
Data Over Cable Service Interface Specification, 1-1
DOCSIS, 1-1

specifying
address, 11-10
address range, 11-9, 11-10
cost of sending packet, 11-19
destination IP address, 4-24
mask, 11-10
maximum CPE host number, 6-48
OSPF
NSSA area parameter, 11-8
specific trap types, 4-25
specific trap versions, 4-25

spectrum data collection, 6-71

spectrum groups, 6-61

spectrum management, 6-60

SPF routing, 10-1

standards
packet cable, 1-1

static routes
status
displaying, 5-30

stub area
configuring, 11-6
default summary route cost
assigning, 11-7

subnet mask
advertisement, 9-32
field, 9-32
rules, 9-32

summary route
advertisement, 11-9

supernet route, 9-32

SYNC interval, 6-12

SYNC message transmission
intervals, 6-12

synchronization message interval
setting, 6-45
verifying, 6-45

system
contact information
configuring, 4-14
location information
configuring, 4-14
log parameters
configuring, 3-11
login
account, 3-6
security access, 3-6

system information, 3-53

system logging, 3-11

T

tab key, 2-14

tag
- route, 9-32

TCP/IP, 5-1

TEK
- grace-time, 6-50
- grace-time value
  - displaying, 6-130
- life-time, 6-50
- life-time value
  - displaying, 6-130
- permanent, 6-50
- setting, 6-49

temporary AK, 6-50

temporary traffic key, 6-50

tiered data services, 1-1

time-to-live (TTL), 8-18

tracing
- router discovery packets, 5-20

traffic key
- temporary, 6-50

Transmission Control Protocol/Internet Protocol, 5-1

transposing
- characters, 2-16

trap destination
- configuring, 4-23

trap rates
- restricting, 4-26

trap source
- configuring, 4-23

traps
- enabling, 4-23

U

UCD message transmission interval
- setting, 6-35

UDP, 9-31

up arrow key, 2-12

update packet
- link state, 11-20

upstream, 6-31

upstream burst parameters
- setting, 6-24, 6-41

upstream cable interface port, 6-26, 6-34

upstream channel, 6-61

upstream channel bandwidth, 6-22

upstream channel width
- setting, 6-28

upstream channel width default
- resetting, 6-28

upstream data
- encrypting, 6-49

upstream data transmission rate-limit
- enabling, 6-25

upstream data-backoff
- setting, 6-29

upstream FEC
- activating, 6-32
- disabling, 6-32
- enabling, 6-32
- re-enabling, 6-32

upstream Forward Error Correction
- activating, 6-32
- re-enabling, 6-32

upstream input power level
- controlling, 6-22

upstream minislot size
- resetting, 6-28
- setting, 6-27

upstream modulation profile
- applying, 6-25

upstream propagation delay, 6-31

upstream range-backoff
- setting, 6-28

upstream range-backoff default
- resetting, 6-29

upstream scrambler
- default, 6-33
- disabling, 6-33
- re-enabling, 6-33

User Datagram Protocol, 9-31

User EXEC mode, 2-4, 2-5, 2-12

user models
- creating, 4-18

using
- command history, 2-11

USM security
- configuring, 4-18

V

VACM security
- configuring, 4-18

verifying
- changes, 6-43
- downstream rate-limiting status, 6-14
- host authorization, 6-6
- maximum CPE host number, 6-49, 6-124
- synchronization message interval, 6-45

viewing
- BPI settings, 6-50
- BPI statistics, 6-50
- CMTS interface BPI settings, 6-50
- CMTS interface BPI statistics, 6-50
- modulation profile groups, 6-43

virtual endpoint, 11-11

virtual private networks, 1-2

VPNs, 1-2