



Cisco ONS 15454 SONET/SDH ML-Series Multilayer Ethernet Card Software Feature and Configuration Guide

Cisco IOS Release 12.1(19)EO
Product and Documentation Release 4.1.x
Last Updated: October 25, 2007

Corporate Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7815702=
Text Part Number: 78-15702-02



THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0710R)

Cisco ONS 15454 SONET/SDH ML-Series Multilayer Ethernet Card Software Feature and Configuration Guide
Copyright © 2007 Cisco Systems, Inc. All rights reserved.



| | |
|---|-----------|
| About the Cisco IOS Documentation | xv |
| Document Objectives | xv |
| Audience | xv |
| Document Organization | xvi |
| Related Documentation | xvii |
| Document Conventions | xviii |
| Where to Find Safety and Warning Information | xix |
| Obtaining Documentation | xix |
| Cisco.com | xix |
| Ordering Documentation | xix |
| Cisco Optical Networking Product Documentation CD-ROM | xix |
| Documentation Feedback | xx |
| Obtaining Technical Assistance | xx |
| Cisco TAC Website | xx |
| Opening a TAC Case | xx |
| TAC Case Priority Definitions | xxi |
| Obtaining Additional Publications and Information | xxi |

CHAPTER 1

| | |
|---|------------|
| Overview | 1-1 |
| ML-Series Card Description | 1-1 |
| ML-Series Feature List | 1-2 |
| Key ML-Series Features | 1-5 |
| SONET/SDH Port Encapsulation (HDLC, PPP/BCP, and LEX) | 1-5 |
| Link Aggregation (FEC, GEC, and POS) | 1-5 |
| SONET Circuits | 1-5 |
| SDH Circuits | 1-6 |
| SONET Alarms | 1-6 |
| SDH Alarms | 1-6 |
| VRF Lite | 1-6 |
| RPR | 1-6 |
| DRPRI | 1-7 |
| TL1 | 1-7 |
| SNMP | 1-7 |
| Cisco IOS | 1-7 |

CHAPTER 2

CTC Operations 2-1

- Displaying ML-Series Ethernet Statistics on CTC 2-1
- Displaying ML-Series POS Statistics on CTC 2-3
- Displaying ML-Series Ethernet Ports Provisioning Information on CTC 2-5
- Displaying ML-Series POS Ports Provisioning Information on CTC 2-7
- Managing SONET/SDH Alarms 2-8
- SONET/SDH Circuit Provisioning 2-9

CHAPTER 3

Initial Configuration 3-1

- Hardware Installation 3-1
- ML-Series Access 3-1
 - Opening a Cisco IOS Session Using CTC 3-2
 - Telnetting to the Node IP Address and Slot Number 3-2
 - Telnetting to a Management Port 3-4
 - ML-Series IOS CLI Console Port 3-4
 - RJ-11 to RJ-45 Console Cable Adapter 3-4
 - Connecting a PC or Terminal to the Console Port 3-5
- Startup Configuration File 3-6
 - Manually Creating a Startup Configuration File Through the Serial Console Port 3-7
 - Passwords 3-7
 - Configuring the Management Port 3-8
 - Configuring the Hostname 3-9
 - Loading a Cisco IOS Startup Configuration File Through CTC 3-9
- Cisco IOS Command Modes 3-10
- Using the Command Modes 3-12
 - Exit 3-12
 - Getting Help 3-12

CHAPTER 4

Configuring Interfaces 4-1

- Interface Configuration 4-1
 - MAC Addresses 4-2
 - Interface Port ID 4-2
- Instructions for Configuring Interfaces 4-3
- Understanding Interfaces 4-4
 - Configuring the Fast Ethernet Interfaces (ML100T-12) 4-4
 - Configuring the Gigabit Ethernet Interface (ML1000-2) 4-5
 - Monitoring Operations on the Fast Ethernet and Gigabit Ethernet Interfaces 4-6
- POS on the ML-Series Card 4-8

| | |
|---|------|
| ML-Series SONET/SDH Transmission Rates | 4-8 |
| SONET Frame Fundamentals | 4-8 |
| C2 Byte | 4-9 |
| C2 Byte and Scrambling | 4-10 |
| Third-Party POS Interfaces | 4-11 |
| Configuring the ML-Series POS Interfaces | 4-12 |
| Monitoring Operations on the POS Interface and POS Controller | 4-13 |
| Additional Configurations | 4-14 |
| Setting the MTU Size | 4-14 |
| Configuring Framing | 4-15 |
| Configuring POS SPE Scrambling | 4-15 |
| SONET/SDH Alarms | 4-15 |
| Configuring SONET/SDH Alarms | 4-15 |
| Common ML-Series POS Configurations | 4-17 |
| ML-Series Card to ML-Series Card | 4-17 |
| ML-Series Card to Cisco 12000 GSR-Series Router | 4-18 |
| ML-Series Card to G-Series Card | 4-20 |

CHAPTER 5**Configuring Bridging 5-1**

| | |
|-----------------------------------|-----|
| Understanding Bridging | 5-1 |
| Configuring Bridging | 5-2 |
| Monitoring and Verifying Bridging | 5-3 |

CHAPTER 6**Configuring STP and RSTP 6-1**

| | |
|--|-----|
| STP Features | 6-1 |
| STP Overview | 6-2 |
| Supported STP Instances | 6-2 |
| Bridge Protocol Data Units | 6-2 |
| Election of the Root Switch | 6-3 |
| Bridge ID, Switch Priority, and Extended System ID | 6-4 |
| Spanning-Tree Timers | 6-4 |
| Creating the Spanning-Tree Topology | 6-4 |
| Spanning-Tree Interface States | 6-5 |
| Blocking State | 6-6 |
| Listening State | 6-7 |
| Learning State | 6-7 |
| Forwarding State | 6-7 |
| Disabled State | 6-7 |
| Spanning-Tree Address Management | 6-8 |

- STP and IEEE 802.1Q Trunks 6-8
- Spanning Tree and Redundant Connectivity 6-8
- Accelerated Aging to Retain Connectivity 6-9
- RSTP 6-9
 - Supported RSTP Instances 6-9
 - Port Roles and the Active Topology 6-9
 - Rapid Convergence 6-10
 - Synchronization of Port Roles 6-12
 - Bridge Protocol Data Unit Format and Processing 6-12
 - Processing Superior BPDU Information 6-13
 - Processing Inferior BPDU Information 6-13
 - Topology Changes 6-13
- Interoperability with 802.1D STP 6-14
- Configuring STP and RSTP Features 6-14
 - Default STP and RSTP Configuration 6-15
 - Disabling STP and RSTP 6-15
 - Configuring the Root Switch 6-16
 - Configuring the Port Priority 6-16
 - Configuring the Path Cost 6-17
 - Configuring the Switch Priority of a Bridge Group 6-18
 - Configuring the Hello Time 6-18
 - Configuring the Forwarding-Delay Time for a Bridge Group 6-18
 - Configuring the Maximum-Aging Time for a Bridge Group 6-19
- Verifying and Monitoring STP and RSTP Status 6-19

CHAPTER 7

Configuring VLANs 7-1

- Understanding VLANs 7-1
- Configuring IEEE 802.1Q VLAN Encapsulation 7-2
- IEEE 802.1Q VLAN Configuration 7-3
- Monitoring and Verifying VLAN Operation 7-5

CHAPTER 8

Configuring IEEE 802.1Q and Layer 2 Protocol Tunneling 8-1

- Understanding IEEE 802.1Q Tunneling 8-1
- Configuring IEEE 802.1Q Tunneling 8-4
 - IEEE 802.1Q Tunneling and Other Features 8-4
 - Configuring an IEEE 802.1Q Tunneling Port 8-4
- Understanding Layer 2 Protocol Tunneling 8-6
- Configuring Layer 2 Protocol Tunneling 8-7

| | |
|---|-----|
| Default Layer 2 Protocol Tunneling Configuration | 8-7 |
| Layer 2 Protocol Tunneling Configuration Guidelines | 8-7 |
| Configuring a Layer 2 Tunneling Port | 8-8 |
| Monitoring and Verifying Tunneling Status | 8-9 |

CHAPTER 9**Configuring Link Aggregation 9-1**

| | |
|--|-----|
| Understanding Link Aggregation | 9-1 |
| Configuring EtherChannel | 9-2 |
| EtherChannel Configuration Example | 9-3 |
| Configuring POS Channel | 9-4 |
| POS Channel Configuration Example | 9-5 |
| Understanding Encapsulation over EtherChannel or POS Channel | 9-7 |
| Configuring Encapsulation over EtherChannel or POS Channel | 9-7 |
| Encapsulation over EtherChannel Example | 9-7 |
| Monitoring and Verifying EtherChannel and POS | 9-9 |

CHAPTER 10**Configuring Networking Protocols 10-1**

| | |
|---|-------|
| Understanding IP Routing Protocols | 10-1 |
| Basic IP Routing Protocol Configuration | 10-2 |
| RIP | 10-2 |
| EIGRP | 10-3 |
| OSPF | 10-3 |
| BGP | 10-4 |
| Enabling IP Routing | 10-4 |
| Configuring IP Routing | 10-5 |
| Configuring RIP | 10-5 |
| RIP Authentication | 10-8 |
| Summary Addresses and Split Horizon | 10-8 |
| Configuring OSPF | 10-9 |
| OSPF Interface Parameters | 10-13 |
| OSPF Area Parameters | 10-15 |
| Other OSPF Behavior Parameters | 10-17 |
| Change LSA Group Pacing | 10-19 |
| Loopback Interface | 10-19 |
| Monitoring OSPF | 10-20 |
| Configuring EIGRP | 10-21 |
| EIGRP Router Mode Commands | 10-23 |
| EIGRP Interface Mode Commands | 10-24 |
| Configure EIGRP Route Authentication | 10-25 |

- Monitoring and Maintaining EIGRP 10-26
- Border Gateway Protocol and Classless Interdomain Routing 10-28
- Configuring BGP 10-28
- Verifying the BGP Configuration 10-28
- Configuring IS-IS 10-30
- Verifying the IS-IS Configuration 10-31
- Configuring Static Routes 10-32
- Monitoring Static Routes 10-33
- Monitoring and Maintaining the IP Network 10-33
- Understanding IP Multicast Routing 10-34
- Configuring IP Multicast Routing 10-35
- Monitoring and Verifying IP Multicast Operation 10-35

CHAPTER 11

Configuring IRB 11-1

- Understanding Integrated Routing and Bridging 11-1
 - Configuring IRB 11-2
 - Configuring IRB 11-3
- Monitoring and Verifying IRB 11-4

CHAPTER 12

Configuring VRF Lite 12-1

- Understanding VRF Lite 12-1
- Configuring VRF Lite 12-2
- VRF Lite Configuration Example 12-2
- Monitoring and Verifying VRF Lite 12-7

CHAPTER 13

Configuring Quality of Service 13-1

- Understanding QoS 13-1
 - Priority Mechanism in IP and Ethernet 13-1
 - IP Precedence and Differentiated Services Code Point 13-2
 - Ethernet CoS 13-2
- ML-Series QoS 13-3
 - Classification 13-4
 - Policing 13-4
 - Marking and Discarding 13-5
 - Queuing 13-6
 - Scheduling 13-6
 - Multicast QoS 13-7
 - Control Packets and L2 Tunneled Protocols 13-8

| | |
|--|-------|
| Priority Marking | 13-8 |
| QinQ Implementation | 13-9 |
| Flow Control Pause and QoS | 13-9 |
| QoS on RPR | 13-10 |
| Configuring QoS | 13-10 |
| Creating a Traffic Class | 13-11 |
| Creating a Traffic Policy | 13-12 |
| Attaching a Traffic Policy to an Interface | 13-15 |
| Configuring CoS-based QoS | 13-16 |
| Monitoring and Verifying QoS | 13-16 |
| Configuration Examples | 13-17 |
| Traffic Classes Defined Example | 13-17 |
| Traffic Policy Created Example | 13-18 |
| class-map match-any and class-map match-all Commands Example | 13-18 |
| match spr1 Interface Example | 13-19 |
| ML-Series VoIP Example | 13-19 |
| ML-Series Policing Example | 13-20 |
| ML-Series CoS-based QoS Example | 13-21 |

CHAPTER 14**Configuring the Switching Database Manager 14-1**

| | |
|--|------|
| Understanding the SDM | 14-1 |
| SDM Regions | 14-1 |
| Configuring SDM | 14-2 |
| Configuring SDM Regions | 14-2 |
| Configuring Access Control List Size in TCAM | 14-3 |

CHAPTER 15**Configuring Access Control Lists 15-1**

| | |
|--|------|
| Understanding ACLs | 15-1 |
| ML-Series ACL Support | 15-1 |
| IP ACLs | 15-2 |
| Named IP ACLs | 15-2 |
| User Guidelines | 15-2 |
| Creating IP ACLs | 15-3 |
| Creating Numbered Standard and Extended IP ACLs | 15-3 |
| Creating Named Standard IP ACLs | 15-4 |
| Creating Named Extended IP ACLs (Control Plane Only) | 15-4 |
| Applying the ACL to an Interface | 15-4 |
| Modifying ACL TCAM Size | 15-5 |

CHAPTER 16

Configuring Resilient Packet Ring 16-1

- Understanding RPR 16-1
 - Packet Handling Operations 16-2
 - Ring Wrapping 16-3
 - MAC Address and VLAN Support 16-4
- Configuring Point-to-Point Circuits on CTC for RPR 16-4
- Configuring RPR on IOS 16-5
 - RPR IOS Configuration 16-7
- Monitoring and Verifying RPR 16-10
- Understanding Dual RPR Interconnect 16-10
- Configuring DRPRI 16-12
 - DRPRI IOS Configuration 16-13
- Monitoring and Verifying DRPRI 16-16

APPENDIX A

Command Reference A-1

APPENDIX B

Unsupported CLI Commands B-1

- Unsupported Privileged Exec Commands B-1
- Unsupported Global Configuration Commands B-1
- Unsupported POS Interface Configuration Commands B-3
- Unsupported FastEthernet or GigabitEthernet Interface Configuration Commands B-4
- Unsupported Port-Channel Interface Configuration Commands B-5
- Unsupported BVI Interface Configuration Commands B-5

APPENDIX C

Using Technical Support C-1

- Gathering Information About Your Internetwork C-1
- Getting the Data from Your ML-Series Card C-2
- Providing Data to Your Technical Support Representative C-3

INDEX



FIGURES

| | | |
|-------------|--|-------|
| Figure 2-1 | Displaying ML-Series Ethernet Statistics | 2-2 |
| Figure 2-2 | Displaying ML-Series POS Statistics | 2-4 |
| Figure 2-3 | Displaying ML-Series Ethernet Port Provisioning Information | 2-6 |
| Figure 2-4 | Displaying POS Port Provisioning Information | 2-7 |
| Figure 2-5 | Managing ML-Series SONET/SDH Alarms | 2-8 |
| Figure 3-1 | CTC IOS Window | 3-2 |
| Figure 3-2 | CTC Node View Showing IP Address and Slot Number | 3-3 |
| Figure 3-3 | Console Cable Adapter | 3-4 |
| Figure 3-4 | Connecting to the Console Port | 3-6 |
| Figure 4-1 | Three SONET Layers | 4-9 |
| Figure 4-2 | ML-Series Card to ML-Series Card POS Configuration | 4-18 |
| Figure 4-3 | ML-Series Card to Cisco 12000 Series Gigabit Switch Router (GSR) POS Configuration | 4-19 |
| Figure 4-4 | ML-Series Card to G-Series Card POS Configuration | 4-20 |
| Figure 6-1 | Spanning-Tree Topology | 6-5 |
| Figure 6-2 | Spanning-Tree Interface States | 6-6 |
| Figure 6-3 | Spanning Tree and Redundant Connectivity | 6-8 |
| Figure 6-4 | Proposal and Agreement Handshaking for Rapid Convergence | 6-11 |
| Figure 6-5 | Sequence of Events During Rapid Convergence | 6-12 |
| Figure 7-1 | VLANs Spanning Devices in a Network | 7-2 |
| Figure 7-2 | Bridging IEEE 802.1Q VLANs | 7-4 |
| Figure 8-1 | IEEE 802.1Q Tunnel Ports in a Service-Provider Network | 8-2 |
| Figure 8-2 | Normal, IEEE 802.1Q, and 802.1Q Tunneled Ethernet Packet Formats | 8-3 |
| Figure 9-1 | Encapsulation over EtherChannel Example | 9-8 |
| Figure 10-1 | IP Routing Protocol Example Using OSPF | 10-12 |
| Figure 12-1 | VRF Lite—Sample Network Scenario | 12-3 |
| Figure 13-1 | IP Precedence and Differentiated Service Code Point (DSCP) | 13-2 |
| Figure 13-2 | Ethernet frame and the CoS bit (802.1p) | 13-3 |
| Figure 13-3 | ML-Series QoS Flow | 13-3 |
| Figure 13-4 | Dual Leaky Bucket Policer Model | 13-5 |
| Figure 13-5 | Queuing and Scheduling Model | 13-7 |
| Figure 13-6 | QinQ | 13-9 |

| | | |
|-----------------------------|--|--------------|
| Figure 13-7 | ML-Series VoIP Example | 13-19 |
| Figure 13-8 | ML-Series Policing Example | 13-20 |
| Figure 13-9 | ML-Series CoS Example | 13-21 |
| Figure 16-1 | RPR Packet Handling Operations | 16-2 |
| Figure 16-2 | RPR Ring Wrapping | 16-3 |
| Figure 16-3 | Dual RPR Interconnect Network and Paired Cards | 16-11 |



TABLES

| | | |
|-------------|---|--------------|
| Table 1 | Documentation Conventions | xviii |
| Table 2-1 | ML-Series Ethernet Statistics Fields and Buttons | 2-2 |
| Table 2-2 | Ethernet Parameters | 2-2 |
| Table 2-3 | ML-Series POS Statistics Fields and Buttons | 2-4 |
| Table 2-4 | POS Parameters | 2-4 |
| Table 3-1 | RJ-11 to RJ-45 Pin Mapping | 3-4 |
| Table 3-2 | IOS Command Modes | 3-11 |
| Table 4-1 | Transmission Multiples Supported by ML-Series Cards | 4-8 |
| Table 4-2 | C2 Byte Common Values | 4-9 |
| Table 4-3 | Default MTU Size | 4-15 |
| Table 4-4 | ML-Series Parameter Configuration for Connection to a Cisco 12000 GSR-Series Router | 4-20 |
| Table 6-1 | Switch Priority Value and Extended System ID | 6-4 |
| Table 6-2 | Spanning-Tree Timers | 6-4 |
| Table 6-3 | Port State Comparison | 6-10 |
| Table 6-4 | RSTP BPDU Flags | 6-12 |
| Table 6-5 | Default STP and RSTP Configuration | 6-15 |
| Table 6-6 | Commands for Displaying Spanning-Tree Status | 6-19 |
| Table 8-1 | Default Layer 2 Protocol Tunneling Configuration | 8-7 |
| Table 8-2 | Commands for Monitoring and Maintaining Tunneling | 8-9 |
| Table 10-1 | Default RIP Configuration | 10-5 |
| Table 10-2 | Default OSPF Configuration | 10-10 |
| Table 10-3 | Show IP OSPF Statistics Commands | 10-20 |
| Table 10-4 | Default EIGRP Configuration | 10-22 |
| Table 10-5 | IP EIGRP Clear and Show Commands | 10-27 |
| Table 10-6 | BGP Show Commands | 10-29 |
| Table 10-7 | IS-IS Show Commands | 10-31 |
| Table 10-8 | Routing Protocol Default Administrative Distances | 10-33 |
| Table 10-9 | Commands to Clear IP Routes or Display Route Status | 10-34 |
| Table 10-10 | IP Multicast Routing Show Commands | 10-36 |
| Table 11-1 | Commands for Monitoring and Verifying IRB | 11-5 |
| Table 11-2 | show interfaces irb Field Descriptions | 11-6 |

| | | |
|----------------------------|---|--------------|
| Table 12-1 | Commands for Monitoring and Verifying VRF Lite | 12-7 |
| Table 13-1 | Commands for QoS Status | 13-16 |
| Table 14-1 | Default Partitioning by Application Region in TCAM | 14-2 |
| Table 14-2 | Partitioning the TCAM Size for ACLs | 14-3 |
| Table 15-1 | Commands for Numbered Standard and Extended IP ACLs | 15-3 |
| Table 15-2 | Applying ACL to Interface | 15-5 |



About the Cisco IOS Documentation



Note

The terms "Unidirectional Path Switched Ring" and "UPSR" may appear in Cisco literature. These terms do not refer to using Cisco ONS 15xxx products in a unidirectional path switched ring configuration. Rather, these terms, as well as "Path Protected Mesh Network" and "PPMN," refer generally to Cisco's path protection feature, which may be used in any topological network configuration. Cisco does not recommend using its path protection feature in any particular topological network configuration.

This section explains the objectives, intended audience, and organization of this publication and describes the conventions that convey instructions and other information.

This section provides the following information:

- [Document Objectives](#)
- [Audience](#)
- [Document Organization](#)
- [Related Documentation](#)
- [Document Conventions](#)
- [Where to Find Safety and Warning Information](#)
- [Obtaining Documentation](#)
- [Documentation Feedback](#)
- [Obtaining Technical Assistance](#)
- [Obtaining Additional Publications and Information](#)

Document Objectives

This guide explains software features and configuration for Cisco IOS on the ML-Series card. The ML-Series card is a module in the Cisco ONS 15454 SONET/SDH system. Use this guide in conjunction with the appropriate publications listed in the [Related Documentation](#) section.

Audience

To use this publication, you should be familiar with Cisco IOS and preferably have technical networking background and experience.

Document Organization

This *Cisco ONS 15454 SONET/SDH ML-Series Multilayer Ethernet Card Software Feature and Configuration Guide, R4.1.x* is organized into the following chapters:

- [Chapter 1, “Overview,”](#) provides a description of the ML-Series card, a feature list and explanations of key features.
- [Chapter 2, “CTC Operations,”](#) provides details and procedures for using Cisco Transport Controller (CTC) software with the ML-Series Card.
- [Chapter 3, “Initial Configuration,”](#) provides procedures to access the ML-Series card and create and manage startup configuration files.
- [Chapter 4, “Configuring Interfaces,”](#) provides information on the ML-Series card interfaces and procedures for the interfaces.
- [Chapter 5, “Understanding Bridging,”](#) provides bridging examples and procedures for the ML-Series card.
- [Chapter 6, “Configuring STP and RSTP,”](#) provides spanning tree and rapid spanning tree examples and procedures for the ML-Series card.
- [Chapter 7, “Configuring VLANs,”](#) provides VLAN examples and procedures for the ML-Series card.
- [Chapter 8, “Configuring IEEE 802.1Q and Layer 2 Protocol Tunneling,”](#) provides tunneling examples and procedures for the ML-Series card.
- [Chapter 9, “Configuring Link Aggregation,”](#) provides Etherchannel and POS channel examples and procedures for the ML-Series card.
- [Chapter 10, “Configuring Networking Protocols,”](#) provides network protocol examples and procedures for the ML-Series card.
- [Chapter 11, “Understanding Integrated Routing and Bridging,”](#) provides integrated routing and bridging (IRB) examples and procedures for the ML-Series card.
- [Chapter 12, “Configuring VRF Lite,”](#) provides VRF Lite examples and procedures for the ML-Series card.
- [Chapter 13, “Configuring Quality of Service,”](#) provides QoS examples and procedures for the ML-Series card.
- [Chapter 14, “Configuring the Switching Database Manager,”](#) provides switching database manager examples and procedures for the ML-Series card.
- [Chapter 15, “Configuring Access Control Lists,”](#) provides access control list (ACL) examples and procedures for the ML-Series card.
- [Chapter 16, “Configuring Resilient Packet Ring,”](#) provides resilient packet ring (RPR) examples and procedures for the ML-Series card.
- [Appendix A, “Command Reference,”](#) is an alphabetical listing of unique ML-Series card Cisco IOS commands with definitions and examples.
- [Appendix B, “Unsupported CLI Commands,”](#) is a categorized and alphabetized listing of Cisco IOS commands that the ML-Series card does not support.
- [Appendix C, “Using Technical Support,”](#) instructs the user on using the Cisco Technical Assistance Center (TAC) for ML-Series card problems.

Related Documentation

Use this *Cisco ONS 15454 SONET/SDH ML-Series Multilayer Ethernet Card Software Feature and Configuration Guide, R4.1.x* in conjunction with the following general ONS 15454 or ONS 15454 SDH system publications:

- To install, turn up, provision, and maintain a Cisco ONS 15454 node and network, refer to the *Cisco ONS 15454 Procedure Guide, Release 4.1.1 and Release 4.5*.
- For alarm clearing, general troubleshooting, and hardware replacement procedures for a Cisco ONS 15454 node, refer to the *Cisco ONS 15454 Troubleshooting Guide, Release 4.1.1 and Release 4.5*.
- For detailed reference information on a Cisco ONS 15454 node, refer to the *Cisco ONS 15454 Reference Manual, R4.1.1 and R4.5*.
- To install, turn up, provision, and maintain a Cisco ONS 15454 SDH node and network, refer to the *Cisco ONS 15454 SDH Procedure Guide, Release 4.1 and Release 4.5*.
- For alarm clearing, general troubleshooting, and hardware replacement procedures for the Cisco ONS 15454 SDH node, refer to the *Cisco ONS 15454 SDH Troubleshooting Guide, Release 4.1 and Release 4.5*.
- For detailed reference information on the Cisco ONS 15454 SDH node, refer to the *Cisco ONS 15454 SDH Reference Manual, R4.1 and R4.5*.

The ML-Series card employs the Cisco IOS Modular QoS CLI (MQC). For more information on general MQC configuration, refer to the following IOS documents:

- *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.1* at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm
- *Cisco IOS Quality of Service Solutions Command Reference, Release 12.1* at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm

The ML-Series card employs Cisco IOS 12.1. For more general information on Cisco IOS 12.1, refer to the extensive Cisco IOS documentation at:

- <http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/> or
<http://www.cisco.com/univercd/home/home.htm>

Document Conventions

This publication uses the following conventions:

Table 1 Documentation Conventions

| Convention | Application |
|-----------------------------|---|
| boldface | Commands and keywords in body text. |
| <i>italic</i> | Command input that is supplied by the user. |
| [] | Keywords or arguments that appear within square brackets are optional. |
| { x x x } | A choice of keywords (represented by x) appears in braces separated by vertical bars. The user must select one. |
| Ctrl | The control key. For example, where Ctrl + D is written, hold down the Control key while pressing the D key. |
| screen font | Examples of information displayed on the screen. |
| boldface screen font | Examples of information that the user must enter. |
| < > | Command parameters that must be replaced by module-specific codes. |



Note

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



Caution

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



Warning

IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. To see translations of the warnings that appear in this publication, refer to the translated safety warnings that accompanied this device.

Note: SAVE THESE INSTRUCTIONS

Note: This documentation is to be used in conjunction with the specific product installation guide that shipped with the product. Please refer to the Installation Guide, Configuration Guide, or other enclosed additional documentation for further details.

Where to Find Safety and Warning Information

For safety and warning information, refer to the *Cisco Optical Transport Products Safety and Compliance Information* document that accompanied the product. This publication describes the international agency compliance and safety information for the Cisco ONS 15xxx systems. It also includes translations of the safety warnings that appear in the ONS 15xxx system documentation.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation on the World Wide Web at this URL:

<http://www.cisco.com/univercd/home/home.htm>

You can access the Cisco website at this URL:

<http://www.cisco.com>

International Cisco websites can be accessed from this URL:

http://www.cisco.com/public/countries_languages.shtml

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpck/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Ordering tool:
<http://www.cisco.com/en/US/partner/ordering/index.shtml>
- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Cisco Optical Networking Product Documentation CD-ROM

Optical networking-related documentation, including Cisco ONS 15454 SONET/SDH product documentation, is available in a CD-ROM package that ships with your product. The Optical Networking Product Documentation CD-ROM is updated periodically and may be more current than printed documentation.

Documentation Feedback

You can submit e-mail comments about technical documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

For all customers, partners, resellers, and distributors who hold valid Cisco service contracts, the Cisco Technical Assistance Center (TAC) provides 24-hour-a-day, award-winning technical support services, online and over the phone. Cisco.com features the Cisco TAC website as an online starting point for technical assistance. If you do not hold a valid Cisco service contract, please contact your reseller.

Cisco TAC Website

The Cisco TAC website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The Cisco TAC website is available 24 hours a day, 365 days a year. The Cisco TAC website is located at this URL:

<http://www.cisco.com/tac>

Accessing all the tools on the Cisco TAC website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a login ID or password, register at this URL:

<http://tools.cisco.com/RPF/register/register.do>

Opening a TAC Case

Using the online TAC Case Open Tool is the fastest way to open P3 and P4 cases. (P3 and P4 cases are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Case Open Tool automatically recommends resources for an immediate solution. If your issue is not resolved using the recommended resources, your case will be assigned to a Cisco TAC engineer. The online TAC Case Open Tool is located at this URL:

<http://www.cisco.com/tac/caseopen>

For P1 or P2 cases (P1 and P2 cases are those in which your production network is down or severely degraded) or if you do not have Internet access, contact Cisco TAC by telephone. Cisco TAC engineers are assigned immediately to P1 and P2 cases to help keep your business operations running smoothly.

To open a case by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)

EMEA: +32 2 704 55 55

USA: 1 800 553-2447

For a complete listing of Cisco TAC contacts, go to this URL:

<http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml>

TAC Case Priority Definitions

To ensure that all cases are reported in a standard format, Cisco has established case priority definitions.

Priority 1 (P1)—Your network is “down” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Priority 2 (P2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Priority 3 (P3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Priority 4 (P4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, and logo merchandise. Go to this URL to visit the company store:
<http://www.cisco.com/go/marketplace/>
- The Cisco *Product Catalog* describes the networking products offered by Cisco Systems, as well as ordering and customer support services. Access the Cisco Product Catalog at this URL:
<http://cisco.com/univercd/cc/td/doc/pcat/>
- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press online at this URL:
<http://www.ciscopress.com>
- *Packet* magazine is the Cisco quarterly publication that provides the latest networking trends, technology breakthroughs, and Cisco products and solutions to help industry professionals get the most from their networking investment. Included are networking deployment and troubleshooting tips, configuration examples, customer case studies, tutorials and training, certification information, and links to numerous in-depth online resources. You can access Packet magazine at this URL:
<http://www.cisco.com/packet>
- *iQ Magazine* is the Cisco bimonthly publication that delivers the latest information about Internet business strategies for executives. You can access iQ Magazine at this URL:
<http://www.cisco.com/go/iqmagazine>

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:
<http://www.cisco.com/ipj>
- Training—Cisco offers world-class networking training. Current offerings in network training are listed at this URL:
<http://www.cisco.com/en/US/learning/index.html>



Overview

This chapter provides an overview of the ML1000-2 and ML100T-12 cards for the ONS 15454 SONET and ONS 15454 SDH. It lists Ethernet and SONET/SDH capabilities and Cisco IOS and Cisco Transport Controller (CTC) software features, with brief descriptions of selected features.

This chapter contains the following major sections:

- [ML-Series Card Description, page 1-1](#)
- [ML-Series Feature List, page 1-2](#)
- [Key ML-Series Features, page 1-5](#)

ML-Series Card Description

The ML-Series cards are Gigabit Ethernet (ML1000-2) and Fast Ethernet (ML100T-12) multilayer switches integrated into the ONS 15454 SONET/SDH platform. An ONS 15454 SONET with a 10 Gigabit Cross-Connect card (XC10G) can host the card in any traffic card slot, but an ONS 15454 SONET with a Cross-Connect card (XC) or Cross Connect Virtual Tributary card (XCVT) can only host the ML-Series card in the four high-speed traffic slots. An ONS 15454 SDH can host the card in any traffic card slot with any cross-connect card.

Each card is an independent data switch that processes up to 5.7 Mpps of Layer 2 and Layer 3 switching. The card ships loaded with Cisco IOS Release 12.1(19)EO, which controls the data functions of the card. Users can access Cisco IOS in three ways: the console port on the faceplate of the card, the Ethernet ports on the ML-Series card assigned to a management VLAN, or a Telnet session. The Telnet sessions can be initiated through a terminal program on the PC or through CTC, the standard ONS 15454 SONET/SDH graphical user interface (GUI).

The Cisco IOS software image used by the ML-Series card is not permanently stored on the ML-Series card but in the flash memory of the TCC+/TCC2 card. During a hard reset, when a card is physically removed and reinserted, the Cisco IOS software image is downloaded from the flash memory of the TCC+/TCC2 to the memory cache of the ML-Series card. The cached image is then decompressed and initialized for use by the ML-Series card.

During a soft reset, when the ML-Series card is reset through CTC or Cisco IOS CLI commands, the ML-Series card checks its cache for an IOS image. If a valid and current IOS image exists, the ML-Series card decompresses and initializes the image. If the image does not exist, the ML-Series requests a new copy of the IOS image from the TCC. Caching the IOS image provides a significant time savings when a warm reset is performed.

The Cisco IOS CLI is the primary user interface for the ML-Series card. Most configuration for the card, such as Ethernet port, bridging and VLAN, can only be done via the Cisco IOS CLI. But CTC, the ONS 15454 SONET/SDH GUI, also supports the ML-Series. CTC offers ML-Series status information, SONET/SDH alarm management, Cisco IOS Telnet session initialization, Cisco IOS configuration file management and SONET/SDH circuit provisioning. SONET/SDH circuits cannot be provisioned through IOS, but must be configured through CTC. SONET circuits can also be provisioned with TL1 on the ONS 15454.

The ML100T-12 features 12 RJ-45 interfaces, and the ML1000-2 features two Small Form Factor Pluggable (SFP) slots supporting short wavelength (SX) and long wavelength (LX) optical modules. The ML100T-12 and the ML1000-2 use the same hardware and software base and offer the same feature sets.

When installed in an ONS 15454 SONET, the card features two virtual ports with a combined STS-48 maximum. The STS circuits are provisioned through the ONS 15454 GUI (CTC) in the same manner as standard OC-N card STS circuits. CTC also provides provisioning, inventory, SONET alarm reporting, and other standard ONS 15454 card functions for the ML-Series.

When installed in an ONS 15454 SDH, the card features two virtual ports with a combined VC4-16c maximum. The STS circuits are provisioned through the ONS 15454 SDH GUI (CTC) in the same manner as standard STM circuits. CTC also provides provisioning, inventory, SDH alarm reporting, and other standard ONS 15454 SDH card functions for the ML-Series.

For detailed card specifications, refer to the “Ethernet Cards” chapter of the *Cisco ONS 15454 Reference Guide* or the *Cisco ONS 15454 SDH Reference Guide*. For step-by-step instructions on configuring an ML-Series SONET STS circuit, refer to the “Creating Circuits and VT Tunnels” chapter of the *Cisco ONS 15454 Procedure Guide* or the “Creating Circuits and Tunnels” chapter of the *Cisco ONS 15454 SDH Procedure Guide*.


Note

When a process makes unusually heavy demands on the CPU of the ML-Series card, it may impair CPU response time and cause a CPUHOG error message to appear on the console. This message indicates which process used a large number of CPU cycles, such as the updating of the routing table with a large number of routes due to an event. Seeing this message as a result of card reset or other infrequent events should not be a cause for concern.

ML-Series Feature List

This section lists the features of the ML100T-12 and the ML1000-2 cards.

- Layer 1 features
 - 10/100BASE-TX half-duplex and full-duplex data transmission
 - 1000BASE-SX, 1000BASE-LX full-duplex data transmission
 - Two SONET virtual ports with maximum bandwidth of STS-48c per card on ONS 15454 SONET
 - Two SDH virtual ports with maximum bandwidth of VC4-16c per card on ONS 15454 SDH
 - Cisco high-level data link control (HDLC) SONET/SDH port encapsulation (no VLAN trunking support)
 - Point-to-Point Protocol/Bridge Control Protocol (PPP/BGP) SONET/SDH port encapsulation (VLAN trunking supported via BGP)
 - LEX SONET/SDH port encapsulation (G-Series card protocol, which supports VLAN trunking)
 - Packet-over-SONET/SDH (POS)

- POS channel (with LEX encapsulation only)
- PPP
- G-Series card compatible
- PPP over SONET/SDH (IP POS and bridging with VLANs)
- Layer 2 bridging features
 - Layer 2 transparent bridging
 - Layer 2 MAC learning, aging, and switching by hardware
 - Spanning Tree Protocol (IEEE 802.1D) per bridge group
 - Protocol tunneling
 - A maximum of 255 active bridge groups
 - Up to 60,000 MAC addresses per card, with a supported limit of 8,000 per bridge group
 - Integrated routing and bridging (IRB)
 - VLAN features
 - 802.1P/Q-based VLAN trunking
 - 802.1Q VLAN tunneling
 - 802.1D Spanning Tree and 802.1W Rapid Spanning Tree
 - Resilient Packet Ring (RPR)
 - Dual Resilient Packet Ring Interconnect (DRPRI)
- Layer 3 routing, switching, and forwarding
 - Default routes
 - IP unicast and multicast forwarding support
 - Reverse Path Forwarding (RPF) multicast (not RPF unicast)
 - Simple IP access control lists (ACLs) (both Layer 2 and Layer 3 forwarding path)
 - Extended IP ACLs in software (control-plane only)
 - IP and IP multicast routing and switching between Ethernet ports
 - Load balancing among equal cost paths based on source and destination IP addresses
 - Up to 18,000 IP routes
 - Up to 20,000 IP host entries
 - Up to 40 IP multicast groups
- Supported routing protocols
 - Virtual Private Network (VPN) Routing and Forwarding Lite (VRF Lite)
 - Intermediate System-to-Intermediate System (IS-IS) Protocol
 - Routing Information Protocol (RIP and RIP II)
 - Enhanced Interior Gateway Routing Protocol (EIGRP)
 - Open Shortest Path First (OSPF) Protocol
 - Protocol Independent Multicast (PIM)—Sparse, sparse-dense and dense modes
 - Secondary addressing
 - Static routes

- Local proxy ARP
- Border Gateway Protocol (BGP)
- Classless interdomain routing (CIDR)
- Fast EtherChannel (FEC) features (ML100T-12)
 - Bundling of up to four Fast Ethernet ports
 - Load sharing based on source and destination IP addresses of unicast packets
 - Load sharing for bridge traffic based on MAC addresses
 - IRB on the Fast EtherChannel
 - IEEE 802.1Q trunking on the Fast EtherChannel
 - Up to 6 active FEC port channels
- Gigabit EtherChannel (GEC) features (ML1000-2)
 - Bundling the two Gigabit Ethernet ports
 - Load sharing for bridge traffic based on MAC addresses
 - IRB on the Gigabit EtherChannel
 - IEEE 802.1Q trunking on the Gigabit EtherChannel
- ACL features
 - IP standard ACL
 - IP extended ACL
- VLAN features
 - IEEE 802.1Q-based VLAN routing and bridging
- QoS features
 - Service Level Agreements (SLAs) with 1-Mbps granularity
 - Input policing
 - Guaranteed bandwidth (weighted round-robin [WDRR] plus strict priority scheduling)
 - Classification based on Layer 2 priority, VLAN ID, Layer 3 TOS/DSCP, and port
 - Low latency queuing support for unicast VoIP
- CTC
 - Standard STS circuit provisioning for SONET virtual ports
 - Standard STM circuit provisioning for SDH virtual ports
 - SONET alarm reporting for path alarms and other ML-Series specific alarms on ONS 15454 SONET
 - SDH alarm reporting for path alarms and other ML-Series specific alarms on ONS 15454 SDH
 - Raw port statistics
 - Standard inventory and card management functions
 - Cisco IOS command-line interface (CLI) Telnet sessions from CTC
 - IOS startup configuration file management
- Additional protocols and features
 - Cisco Discovery Protocol (CDP) support on Ethernet ports

- Dynamic Host Configuration Protocol (DHCP) relay
- Hot Standby Router Protocol (HSRP) over 10/100 Ethernet, Gigabit Ethernet, FEC, GEC, and Bridge Group Virtual Interface (BVI)
- Internet Control Message Protocol (ICMP)
- IRB routing mode support
- Simple Network Management Protocol (SNMP)
- Transaction Language 1 (TL1)
- Cisco IOS
- NEBS3 compliant

Key ML-Series Features

This section describes selected key features of the ML-Series.

SONET/SDH Port Encapsulation (HDLC, PPP/BCP, and LEX)

The ML-Series supports three forms of SONET/SDH port encapsulation: Cisco HDLC, PPP/BCP and LEX. Cisco HDLC is standard on most Cisco data devices. It does not offer VLAN trunking support. PPP/BCP is a popular standard linked to RFC 2878. It supports VLAN trunking via BCP. LEX is a protocol used by the G-Series cards. This protocol supports VLAN trunking and is based on PPP over HDLC.

The SONET/SDH port encapsulation allows the ML-Series to connect to the OC-N ports of switches and routers supporting POS, as well as the G-Series Ethernet cards on the Cisco ONS 15454 SONET, ONS 15454 SDH, and ONS 15327. All three formats support bridging and routing, standard SONET/SDH payload scrambling, and HDLC frame check sequence.

Link Aggregation (FEC, GEC, and POS)

The ML-Series offers Fast EtherChannel, Gigabit EtherChannel, and Packet-over-SONET/SDH (POS) channel link aggregation. Link aggregation groups multiple ports into a larger logical port and provides resiliency during the failure of any individual ports. The ML-Series supports a maximum of four Ethernet ports in Fast EtherChannel, two Ethernet ports in Gigabit EtherChannel, and two SONET/SDH virtual ports in the POS channel. The POS channel is only supported with LEX encapsulation.

Traffic flows map to individual ports based on MAC source address (SA)/destination address (DA) for bridged packets and IP SA/DA for routed packets. There is no support for policing or class-based packet priorities when link aggregation is configured.

SONET Circuits

On the ONS 15454 SONET, ML-Series cards feature two SONET virtual ports with a maximum combined bandwidth of STS-48. Each port carries an STS circuit with a size of STS-1, STS-3c, STS-6c, STS-9c, STS-12c, or STS-24c.

SDH Circuits

On the ONS 15454 SDH, ML-Series cards feature two SDH virtual ports with a maximum combined bandwidth of VC4-16c. Each port carries an STM circuit with a size of VC3, VC4, VC4-2C, VC4-3C, VC4-4C or VC4-8C.

SONET Alarms

On the ONS 15454 SONET, the ML-Series card reports Telcordia GR-253 SONET alarms on the Alarms panel of CTC and in the Cisco IOS CLI. The card reports SONET Path alarms, including AIS-P, LOP-P, UNEQ-P, RFI-P, TIM-P, PLM-P, PDI-P, BER-SF-B3, and BER-SD-B3. It also reports other alarms, including BPU/COM Fail, Board Fail, port link-down, and no-config. The ML-Series also supports path trace, path, and raw port statistics on CTC. For more information on alarms and alarm definitions, refer to the “Alarm Troubleshooting” chapter of the *Cisco ONS 15454 Troubleshooting Guide* and the “Manage Alarms” chapter of the *Cisco ONS 15454 Procedure Guide*.

SDH Alarms

On the ONS 15454 SDH, the ML-Series card reports SDH alarms on the Alarms panel of CTC and other alarms, including BPU/COM Fail, Board Fail, port link-down, and no-config. The ML-Series also supports path trace, path, and raw port statistics on CTC. For more information on alarms, refer to the “Alarm Troubleshooting” chapter of the *Cisco ONS 15454 SDH Troubleshooting Guide* and the “Manage Alarms” chapter of the *Cisco ONS 15454 SDH Procedure Guide*.

VRF Lite

VPN Routing/Forwarding Lite (VRF Lite) is an ML-Series specific implementation of a VPN routing/forwarding instance (VRF). Unlike standard VRF, VRF Lite does not contain Multi-Protocol internal BGP (MP-iBGP).

Standard VRF is an extension of IP routing that provides multiple routing instances and separate IP routing and forwarding tables for each VPN. VRF is used in concert with internal MP-iBGP. MP-iBGP distributes the VRF information between routers to provide Layer 3 Multiprotocol Label Switching (MPLS)-VPN.

VRF Lite stores VRF information locally and does not distribute the VRF information to connected equipment. VRF information directs traffic to the correct interfaces and subinterfaces when the traffic is received from customer routers or from service provider router(s).

VRF Lite allows an ML-Series card, acting as customer equipment, to have multiple interfaces and subinterfaces with service provider equipment. The customer ML-Series card can then service multiple customers. Normal customer equipment serves a single customer.

RPR

RPR is an emerging network architecture designed for metro fiber ring networks. This new MAC protocol is designed to overcome the limitations of STP, RSTP and SONET in packet-based networks. RPR convergence times are comparable to SONET and much faster than STP or RSTP. RPR operates at the Layer 2 level and is compatible with Ethernet and protected or unprotected SONET circuits.

DRPRI

The bridge-group protocol DRPRI is an RPR mechanism that interconnects rings for protection from ONS node failure. The protocol provides two parallel connections of the rings linked by a special instance of RSTP. One connection is the active node and the other is the standby node. During a failure of the active node, link or card, a proprietary algorithm detects the failure and causes a switchover to the standby node.

TL1

On the ONS 15454 SONET, the Transaction Language 1 (TL1) on the ML-Series can be used for card inventory, fault or alarm management, card provisioning, and retrieval of status information for both data and SONET ports. TL1 can also be used to provision SONET STS circuits and transfer a Cisco IOS startup configuration file to the Timing Communications and Control+ Card (TCC+) or Timing Communications and Control 2 Card (TCC2) memory. For specific TL1 commands and general TL1 information, refer to the *Cisco ONS 15454 and Cisco ONS 15327 TL1 Command Guide*.

SNMP

Both the ONS 15454 SONET/SDH and the ML-Series have Simple Network Management Protocol (SNMP) agents and support SNMP Version 1 (SNMPv1) and SNMP Version 2c (SNMPv2c) sets and traps. The ONS 15454 accepts, validates, and forwards get/getNext/set requests to the ML-Series through a proxy agent. The ML-Series requests contain the slot identification of the ML-Series card to distinguish the request from a general ONS 15454 SNMP request. Responses from the ML-Series are relayed by the ONS 15454 to the requesting SNMP agents.

The ML-Series supports SNMP traps, including Spanning Tree Protocol (STP) traps from Bridge-MIB (management information base) (RFC 1493), the authentication traps from RFC 1157, and the Link-up and Link-down traps for Ethernet ports from IF-MIB (RFC 1573). For more information on how the ONS 15454 implements SNMP, refer to the “SNMP” chapter of the *Cisco ONS 15454 Reference Guide*.

Cisco IOS

Cisco IOS controls the data functions of the ML-Series card and comes preloaded on the ONS 15454 TCC+/TCC2 card.

Users cannot update the ML-Series Cisco IOS image in the same manner as the Cisco IOS system image on a Cisco Catalyst Series. An ML-Series Cisco IOS image upgrade is accomplished only through the ONS 15454 SONET/SDH CTC, and Cisco IOS images for the ML-Series card are available only as part of an ONS 15454 software release. This Cisco IOS image is included on the standard ONS 15454 SONET/SDH System Software CD under the package file name M_I.bin and full file name ons15454m-i7-mz. The images are not available for download or shipped separately.



CTC Operations

This chapter covers Cisco Transport Controller (CTC) operations of the ML-Series card. All operations described in the chapter take place at the card-level view of CTC. CTC shows provisioning information and statistics for both the Ethernet and packet over SONET/SDH (POS) ports of the ML-Series card. For the ML-Series cards, CTC manages SONET/SDH alarms and provisions STS/STM circuits in the same manner as other ONS 15454 SONET/SDH traffic cards.

Use CTC to load a Cisco Internet Operating System (IOS) configuration file or to open a Cisco IOS command line interface (CLI) session, see the [“Loading a Cisco IOS Startup Configuration File Through CTC”](#) section on page 3-9 or the [“Loading a Cisco IOS Startup Configuration File Through CTC”](#) section on page 3-9.

This chapter contains the following major sections:

- [Displaying ML-Series Ethernet Statistics on CTC, page 2-1](#)
- [Displaying ML-Series POS Statistics on CTC, page 2-3](#)
- [Displaying ML-Series Ethernet Ports Provisioning Information on CTC, page 2-5](#)
- [Displaying ML-Series POS Ports Provisioning Information on CTC, page 2-7](#)
- [Managing SONET/SDH Alarms, page 2-8](#)
- [SONET/SDH Circuit Provisioning, page 2-9](#)

Displaying ML-Series Ethernet Statistics on CTC

The Ethernet statistics window ([Figure 2-1 on page 2-2](#)) lists Ethernet port-level statistics. The ML-Series Ethernet ports are zero based. Display the CTC card view for the ML-Series card and click the **Performance > Ether Ports** tabs to display the window.

Figure 2-1 Displaying ML-Series Ethernet Statistics

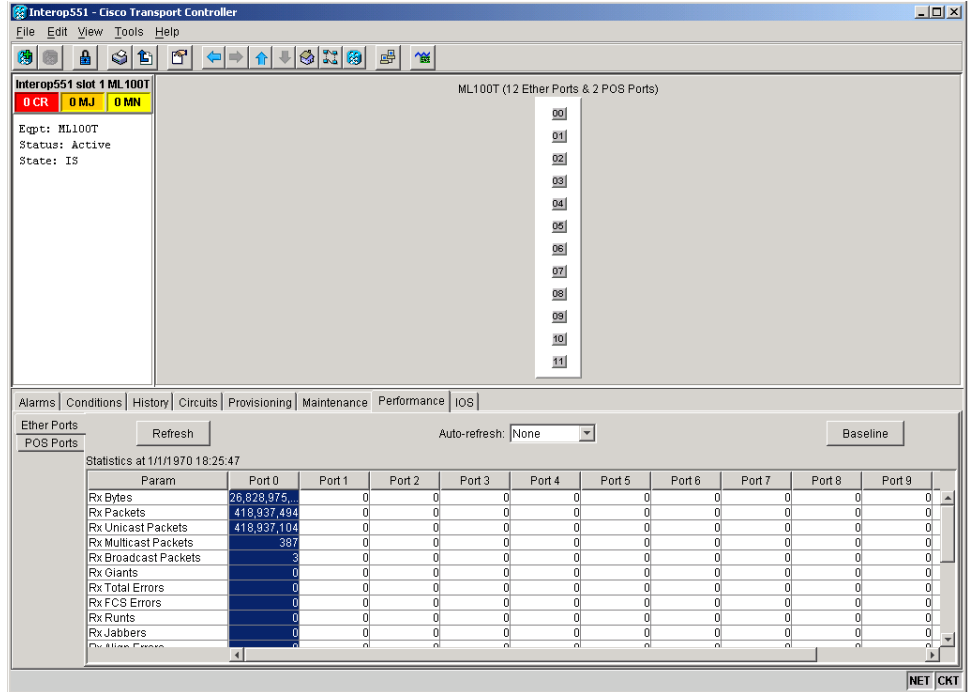


Table 2-1 describes the buttons in the EtherPorts window.

Table 2-1 ML-Series Ethernet Statistics Fields and Buttons

| Button or Field | Description |
|---------------------|---|
| Baseline | Resets the software counters (in that particular CTC client only) temporarily to zero without affecting the actual statistics on the card. After the reset, only the delta in counters appear in the CTC session. These new baselined counters appear as long as the user displays the Performance window. If the user navigates to another CTC window and comes back to the Performance window, the true actual statistics retained by the card are shown. |
| Refresh | Queries the current values from the card and updates the CTC display. |
| Auto-Refresh | Sets a time interval for the automatic refresh of statistics. |

Table 2-2 lists the ONS 15454 Ethernet parameters.

Table 2-2 Ethernet Parameters

| Parameter | Meaning |
|----------------------|---|
| Rx Bytes | Number of bytes received since the last counter reset |
| Rx Packets | Number of packets received since the last counter reset |
| Rx Unicast Packets | Number of unicast packets received |
| Rx Multicast Packets | Number of multicast packets received |
| Rx Broadcast Packets | Number of broadcast packets received |

Table 2-2 Ethernet Parameters (continued)

| Parameter | Meaning |
|------------------------|---|
| Rx Giants | Number of packets received that are greater than 1530 bytes in length |
| Rx Total Errors | Total number of receive errors |
| Rx FCS Errors | Number of packets with a frame check sequence (FCS) error |
| Rx Runts | Total number of frames received that are less than 64 bytes in length and have a cyclic redundancy check (CRC) error |
| Rx Jabbers | Total number of frames received that exceed the maximum 1548 bytes and contain CRC errors |
| Rx Align Errors | Number of received packets with alignment errors |
| Tx Bytes | Number of bytes transmitted since the last counter reset |
| Tx Packets | Number of packets transmitted since the last counter reset |
| Tx Unicast Packets | Number of unicast packets transmitted |
| Tx Multicast Packets | Number of multicast packets transmitted |
| Tx Broadcast Packets | Number of broadcast packets transmitted |
| Tx Giants | Number of packets transmitted that are greater than 1548 bytes in length |
| Tx Collisions | Number of transmitted packets that collided |
| Port Drop Counts | Number of received frames dropped at the port level |
| Rx Pause Frames | Number of received pause frames (applies only to the ML1000-2 Ethernet ports) |
| Rx Threshold Oversizes | Number of received packets larger than the ML-Series remote monitoring (RMON) threshold (applies only to the ML1000-2 Ethernet ports) |
| Rx GMAC Drop Counts | Number of received frames dropped by MAC module (applies only to the ML1000-2 Ethernet ports) |
| Tx Pause Frames | Number of transmitted pause frames (applies only to the ML1000-2 Ethernet ports) |

Displaying ML-Series POS Statistics on CTC

The POS statistics window lists POS port-level statistics (Figure 2-2). Display the CTC card view for the ML-Series card and click the **Performance > POS Ports** tabs to display the window.

Figure 2-2 Displaying ML-Series POS Statistics

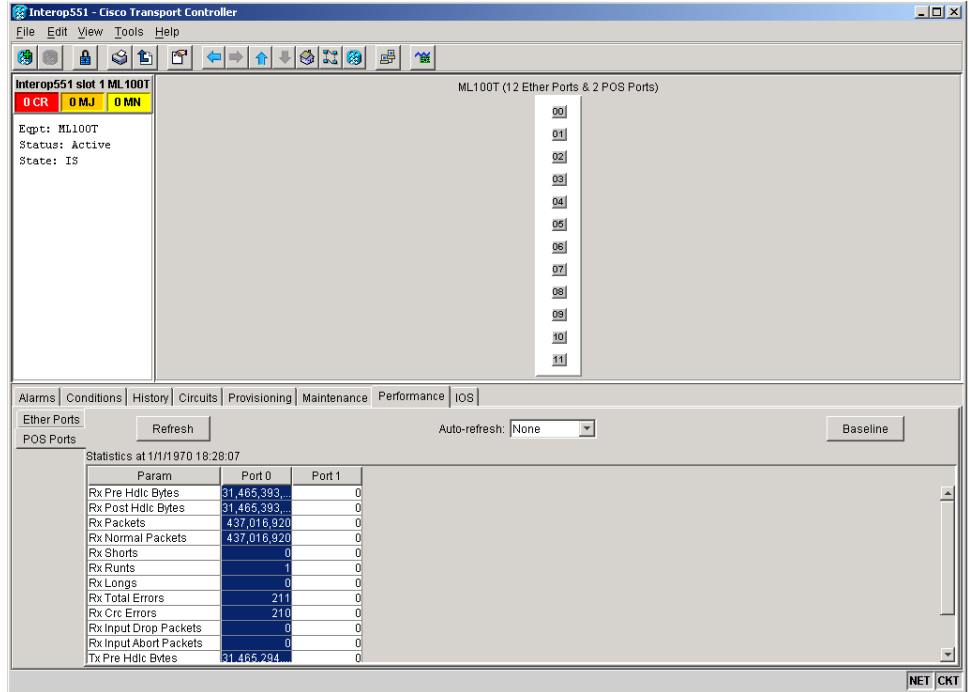


Table 2-3 describes the buttons in the POS Ports window.

Table 2-3 ML-Series POS Statistics Fields and Buttons

| Button or Field | Description |
|---------------------|--|
| Baseline | Resets the software counters (in that particular CTC client only) temporarily to zero without affecting the actual statistics on the card. From that point on, only the delta in counters are displayed by this CTC. These new baselined counters are shown only as long as the user displays the Performance window. If the user navigates to another CTC window and comes back to the Performance window, the true actual statistics retained by the card are shown. |
| Refresh | Manually refreshes the statistics. |
| Auto-Refresh | Sets a time interval for the automatic refresh of statistics. |

Table 2-4 lists the ONS 15454 POS parameters.

Table 2-4 POS Parameters

| Parameter | Meaning |
|--------------------|--|
| Rx Pre Hdlc Bytes | Number of bytes received prior to the bytes undergoing HLDC encapsulation by the policy engine |
| Rx Post Hdlc Bytes | Number of bytes received after the bytes undergoing HLDC encapsulation by the policy engine |
| Rx Packets | Total number of packets received since the last counter reset |

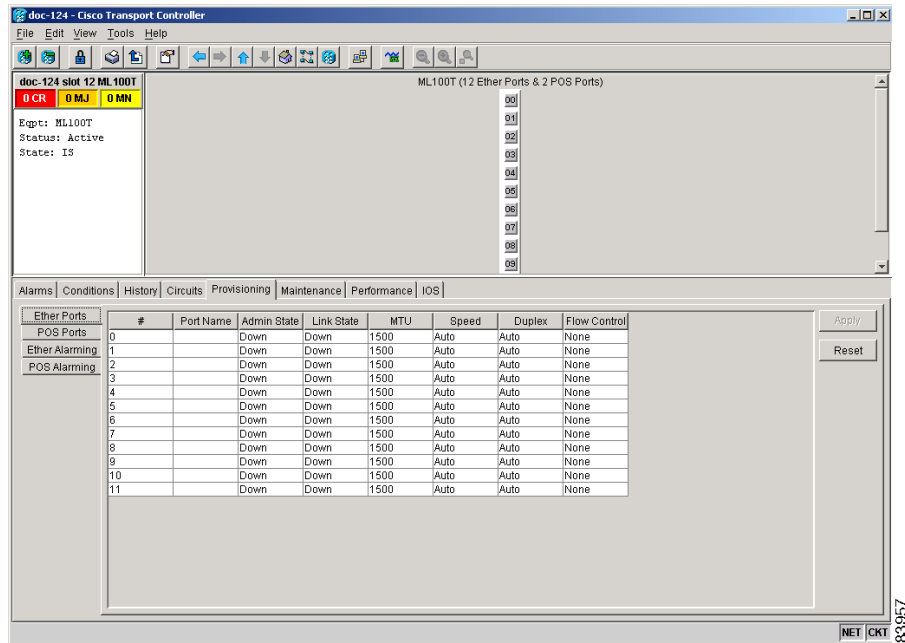
Table 2-4 POS Parameters (continued)

| Parameter | Meaning |
|------------------------|---|
| Rx Normal Packets | Number of packets between the minimum and maximum packet size received |
| Rx Shorts | Number of packets below the minimum packet size received |
| Rx Runts | Total number of frames received that are less than 64 bytes in length and have a CRC error |
| Rx Longs | Counter for the number of received frames that exceed the maximum valid packet length of 1518 bytes |
| Rx Total Errors | Total number of receive errors |
| Rx Crc Errors | Number of packets with a CRC error |
| Rx Input Drop Packets | Number of received packets dropped on input |
| Rx Input Abort Packets | Number of received packets aborted on input |
| Tx Pre Hdlc Bytes | Number of bytes transmitted prior to the bytes undergoing HLDC encapsulation by the policy engine |
| Tx Post Hdlc Bytes | Number of bytes transmitted after the bytes undergoing HLDC encapsulation by the policy engine |
| Tx Packets | Number of packets transmitted since the last counter reset |
| Port Drop Counts | Number of received frames dropped at the port level |

Displaying ML-Series Ethernet Ports Provisioning Information on CTC

The Ethernet port provisioning window displays the provisioning status of the Ethernet ports ([Figure 2-3](#)). Click the **Provisioning > Ether Ports** tabs to display this window. For ML-Series cards, only the Port Name field can be provisioned from CTC. The user must configure ML-Series ports using the Cisco IOS CLI.

Figure 2-3 Displaying ML-Series Ethernet Port Provisioning Information



The Provisioning > Ether Ports tab displays the following information:

- Port Name—Configurable identifier for the port.
- Admin State—Configured port state, which is administratively active or inactive. Possible values are UP and DOWN.
- Link State—Status between signalling points at port and attached device. Possible values are UP and DOWN.
- MTU—(maximum transfer unit) Largest acceptable packet size configured for that port. Default value is 1500.
- Speed—ML1000-2 possible values are Auto or 1 Gbps. ML100T-12 possible values are Auto, 10Mbps or 100Mbps.
- Duplex—Setting of the port. ML1000-2 possible values are Auto or Full. ML100T-12 possible values are Auto, Full, or Half.
- Flow Control—Negotiated flow control mode. Possible values are None, Symmetrical, and Asymmetrical.
- Optics—Small form-factor pluggable (SFP) physical media type. Possible values are Unplugged, 1000 SX, or 1000 LX. (This information does not apply to the ML100T-12 card.)

**Note**

Auto indicates the port is set to auto-negotiate capabilities with the attached link partner.

**Note**

The port name field configured in CTC and the port name configured in Cisco IOS are independent of each other. The name for the same port under Cisco IOS and CTC will not match, unless the same name is used to configure the port name in both CTC and Cisco IOS.

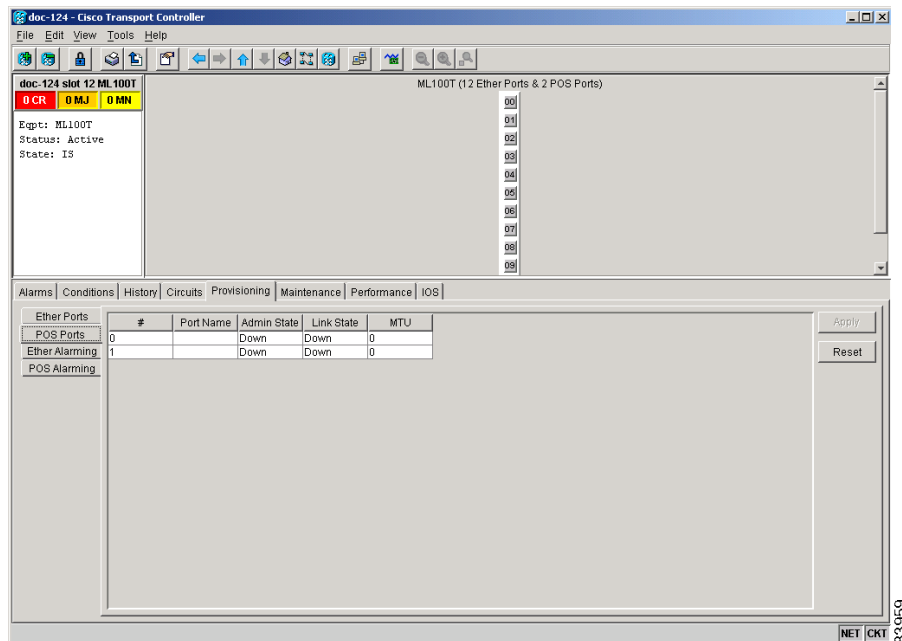
**Note**

When set to auto-negotiate, the ML1000-2 might show Auto in the speed and duplex columns of the Ether ports provisioning screen. This indicates that the ML1000-2 is set to auto-negotiate flow control with the link partner. It does not mean the speed or duplex mode of the card will vary from the 1-Gbps, full duplex characteristics of Gigabit Ethernet.

Displaying ML-Series POS Ports Provisioning Information on CTC

The POS ports provisioning window displays the provisioning status of the card's POS ports (Figure 2-4). Click the **Provisioning > POS Ports** tabs to display this window. For ML-Series cards, only the POS Port Name field can be provisioned from CTC. The user must configure ML-Series ports through the Cisco IOS CLI.

Figure 2-4 Displaying POS Port Provisioning Information



The Provisioning > POS Ports tab displays the following information:

- Port Name—Configurable identifier for the port.
- Admin State—Configured administrative port state, which is active or inactive. Possible values are UP and DOWN. For the UP value to appear, a POS port must be both administratively active and have a SONET/SDH circuit provisioned.
- Link State—Status between signaling points at the port and an attached device. Possible values are UP and DOWN.
- MTU—(maximum transfer unit) Largest acceptable packet size configured for that port. Maximum setting is 9000 and default size is 1500 for LEX encapsulation and 4470 for PPP and HDLC encapsulation. The MTU value is 0 until the POS port is used in creating a circuit.

**Note**

POS interfaces are first created when a CTC STS/STM circuit is provisioned.

**Note**

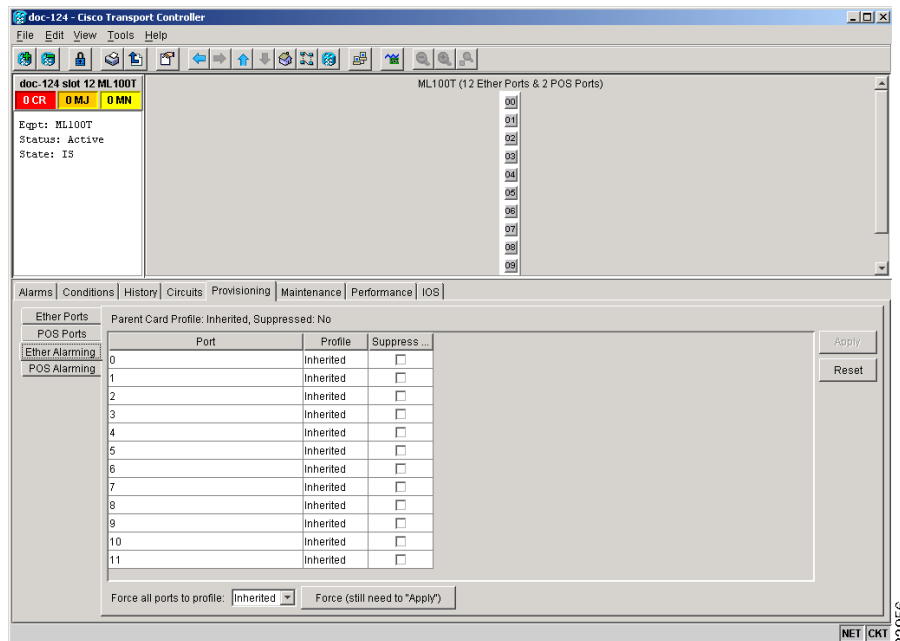
The port name field configured in CTC and the port name configured in Cisco IOS are independent of each other. The name for the same port under Cisco IOS and CTC will not match, unless the same name is used to configure the port name in both CTC and Cisco IOS.

Managing SONET/SDH Alarms

CTC manages the ML-Series SONET/SDH alarm behavior in the same manner as it manages alarm behavior for other ONS 15454 SONET/SDH cards. Refer to the “Manage Alarms” chapter of the *Cisco ONS 15454 Procedure Guide* or the *Cisco ONS 15454 SDH Procedure Guide* for detailed information. For information on specific alarms, refer to the “Alarm Troubleshooting” chapter of the *Cisco ONS 15454 Troubleshooting Guide* or *Cisco ONS 15454 SDH Troubleshooting Guide* for detailed information.

To view the window, click the **Ether Alarming > Provisioning** tabs for the Ethernet ports or **POS Alarming > Provisioning** tabs for the POS ports. [Figure 2-5](#) shows the Ethernet ports alarming pane.

Figure 2-5 Managing ML-Series SONET/SDH Alarms



SONET/SDH Circuit Provisioning

CTC provisions STS/STM level circuits for the two virtual SONET/SDH ports of the ML Series in the same manner as it provisions other ONS 15454 SONET/SDH OC-N cards. For the ONS 15454 SONET, refer to the “Create Circuits and VT Tunnels” chapter of the *Cisco ONS 15454 Procedure Guide* to create ML-Series STS circuits. For the ONS 15454 SDH, refer to the “Create Circuits and Low-Order Tunnels” chapter of the *Cisco ONS 15454 SDH Procedure Guide* to create ML-Series SDH circuits.



Initial Configuration

This chapter describes the initial configuration of the ML-Series card and contains the following major sections:

- [Hardware Installation, page 3-1](#)
- [ML-Series Access, page 3-1](#)
- [Startup Configuration File, page 3-6](#)
- [Cisco IOS Command Modes, page 3-10](#)
- [Using the Command Modes, page 3-12](#)

Hardware Installation

This section lists hardware installation tasks, including booting up the ML-Series card. Because ONS 15454 SONET/SDH card slots may be preprovisioned for an ML-Series line card, the following steps can be performed before or after the provisioning of the slot has taken place.

- Install the ML-Series card into the ONS 15454 SONET/SDH. Refer to Chapter 2, “Install Cards and Fiber-Optic Cable” of the *Cisco ONS 15454 Procedure Guide* or *Cisco ONS 15454 SDH Procedure Guide* for information.
- Connect the Ethernet cables to the ML-Series card.
- Connect the console terminal to the ML-Series card (optional).



Caution

Do not attempt to modify the boot system parameters of the ML-Series card.



Note

A NO-CONFIG condition is reported in CTC under the Alarms pane when an ML-Series card is inserted and no valid Cisco IOS startup configuration file exists. Loading or creating this file clears the condition. See the “[Startup Configuration File](#)” section on [page 3-6](#) for information on loading or creating the file.

ML-Series Access

There are four ways to access the ML-Series card Cisco Internet Operating System (IOS) configuration: opening a Cisco IOS session on CTC, telnetting to the IP Address and slot number plus 2000, Telnetting to a configured management port, or directly connecting to the console port.

Opening a Cisco IOS Session Using CTC

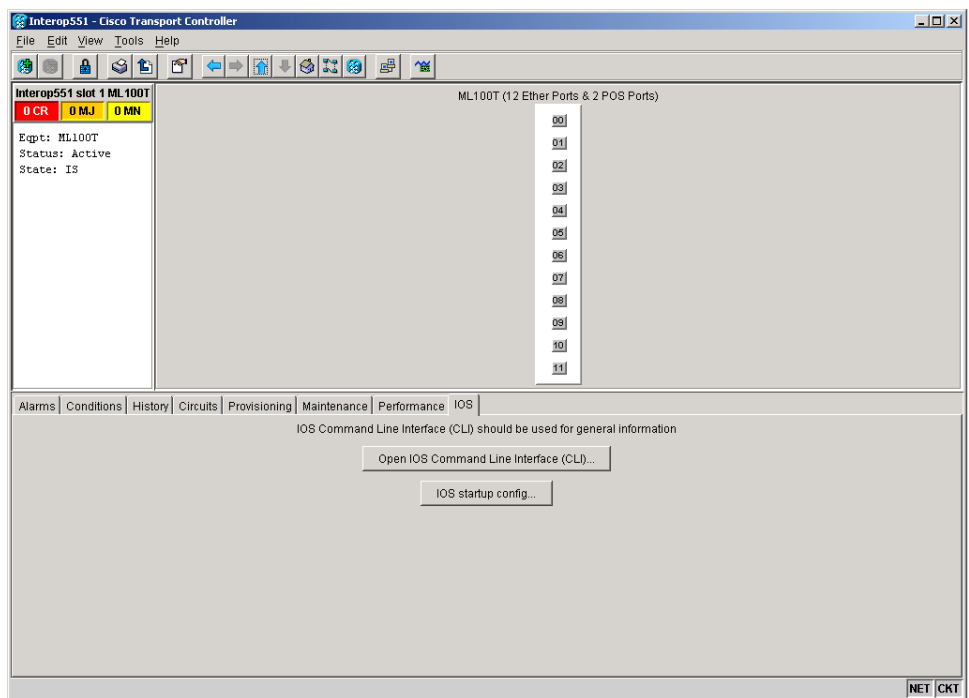
Users can initiate a Cisco IOS CLI session for the ML-Series card using CTC, the ONS 15454 SONET/SDH GUI. Click the **IOS** tab at the card-level CTC view, then click the **Open IOS Command Line Interface (CLI)** button (Figure 3-1). A window opens and a standard Cisco IOS CLI User EXEC command mode prompt appears.



Note

An IOS startup configuration file must be loaded and the ML-Series card must be installed and initialized prior to opening a Cisco IOS CLI session on CTC. See the “[Startup Configuration File](#)” section on page 3-6 for more information.

Figure 3-1 CTC IOS Window



Telnetting to the Node IP Address and Slot Number

Users can telnet to the Cisco IOS CLI using the IP address and the slot number of the ONS 15454 SONET/SDH plus 2000.



Note

An IOS startup configuration file must be loaded and the ML-Series card must be installed and initialized prior to telnetting to the IP address and slot number plus 2000. See the “[Startup Configuration File](#)” section on page 3-6 for more information.

**Note**

If the ONS 15454 SONET/SDH node is set up as a proxy server, where one ONS 15454 SONET/SDH node in the ring acts as a gateway network element (GNE) for the other nodes in the ring, telnetting over the GNE firewall to the IP address and slot number of a non-GNE or end network element (ENE) requires the user's telnet client to be SOCKS v5 aware (RFC 1928). Configure the Telnet client to recognize the GNE as the Socks v5 proxy for the telnet session and to recognize the ENE as the host.

- Step 1** Obtain the node IP address from the LCD on the front of the physical ONS 15454 SONET/SDH or the IP Addr field shown at the CTC node view (Figure 3-2).
- Step 2** Identify the slot number containing the targeted ML-Series card from either the physical ONS 15454 SONET/SDH or the CTC node view (Figure 3-2). For example, Slot 13.

Figure 3-2 CTC Node View Showing IP Address and Slot Number

Node IP address

The screenshot shows the CTC Node View for node doc-124. The left pane displays the following information:

```

doc-124
0 CR 0 MJ 0 MN
IP Addr : 10.92.18.124
Booted  : 1/9/03 12:53 PM
User    : CISCO15
Authority : Superuser
SW Version: 04.00-003A-09.03
Defaults : Factory Defaults
  
```

The right pane shows a rack of 12 slots. The status of each slot is shown as 'Act' (Active) or 'Sby' (Standby). The table below shows a list of alarms with columns for Num, Ref, New, Date, Object, Eqpt Type, Slot, Port, Sev, ST, SA, and Cond.

| Num | Ref | New | Date | Object | Eqpt Type | Slot | Port | Sev | ST | SA | Cond |
|-----|-----|-----|------|--------|-----------|------|------|-----|----|----|------|
| | | | | | | | | | | | |

- Step 3** Use the IP address and the total of the slot number plus 2000 as the Telnet address in your preferred communication program. For example, for an IP address of 10.92.18.124 and Slot 13, you would enter or telnet 10.92.18.124 2013.

Telnetting to a Management Port

Users can access the ML-Series through a standard Cisco IOS management port in the same manner as other Cisco IOS platforms. For further details about configuring ports and lines for management access, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.

As a security measure, the virtual type terminal (vty) lines used for telnet access are not fully configured. In order to gain telnet access to the ML-Series card, you must configure the vty lines via the serial console connection or preload a startup-configuration file that configures the vty lines. A port on the ML-Series must first be configured as the management port; see “[Configuring the Management Port](#)” section on page 3-8 or the “[Loading a Cisco IOS Startup Configuration File Through CTC](#)” section on page 3-9.

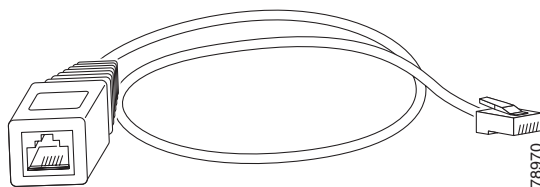
ML-Series IOS CLI Console Port

The ML-Series card has an RJ-11 serial console port on the card faceplate labeled CONSOLE. The console port is wired as data circuit-terminating equipment (DCE). It enables communication from the serial port of a PC or workstation running terminal emulation software to the Cisco IOS command line interface on a specific ML-Series card.

RJ-11 to RJ-45 Console Cable Adapter

Due to space limitations on the ML-Series card faceplate, the console port is an RJ-11 modular jack instead of the more common RJ-45 modular jack. Cisco supplies an RJ-11 to RJ-45 console cable adapter (P/N 15454-CONSOLE-02) with each ML-Series card. After connecting the adapter, the console port functions like the standard Cisco RJ-45 console port. [Figure 3-3](#) shows the RJ-11 to RJ-45 console cable adapter.

Figure 3-3 Console Cable Adapter



[Table 3-1](#) shows the mapping of the RJ-11 pins to the RJ-45 pins.

Table 3-1 RJ-11 to RJ-45 Pin Mapping

| RJ-11 Pin | RJ-45 Pin |
|-----------|-----------|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| None | 5 |
| 5 | 6 |

Table 3-1 RJ-11 to RJ-45 Pin Mapping (continued)

| RJ-11 Pin | RJ-45 Pin |
|-----------|-----------|
| None | 7 |
| 6 | 8 |

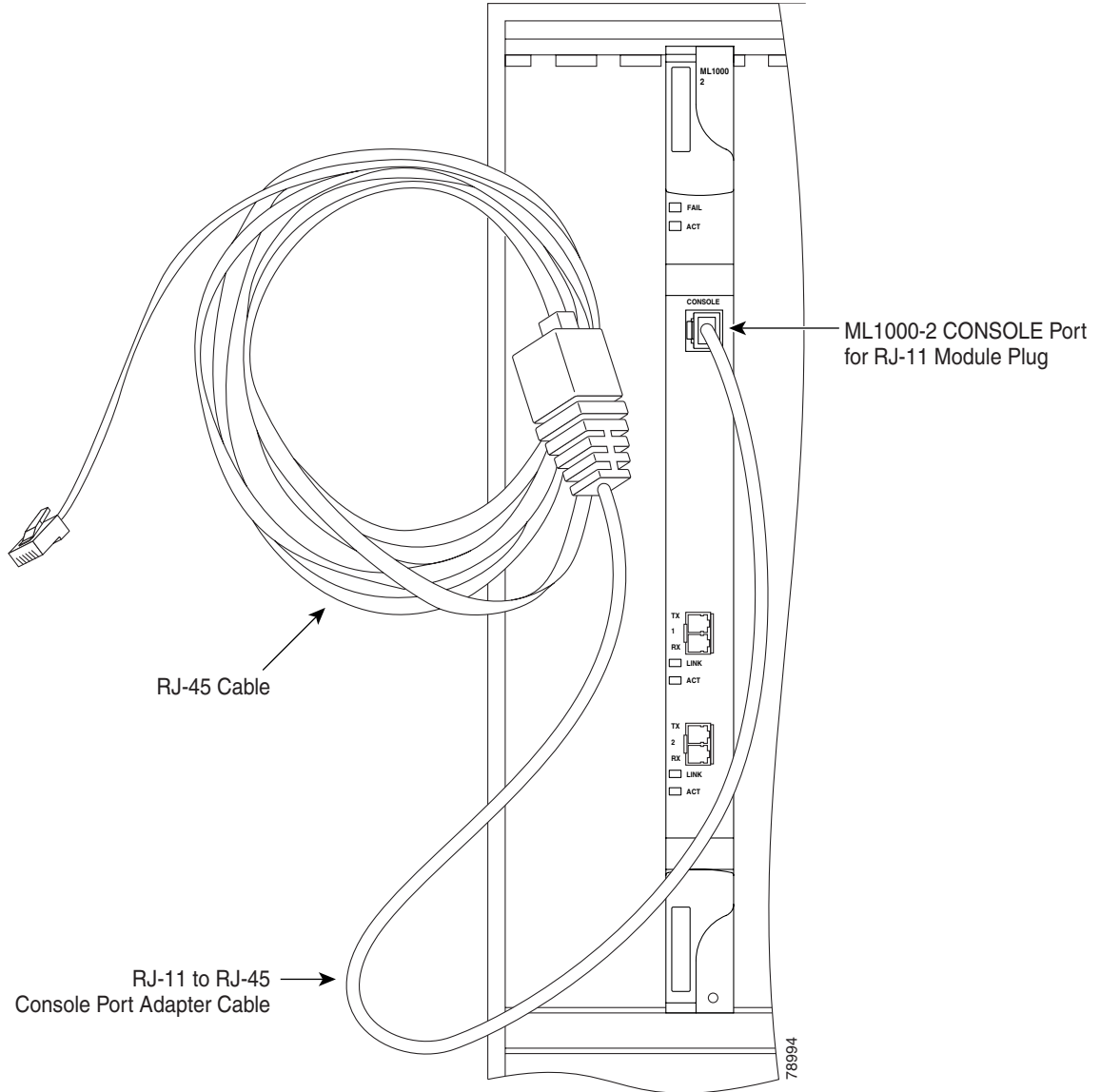
Connecting a PC or Terminal to the Console Port

Use the supplied cable, an RJ-11 to RJ-45 console cable adapter, and a DB-9 adapter to connect a PC to the ML-Series console port.

The PC must support VT100 terminal emulation. The terminal-emulation software—frequently a PC application such as Hyperterminal or Procomm Plus—makes communication between the ML-Series and your PC or terminal possible during the setup program.

-
- Step 1** Configure the data rate and character format of the PC or terminal to match these console port default settings:
- 9600 baud
 - 8 data bits
 - 1 stop bit
 - No parity
- Step 2** Insert the RJ-45 connector of the supplied cable into the female end of the supplied console cable adapter.
- Step 3** Insert the RJ-11 modular plug end of the supplied console cable adapter into the RJ-11 serial console port, labeled CONSOLE, on the ML-Series card faceplate. [Figure 3-4](#) shows the ML1000-2 faceplate with console port. The console port on the ML100-12 is at the bottom of the card faceplate.

Figure 3-4 Connecting to the Console Port



- Step 4** Attach the supplied RJ-45-to-DB-9 female DTE adapter to the nine-pin DB-9 serial port on the PC.
- Step 5** Insert the other end of the supplied cable in the attached adapter.

Startup Configuration File

The ML-Series card needs a startup configuration file in order to configure itself beyond the default configuration when the card is reset. If no startup configuration file exists in the TCC+/TCC2 flash memory, then the card will boot up to a default configuration. It will not be possible to establish a Telnet connection to the card until a startup configuration file is loaded onto the ML-Series card. Access to

the card can only be achieved via the console port. Users can manually set up the startup configuration file through the serial console port and the Cisco IOS CLI configuration mode or load a Cisco IOS supplied sample startup configuration file through CTC.

**Note**

The ML-Series card does not allow users to access the read-only memory monitor mode (ROMMON). The ML-Series card ROMMON is preconfigured to boot the correct Cisco IOS software image for the ML-Series card.

**Note**

When the running configuration file is altered, a RUNCFG-SAVENEED condition appears in CTC. This condition is a reminder to enter a **copy running-config startup-config** command in the Cisco IOS CLI, or the changes will be lost, when the ML-Series card reboots.

Manually Creating a Startup Configuration File Through the Serial Console Port

Configuration through the serial console port is familiar to those who have worked with other products using Cisco IOS. At the end of the configuration procedure, the **copy running-config startup-config** command will save a startup configuration file.

The serial console port gives the user visibility to the entire booting process of the ML-Series card. During initialization the ML-Series card first checks for a locally, valid cached copy of IOS. It will then either download the Cisco IOS software image from the TCC+/TCC2 or proceed directly to decompressing and initializing the image. Following Cisco IOS initialization the CLI prompt appears, at which time the user can enter the Cisco IOS CLI configuration mode and setup the basic ML-Series configuration.

Passwords

There are two types of passwords that you can configure for an ML-Series card: an enable password and an enable secret password. For maximum security, make the enable password different from the enable secret password.

- Enable password—The enable password is a non-encrypted password. It can contain any number of uppercase and lowercase alphanumeric characters. Give the enable password only to users permitted to make configuration changes to the ML-Series card.
- Enable secret password—The enable secret password is a secure, encrypted password. By setting an encrypted password, you can prevent unauthorized configuration changes. On systems running Cisco IOS software, you must enter the enable secret password before you can access global configuration mode.

An enable secret password can contain from 1 to 25 uppercase and lowercase alphanumeric characters. The first character cannot be a number. Spaces are valid password characters. Leading spaces are ignored; trailing spaces are recognized.

Passwords are configured in the next section, “[Configuring the Management Port.](#)”

Configuring the Management Port

Because there is no separate management port on ML-Series cards, any Fast Ethernet interface (0-11 on the ML100T-12 card), any Gigabit Ethernet interface (0-1 on the ML1000-2 card), or any POS interface (0-1 on either ML-Series card) can be configured as a management port. For the POS interface to exist, an STS or STM circuit must first be created through CTC or TL1.

You can remotely configure the ML-Series card through the management port, but first you must configure an IP address so that the ML-Series card is reachable or load a startup configuration file. You can manually configure the management port interface from the Cisco IOS command line interface (CLI) via the serial console connection.

To configure Telnet for remote management access, perform the following procedure, beginning in user EXEC mode:

| | Command | Purpose |
|----------------|--|---|
| Step 1 | Router> enable Router# | Activates user EXEC (or enable) mode. The # prompt indicates enable mode. |
| Step 2 | Router# configure terminal Router(config)# | Activates global configuration mode. You can abbreviate the command to confi g t . The Router(config)# prompt indicates that you are in global configuration mode. |
| Step 3 | Router(config)# enable password <i>password</i> | Sets the enable password. See the “ Passwords ” section on page 3-7. |
| Step 4 | Router(config)# enable secret <i>password</i> | Allows you to enter an enable secret password. See the “ Passwords ” section on page 3-7. A user must enter the enable secret password to gain access to global configuration mode. |
| Step 5 | Router(config)# interface <i>type number</i> Router(config-if)# | Activates interface configuration mode on the interface. |
| Step 6 | Router(config-if)# ip address <i>ip-address subnetmask</i> | Allows you to enter the IP address and IP subnet mask for the interface specified in Step 5. |
| Step 7 | Router(config-if)# no shutdown | Enables the interface. |
| Step 8 | Router(config-if)# exit Router(config)# | Returns to global configuration mode. |
| Step 9 | Router(config)# line vty <i>line-number</i> Router(config-line)# | Activates line configuration mode for virtual terminal connections. Commands entered in this mode control the operation of Telnet sessions to the ML-Series card. |
| Step 10 | Router(config-line)# password <i>password</i> | Allows you to enter a password for Telnet sessions. |
| Step 11 | Router(config-line)# end Router# | Returns to privileged EXEC mode. |
| Step 12 | Router# copy running-config startup-config | (Optional) Saves your configuration changes to NVRAM. |

After you have completed configuring remote management on the management port, you can use Telnet to remotely assign and verify configurations.

Configuring the Hostname

In addition to the system passwords and enable password, your initial configuration should include a hostname to easily identify your ML-Series card. To configure the hostname, perform the following task, beginning in enable mode:

| | Command | Purpose |
|--------|--|--|
| Step 1 | Router# configure terminal Router(config)# | Activates global configuration mode. |
| Step 2 | Router(config)# hostname <i>name-string</i> | Allows you to enter a system name. In this example, we set the hostname to "Router." |
| Step 3 | Router(config)# end Router# | Returns to privileged EXEC mode. |
| Step 4 | Router# copy running-config startup-config | (Optional) Copies your configuration changes to NVRAM. |

Loading a Cisco IOS Startup Configuration File Through CTC

CTC allows a user to load the startup configuration file required by the ML-Series card. A Cisco-supplied sample IOS startup configuration file, named Basic-IOS-startup-config.txt, is available on the Cisco ONS 15454 SONET/SDH software CD. CISCO15 is the IOS CLI default line password and the enable password for this configuration. Users can also create their own startup configuration file, see the ["Manually Creating a Startup Configuration File Through the Serial Console Port"](#) section on page 3-7.

CTC can load a Cisco IOS startup configuration file into the TCC+/TCC2 card flash before the ML-Series card is physically installed in the slot. When installed, the ML-Series card downloads and applies the Cisco IOS software image and the preloaded Cisco IOS startup-configuration file. Preloading the startup configuration file allows an ML-Series card to immediately operate as a fully configured card when inserted into the ONS 15454 SONET/SDH.

If the ML-Series card is booted up prior to the loading of the Cisco IOS startup configuration file into TCC+/TCC2 card flash, then the ML-Series card must be reset to use the Cisco IOS startup configuration file or the user can issue the command **copy start run** at the Cisco IOS CLI to configure the ML-Series card to use the Cisco IOS startup configuration file.

This procedure details the initial loading of a Cisco IOS Startup Configuration file through CTC.

-
- Step 1** At the card-level view of the ML-Series card, click the **IOS** tab.
The CTC IOS window appears ([Figure 3-1 on page 3-2](#)).
- Step 2** Click the **IOS startup config** button.
The config file dialog appears.
- Step 3** Click the **Local -> TCC** button.

- Step 4** The sample IOS startup configuration file can be installed from either the ONS 15454 SONET/SDH software CD or from a PC or network folder:
- To install the Cisco supplied startup config file from the ONS 15454 SONET/SDH software CD, insert the CD into the CD drive of the PC or workstation. Using the CTC config file dialog, navigate to the CD drive of the PC or workstation and double-click the *Basic-IOS-startup-config.txt* file.
 - To install the Cisco supplied config file from a PC or network folder, navigate to the folder containing the desired IOS startup config file and double-click the desired IOS startup config file.
- Step 5** At the Are you sure? dialog box, click the **Yes** button.
- The Directory and Filename fields on the configuration file dialog update to reflect that the IOS startup config file is loaded onto the TCC+/TCC2.
- Step 6** Load the IOS startup config file from the TCC+/TCC2 to the ML-Series card:
- a. If the ML-Series card has already been installed, right-click on the ML-Series card at the node level CTC view and select reset card.
- After the reset, the ML-Series card runs under the newly-loaded IOS startup config.
- b. If the ML-Series card is not yet installed, installing the ML-Series card into the slot will load and run the newly loaded IOS startup configuration on the ML-Series card.

**Note**

When the Cisco IOS startup configuration file is downloaded and parsed at initialization, if there is an error in the parsing of this file, an ERROR-CONFIG alarm is reported and appears under the CTC alarms pane or in TL1. No other Cisco IOS error messages regarding the parsing of text are reported to the CTC or in TL1. An experienced Cisco IOS user can locate and troubleshoot the line in the startup configuration file that produced the parsing error by opening the Cisco IOS CLI and entering a **copy start run** command.

**Note**

A standard ONS 15454 SONET/SDH database restore reinstalls the IOS startup config file on the TCC+/TCC2, but does not implement the IOS startup config on the ML-Series. Complete [Step 6](#) to load the IOS startup config file from the TCC+/TCC2 to the ML-Series card.

Cisco IOS Command Modes

The Cisco IOS user interface has several different modes. The commands available to you depend on which mode you are in. To get a list of the commands available in a given mode, type a question mark (?) at the system prompt.

[Table 3-2 on page 3-11](#) describes the most commonly used modes, how to enter the modes, and the resulting system prompts. The system prompt helps you identify which mode you are in and, therefore, which commands are available to you.

**Note**

Router or Switch is used as a generic prompt in documentation. Your specific prompt will vary.

Table 3-2 IOS Command Modes

| Mode | What You Use It For | How to Access | Prompt |
|---|---|---|----------------------|
| User EXEC | Connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and display system information. | Log in. | Router> |
| Privileged EXEC (also called Enable mode) | Set operating parameters. The privileged command set includes the commands in user EXEC mode, as well as the configure command. Use this command mode to access the other command modes. | From user EXEC mode, enter the enable command and the enable password. | Router# |
| Global configuration | Configure features that affect the system as a whole. | From privileged EXEC mode, enter the configure terminal command. | Router(config)# |
| Interface configuration | Enable features for a particular interface. Interface commands enable or modify the operation of a Fast Ethernet, Gigabit Ethernet or Packet over SONET (POS) port. | From global configuration mode, enter the interface type number command. For example, enter interface fastethernet 0 for Fast Ethernet or interface gigabitethernet 0 for Gigabit Ethernet interfaces or interface pos 0 for Packet over SONET interfaces. | Router(config-if)# |
| Line configuration | Configure the console port or VTY line from the directly connected console or the virtual terminal used with Telnet. | From global configuration mode, enter the line console 0 command to configure the console port or the line vty line-number command to configure a VTY line. | Router(config-line)# |

When you start a session on the ML-Series card, you begin in user EXEC mode. Only a small subset of the commands are available in user EXEC mode. To have access to all commands, you must enter privileged EXEC mode, also called Enable mode. From privileged EXEC mode, you can type in any EXEC command or access global configuration mode. Most of the EXEC commands are single-use commands, such as **show** commands, which show the current configuration status, and **clear** commands, which clear counters or interfaces. The EXEC commands are not saved across reboots of the ML-Series card.

The configuration modes allow you to make changes to the running configuration. If you later save the configuration, these commands are stored across ML-Series card reboots. You must start in global configuration mode. From global configuration mode, you can enter interface configuration mode, subinterface configuration mode, and a variety of protocol-specific modes.

Read-only memory monitor mode (ROMMON) is a separate mode used when the ML-Series card cannot boot properly. For example, your ML-Series card might enter ROM monitor mode if it does not find a valid system image when it is booting, or if its configuration file is corrupted at startup.

Using the Command Modes

The Cisco IOS command interpreter, called the EXEC, interprets and executes the commands you enter. You can abbreviate commands and keywords by entering just enough characters to make the command unique from other commands. For example, you can abbreviate the **show** command to **sh** and the **configure terminal** command to **conf t**.

Exit

When you type **exit**, the ML-Series card backs out one level. In general, typing **exit** returns you to global configuration mode. Enter **end** to exit configuration mode completely and return to privileged EXEC mode.

Getting Help

In any command mode, you can get a list of available commands by entering a question mark (?).

```
Router> ?
```

To obtain a list of commands that begin with a particular character sequence, type in those characters followed immediately by the question mark (?). Do not include a space. This form of help is called word help, because it completes a word for you.

```
Router# co?
configure
```

To list keywords or arguments, enter a question mark in place of a keyword or argument. Include a space before the question mark. This form of help is called command syntax help, because it reminds you which keywords or arguments are applicable based on the command, keywords, and arguments you have already entered.

```
Router#configure ?
memory          Configure from NV memory
network         Configure from a TFTP network host
overwrite-network Overwrite NV memory from TFTP network host
terminal        Configure from the terminal
<cr>
```

To redisplay a command you previously entered, press the Up Arrow key. You can continue to press the Up Arrow key to see more of the previously issued commands.



Tip

If you are having trouble entering a command, check the system prompt, and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

You can press **Ctrl-Z** or type **end** in any mode to immediately return to privileged EXEC (enable) mode, instead of entering **exit**, which returns you to the previous mode.



Configuring Interfaces

This chapter describes the basic interface configuration for the ML-Series card to help you get your ML-Series card up and running. For more information about the Cisco IOS commands used in this chapter, refer to the *Cisco IOS Command Reference* publication.

This chapter contains the following major sections:

- [Interface Configuration, page 4-1](#)
- [Instructions for Configuring Interfaces, page 4-3](#)
- [Understanding Interfaces, page 4-4](#)
- [POS on the ML-Series Card, page 4-8](#)
- [Configuring the ML-Series POS Interfaces, page 4-12](#)
- [Common ML-Series POS Configurations, page 4-17](#)



Note

Complete the initial configuration of your ML-Series card before proceeding with configuring interfaces.

Interface Configuration

The main function of the ML-Series card is to relay packets from one data link to another. Consequently, you must configure the characteristics of the interfaces, which receive and send packets. Interface characteristics include, but are not limited to, IP address, address of the port, data encapsulation method, and media type.

Many features are enabled on a per-interface basis. Interface configuration mode contains commands that modify the interface operation (for example, of an Ethernet port). When you enter the **interface** command, you must specify the interface type and number.

The following general guidelines apply to all physical and virtual interface configuration processes:

- All interfaces have a name which is comprised of an interface type (word) and a Port ID (number). For example, FastEthernet 2.
- Configure each interface with a bridge-group or IP address and IP subnet mask.
- VLANs are supported through the use of subinterfaces. The subinterface is a logical interface configured separately from the associated physical interface.
- Each physical interface, and the internal Packet-over-SONET/SDH (POS) interfaces, have an assigned MAC address.

MAC Addresses

Every port or device that connects to an Ethernet network needs a MAC address. Other devices in the network use MAC addresses to locate specific ports in the network and to create and update routing tables and data structures.

To find MAC addresses for a device, use the **show interfaces** command, as follows:

```
Router# sh interfaces fastEthernet 0
FastEthernet0 is up, line protocol is up
  Hardware is epif_port, address is 0005.9a39.6634 (bia 0005.9a39.6634)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, Auto Speed, 100BaseTX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:18, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    11 packets input, 704 bytes
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog, 11 multicast
    0 input packets with dribble condition detected
    3 packets output, 1056 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
```

Interface Port ID

The interface port ID designates the physical location of the interface within the ML-Series card. It is the name that you use to identify the interface you are configuring. The system software uses interface port IDs to control activity within the ML-Series card and to display status information. Interface port IDs are not used by other devices in the network; they are specific to the individual ML-Series card and its internal components and software.

The ML100T-12 port IDs for the 12 Fast Ethernet interfaces are Fast Ethernet 0 through 11. The ML1000-2 port IDs for the two Gigabit Ethernet interfaces are Gigabit Ethernet 0 and 1. Both ML-Series cards feature two POS ports, and the ML-Series port IDs for the two POS interfaces are POS 0 and 1. You can use user-defined abbreviations such as f0 through f11 to configure the 12 Fast Ethernet interfaces, gi0 or gi1 to configure the two Gigabit Ethernet interfaces, and POS0 and POS1 to configure the two POS ports.

You can use Cisco IOS **show** commands to display information about any or all the interfaces of the ML-Series card.



Caution

Do not use the g0 or g1 for a Gigabit Ethernet user-defined abbreviation. This will create an unsupported group asynchronous interface.

Instructions for Configuring Interfaces

The following general configuration instructions apply to all interfaces. Before you configure interfaces, develop a plan for a bridge or routed network.

To configure an interface, do the following:



Note

Router or Switch is used as a generic prompt in documentation. Your specific prompt will vary.

- Step 1** Enter the **configure EXEC** command at the privileged EXEC prompt to enter global configuration mode.

```
Router> enable
Password:
Router# configure terminal
Router(config)#
```

- Step 2** Enter the **interface** command, followed by the interface type (for example, fastethernet, gigabitethernet, or pos), and its interface port ID (see the “[Interface Port ID](#)” section on page 4-2).

For example, to configure a Gigabit Ethernet port, enter this command:

```
Router(config)# interface gigabit-ethernet-number
```

- Step 3** Follow each **interface** command with the interface configuration commands required for your particular interface.

The commands you enter define the protocols and applications that will run on the interface. The ML-Series card collects and applies commands to the **interface** command until you enter another **interface** command or a command that is not an interface configuration command. You can also enter **end** to return to privileged EXEC mode.

- Step 4** Check the status of the configured interface by entering the EXEC **show interface** command.

```
Router# sh interface fastEthernet 0
FastEthernet0 is up, line protocol is up
Hardware is epif_port, address is 0005.9a39.6634 (bia 0005.9a39.6634)
MTU 1500 bytes, BW 100000 Bit, DLY 100 use,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, Auto Speed, 100BaseTX
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:01, output 00:00:18, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue :0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 11 packets input, 704 bytes
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 watchdog, 11 multicast
  0 input packets with dribble condition detected
  3 packets output, 1056 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out
```

Understanding Interfaces

ML-Series cards support Fast Ethernet, Gigabit Ethernet and POS interfaces. This section provides some examples of configurations for all interface types.

To configure an IP address or bridge-group number on a Fast Ethernet, Gigabit Ethernet, or POS interface, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router(config)# interface <i>type number</i> | Activates interface configuration mode to configure either the Gigabit Ethernet interface, the Fast Ethernet interface or the POS interface. |
| Step 2 | Router(config-if)# { ip address <i>ip-address subnet-mask</i> / bridge-group <i>bridge-group-number</i> } | Sets the IP address and IP subnet mask to be assigned to the interface. or Assigns a network interface to a bridge group. |
| Step 3 | Router(config-if)# no shutdown | Enables the interface by preventing it from shutting down. |
| Step 4 | Router(config)# end | Returns to privileged EXEC mode. |
| Step 5 | Router# copy running-config startup-config | (Optional) Saves configuration changes to timing and control card (TCC+/TCC2) Flash database. |



Note

Repeat Steps 1 through 3 to configure the other interfaces on the ML-Series card.

Configuring the Fast Ethernet Interfaces (ML100T-12)

To configure the IP address or bridge-group number, autonegotiation, and flow control on a Fast Ethernet interface, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|---|
| Step 1 | Router(config)# interface fastethernet <i>number</i> | Activates interface configuration mode to configure the Fast Ethernet interface. |
| Step 2 | Router(config-if)# { ip address <i>ip-address subnet-mask</i> / bridge-group <i>bridge-group-number</i> } | Sets the IP address and IP subnet mask to be assigned to the interface. or Assigns a network interface to a bridge group. |

| | Command | Purpose |
|--------|---|---|
| Step 3 | Router(config-if)# {no} speed [10 100 auto] | Configures the transmission speed for 10 or 100 Mbps. If you set the speed or duplex for auto , you enable autonegotiation on the system—the ML-Series card matches the speed and duplex mode of the partner node. |
| Step 4 | Router(config-if)# {no} duplex {full half auto} | for full duplex, half duplex, or autonegotiate. |
| Step 5 | Router(config-if)# flowcontrol send {on off desired} | (Optional) Sets the send flow control value for an interface. Flow control works only with port-level policing. |
| Step 6 | Router(config-if)# no shutdown | Enables the interface by preventing it from shutting down. |
| Step 7 | Router(config)# end | Returns to privileged EXEC mode. |
| Step 8 | Router# copy running-config startup-config | (Optional) Saves your configuration changes to TCC+/TCC2 Flash database. |

The following example shows how to do the initial configuration of a Fast Ethernet interface with an IP address, autonegotiated speed, and autonegotiated duplex:

Example 4-1 Initial Configuration of a Fast Ethernet Interface

```
Router(config)# interface fastethernet 1
Router(config-if)# ip address 10.1.2.4 255.0.0.0
Router(config-if)# speed auto
Router(config-if)# duplex auto
Router(config-if)# no shutdown
Router(config-if)# end
Router# copy running-config startup-config
```

Configuring the Gigabit Ethernet Interface (ML1000-2)

To configure IP address or bridge-group number, autonegotiation, and flow control on a Gigabit Ethernet interface, perform the following procedure, beginning in global configuration mode:



Note The default setting for the negotiation mode is **auto** for the Gigabit Ethernet and Fast Ethernet interfaces. The Gigabit Ethernet port always operates at 1000 Mbps in full-duplex mode.

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router# interface gigabitethernet <i>number</i> | Activates interface configuration mode to configure the Gigabit Ethernet interface. |
| Step 2 | Router#(config-if)# {ip address ip-address subnet-mask bridge-group bridge-group-number} | Sets the IP address and subnet mask. or Assigns a network interface to a bridge group. |

| | Command | Purpose |
|--------|---|--|
| Step 3 | Router#(config-if)# [no] negotiation auto | Sets negotiation mode to auto . The Gigabit Ethernet port attempts to negotiate the link with the partner port. If you want the port to force the link up no matter what the partner port setting is, set the Gigabit Ethernet interface to no negotiation auto . |
| Step 4 | Router(config-if)# flowcontrol {send receive} {on off desired} | (Optional) Sets the send or receive flow control value for an interface. Flow control works only with port-level policing. |
| Step 5 | Router#(config-if)# no shutdown | Enables the interface by preventing it from shutting down. |
| Step 6 | Router#(config)# end | Returns to privileged EXEC mode. |
| Step 7 | Router# copy running-config startup-config | (Optional) Saves configuration changes to TCC+/TCC2 Flash database. |



Note Repeat Steps 1 to 4 to configure the other Gigabit Ethernet interfaces.

The following example shows how to do an initial configuration of a Gigabit Ethernet interface with autonegotiation and an IP address:

Example 4-2 Initial Configuration of a Gigabit Ethernet Interface

```
Router(config)# interface gigabitethernet 0
Router(config-if)# ip address 10.1.2.3 255.0.0.0
Router(config-if)# negotiation auto
Router(config-if)# no shutdown
Router(config-if)# end
Router# copy running-config startup-config
```

Monitoring Operations on the Fast Ethernet and Gigabit Ethernet Interfaces

To verify the settings after you have configured Fast Ethernet interfaces, enter the **show interface** command.

The following output from the **show interface** command displays the status of the Fast Ethernet interface including port speed and duplex operation:

Example 4-3 show interface command output

```
Router# show interface fastEthernet 0
FastEthernet0 is up, line protocol is up
Hardware is epif_port, address is 0005.9a39.6634 (bia 0005.9a39.6634)
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 100Mb/s, 100BaseTX
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output 00:00:23, output hang never
```

```

Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  0 packets input, 0 bytes
Received 0 broadcasts (0 IP multicast)
  0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 watchdog, 0 multicast
  0 input packets with dribble condition detected
  4 packets output, 1488 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out

```

Enter the **show controller** command to display information about the Fast Ethernet controller chip.

The following output from the **show controller** command shows statistics, including information about initialization block information, transmit ring, receive ring, and errors:

Example 4-4 *show controller command output*

```

Router#show controller fastEthernet 0
IF Name: FastEthernet0
Port Status DOWN
Send Flow Control      : Disabled
Receive Flow Control  : Enabled
MAC registers
CMCR : 0x0000042D (Tx Enabled, Rx Disabled)
CMPR : 0x150B0A80 (Long Frame Disabled)
FCR  : 0x0000A00B (Rx Pause detection Enabled)
MII registers:
Control Register          (0x0): 0x4000 (Auto negotiation disabled)
Status Register          (0x1): 0x7809 (Link status Down)
PHY Identification Register 1 (0x2): 0x40
PHY Identification Register 2 (0x3): 0x61D4
Auto Neg. Advertisement Reg (0x4): 0x1E1 (Speed 100, Duplex Full)
Auto Neg. Partner Ability Reg (0x5): 0x0 (Speed 10, Duplex Half)
Auto Neg. Expansion Register (0x6): 0x4
100Base-X Aux Control Reg (0x10): 0x2000
100Base-X Aux Status Register(0x11): 0x0
100Base-X Rcv Error Counter (0x12): 0x0
100Base-X False Carr. Counter(0x13): 0x0

```

Enter the **show run interfaces fastEthernet 0** command to display information about the configuration of the Fast Ethernet interface. The command is useful when there are multiple interfaces and you want to look at the configuration of a specific interface.

The following output from the **show controller** command includes the IP or lack of IP address and the state of the interface:

Example 4-5 *show controller command output*

```

daytona#show run interface fastEthernet 0
Building configuration...

Current configuration : 56 bytes
!
interface FastEthernet0

```

```

no ip address
shutdown

end

```

POS on the ML-Series Card

Packet over SONET/SDH (POS) is a high-speed method of transporting IP traffic between two points. This technology combines the Point-to-Point Protocol (PPP) with SONET and Synchronous Digital Hierarchy (SDH) interfaces. SONET is an octet-synchronous multiplex scheme defined by the American National Standards Institute (ANSI) standard (T1.105.1988) for optical digital transmission, and SDH is the European Telecommunications Standards Institute (ETSI) equivalent.

ML-Series SONET/SDH Transmission Rates

SONET transmission rates are integral multiples of 51.840 Mbps. [Table 4-1](#) shows supported transmission multiples.

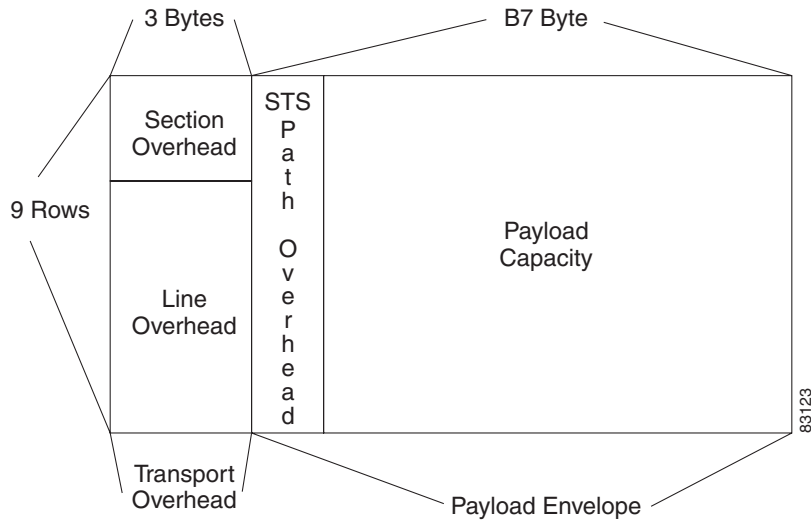
Table 4-1 *Transmission Multiples Supported by ML-Series Cards*

| Topology | Supported Sizes |
|---|--|
| Circuits terminated by two ML-Series cards | STS-1, STS-3c, STS-6c, STS-9c, STS-12c, and STS-24c (SONET) or VC4, VC4-2c, VC4-3c, VC4-4c, and VC4-8c (SDH) |
| Circuits terminated by G-Series card and ML-Series card | STS-1, STS-3c, STS-6c, STS-9c, STS-12c (SONET) or VC4, VC4-2c, VC4-3c, VC4-4c, and VC4-8c (SDH) |
| Circuits terminated by ML-Series card and External POS device | STS-3c and STS-12c (SONET) or VC4 and VC4-4c (SDH) |

SONET Frame Fundamentals

SONET is a Layer 1 protocol that uses a layered architecture. [Figure 4-1](#) shows SONET's three layers: section, line, and path. The section overhead (SOH) and line overhead (LOH) form the transport overhead (TOH), while the path overhead (POH) and actual payload (referred to as payload capacity) form the synchronous payload envelope (SPE). Each layer adds a number of overhead bytes to the SONET frame.

Figure 4-1 Three SONET Layers



C2 Byte

One of the overhead bytes in the SONET frame is the C2 Byte. The SONET standard defines the C2 byte as the path signal label. The purpose of this byte is to communicate the payload type being encapsulated by the SONET framing overhead (FOH). The C2 byte functions similarly to EtherType and Logical Link Control (LLC)/Subnetwork Access Protocol (SNAP) header fields on an Ethernet network; it allows a single interface to transport multiple payload types simultaneously. [Table 4-2](#) provides C2 byte hex values.

Table 4-2 C2 Byte Common Values

| Hex Value | SONET Payload Contents |
|-----------|--|
| 00 | Unequipped |
| 01 | Equipped non specific payload |
| 02 | Virtual Tributaries (VTs) inside (default) |
| 03 | VTs in locked mode (no longer supported) |
| 04 | Asynchronous DS3 mapping |
| 12 | Asynchronous DS-4NA mapping |
| 13 | Asynchronous Transfer Mode (ATM) cell mapping |
| 14 | Distributed Queue Dual Bus (DQDB) protocol cell mapping |
| 15 | Asynchronous Fiber Distributed Data Interface (FDDI) mapping |
| 16 | IP inside PPP with scrambling |
| CF | IP inside PPP without scrambling |
| FE | Test signal mapping (see ITU-T G.707) |

C2 Byte and Scrambling

As listed in [Table 4-2](#), POS interfaces use a value of 0x16 or 0xCF in the C2 byte depending on whether ATM-style scrambling is enabled or not. RFC 2615, which defines PPP over SONET, mandates the use of these values based on the scrambling setting. The RFC defines the C2 byte values as follows: “the value of 22 (16 hex) is used to indicate PPP with X⁴³+ 1 scrambling [4]. For compatibility with RFC 1619 (STS-3c-SPE/VC-4 only), if scrambling has been configured to be off, then the value 207 (CF hex) is used for the Path Signal Label to indicate PPP without scrambling.”

In other words:

- If scrambling is enabled, POS interfaces use a C2 value of 0x16 (PPP and HDLC encapsulation).
- If scrambling is disabled, POS interfaces use a C2 value of 0xCF (PPP and HDLC encapsulation).
- LEX encapsulation uses a C2 value of 0x01 regardless of the scrambling setting.

Most POS interfaces that use a default C2 value of 0x16 (22 decimal) insert the **pos flag c2 22** command in the configuration, although this line does not appear in the running configuration since it is the default. Use the **pos flag c2** command to change the value from its default, as follows:

Example 4-6 pos flag c2 command

```
Router(config-if)# pos flag c2 ?
<0-255> byte value, default 0x16
```



Note

Changing the C2 value from the default value does not affect POS scrambling settings.

Use the **show run** command to confirm your change. The **show controller pos** command outputs the receive and transmit values and the C2 value. Thus, changing the value on the local end will not change the value in the **show controller** command output.

Example 4-7 show controller pos command

```
Router# sh controllers pos 0
Interface POS0
Hardware is Packet/Ethernet over Sonet
PATH
  PAIS      = 0          PLOP      = 0          PRDI      = 0          PTIM      = 0
  PPLM      = 0          PUNEQ    = 0          PPDI      = 0
  BER_SF_B3 = 0          BER_SD_B3 = 0          BIP(B3)   = 14         REI       = 155
  NEWPTR    = 0          PSE      = 0          NSE       = 0

Active Alarms : None
Demoted Alarms: None
Active Defects: None
Alarms reportable to TCC/CLI: PAIS PRDI PLOP PUNEQ PPLM PTIM PPDI BER_SF_B3 BER_
SD_B3
Link state change defects: PAIS PLOP PRDI PPDI BER_SF_B3
Link state change time : 200 (msec)

DOS FPGA channel number: 0
Starting STS (0 based) : 0
Circuit size           : STS-24c
RDI Mode                : 1 bit
C2 (tx / rx)           : 0x01 / 0x01
Framing                 : SONET

Path Trace
```

```

Mode          : off
Buffer        : Unstable
Remote hostname :
Remote interface:
Remote IP addr  :

```

```

B3 BER thresholds:
SFBER = 1e-5,   SDBER = 1e-7

```

```

1106 total input packets, 80059 post-HDLC bytes
0 input short packets, 80714 pre-HDLC bytes
0 input long packets, 205 input runt packets
17 input CRCerror packets, 0 input drop packets
0 input abort packets
1107 input packets dropped by ucode

```

```

0 total output packets, 0 output pre-HDLC bytes
0 output post-HDLC bytes

```

```
Carrier delay is 200 msec
```

The **show interface pos0** command shows scrambling.

```

daytona#show interface pos0
POS0 is up, line protocol is up
Hardware is Packet/Ethernet over Sonet, address is 0005.9a3b.bf90 (bia 0005.9a3b.bf90)
MTU 1500 bytes, BW 1244160 Kbit, DLY 100 usec,
    reliability 243/255, txload 1/255, rxload 166/255
Encapsulation ONS15454-G1000, crc 32, loopback not set
Keepalive set (10 sec)
Scramble enabled
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 2385314109 bytes
    Received 0 broadcasts (0 IP multicast)
    0 runts, 0 giants, 0 throttles
    0 parity
    2839625 input errors, 2839625 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
    9 packets output, 3393 bytes, 0 underruns
    0 output errors, 0 applique, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions

```

Third-Party POS Interfaces

If a Cisco POS interface fails to come up when connected to a third-party device, confirm the scrambling and cyclic redundancy check (CRC) settings as well as the advertised value in the C2 byte. On routers from Juniper Networks, configuring RFC 2615 mode sets the following three parameters:

- Scrambling enabled

- C2 value of 0x16
- CRC-32

Previously, when scrambling was enabled, these third-party devices continued to use a C2 value of 0xCF, which did not properly reflect the scrambled payload.

Configuring the ML-Series POS Interfaces

To configure the POS interface, perform the following procedure, beginning in global configuration mode. Encapsulation changes on POS ports are allowed only when the interface is in a manual shutdown (ADMIN_DOWN):

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router(config)# interface pos <i>number</i> | Activates interface configuration mode to configure the POS interface. The POS interface is created upon the creation of a SONET/SDH circuit. |
| Step 2 | Router#(config-if)# { ip address <i>ip-address</i> <i>subnet-mask</i> / bridge-group <i>bridge-group-number</i> } | Sets the IP address and subnet mask. or Assigns a network interface to a bridge group. |
| Step 3 | Router#(config-if)# shutdown | Manually shuts down the interface. Encapsulation changes on POS ports are allowed only when the interface is shutdown (ADMIN_DOWN). |
| Step 4 | Router#(config-if)# encapsulation <i>type</i> | Sets the encapsulation type. Valid values are: <ul style="list-style-type: none"> • hdlc—Cisco HDLC • lex—LAN extension, special encapsulation for use with Cisco ONS G-Series Ethernet line cards • ppp—Point-to-Point Protocol |
| Step 5 | Router#(config-if)# pos flag c2 <i>byte value</i> | (Optional) Sets the C2 byte value. Valid choices are 0 to 255 (decimal). The default value is 0x01 (hex) for LEX. |
| Step 6 | Router#(config-if)# no shutdown | Restarts the shutdown interface. |
| Step 7 | Router#(config)# end | Returns to privileged EXEC mode. |
| Step 8 | Router# copy running-config startup-config | (Optional) Saves configuration changes to NVRAM. |



Note

The POS interface is not present until a SONET STS or SDH STM circuit is created.

Monitoring Operations on the POS Interface and POS Controller

The following output from the **show interface** command displays the POS interface's status and global parameters:

Example 4-8 show interface command

```
Router# show interface pos 0
POS0 is up, line protocol is up
  Hardware is Packet/Ethernet over Sonet, address is 0005.9a39.6630 (bia 0005.9a
39.6630)
  MTU 1500 bytes, BW 311040 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ONS15454-G1000, crc 32, loopback not set
  Keepalive set (10 sec)
  Scramble enabled
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:02:34, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    1107 packets input, 11267427 bytes
    Received 0 broadcasts (0 IP multicast)
    0 runs, 0 giants, 0 throttles
      0 parity
    1 input errors, 1 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 applique, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
```

The following output from the **show controllers** command displays the POS controllers:

Example 4-9 show controllers command

```
Router# show controllers pos 0
Interface POS0
Hardware is Packet/Ethernet over Sonet
PATH
  PAIS      = 1          PLOP      = 0          PRDI      = 0          PTIM = 0
  PPLM      = 0          PUNEQ    = 0          PPDI      = 0
  BER_SF_B3 = 0          BER_SD_B3 = 0          BIP(B3) = 2975      REI = 7
  NEWPTR    = 1          PSE      = 0          NSE      = 0

Active Alarms : None
Demoted Alarms: None
Active Defects: None
Alarms reportable to CLI: PAIS PRDI PLOP PUNEQ PPLM PTIM PPDI BER_SF_B3 BER_
3
Link state change defects: PAIS PLOP PRDI PPDI BER_SF_B3
Link state change time   : 200 (msec)

DOS FPGA channel number: 0
Starting STS (0 based)  : 0
Circuit size            : STS-6c
```

```

RDI Mode           : 1 bit
C2 (tx / rx)      : 0x01 / 0x01
Framing           : SONET

Path Trace
Mode              : off
Buffer            : Unstable
Remote hostname   :
Remote interface:
Remote IP addr    :

B3 BER thresholds:
SFBER = 1e-5,    SDBER = 1e-7

1107 total input packets, 11267259 post-HDLC bytes
0 input short packets, 11267427 pre-HDLC bytes
0 input long packets , 0 input runt packets
1 input CRCError packets , 0 input drop packets
0 input abort packets
945 input packets dropped by ucode

0 total output packets, 0 output pre-HDLC bytes
0 output post-HDLC bytes

Carrier delay is 200 msec

```

Additional Configurations

To configure additional properties to match those of the interface at the far end, perform the following steps, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|---|
| Step 1 | Router(config-if)# no keepalive | Turns off keep alive messages. Keep alive messages, though not required, are recommended. |
| Step 2 | Router(config-if)# crc {16 32} | Sets the CRC value. If the device to which the POS module is connected does not support the default CRC value of 32, set both devices to use a value of 16. |

Setting the MTU Size

To set the maximum transmission unit (MTU) size, perform the following steps, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|---|
| Step 1 | Router(config)# interface pos number | Enters interface configuration mode and specifies the POS interface to configure. |
| Step 2 | Router(config-if)# mtu bytes | Configures the MTU size up to a maximum of 9000 bytes. See Table 4-3 on page 4-15 . |

Table 4-3 shows the default MTU sizes.

Table 4-3 Default MTU Size

| Encapsulation Type | Default Size |
|--------------------|--------------|
| LEX (default) | 1500 |
| HDLC | 4470 |
| PPP | 4470 |

Configuring Framing

No Cisco IOS configuration is necessary. Framing type is determined during circuit configuration.

Configuring POS SPE Scrambling

To configure POS SPE scrambling, perform the following steps, beginning in global configuration mode:

| | Command | Purpose |
|---------------|---|--|
| Step 1 | Router(config)# interface pos number | Enters interface configuration mode and specifies the POS interface to configure. |
| Step 2 | Router(config-if)# no pos scramble-spe | Disables payload scrambling on the interface. Payload scrambling is on by default. |
| Step 3 | Router(config-if)# no shutdown | Enables the interface with the previous configuration. |

SONET/SDH Alarms

The ML-Series cards report SONET/SDH alarms under both Cisco IOS and CTC/TL1. A number of path alarms are reported in the Cisco IOS console. Configuring Cisco IOS console alarm reporting has no effect on CTC alarm reporting. The “[Configuring SONET/SDH Alarms](#)” procedure specifies the alarms reported to the Cisco IOS console.

CTC/TL1 has sophisticated SONET/SDH alarm reporting capabilities. As a card in the ONS node, the ML-Series card reports alarms to CTC/TL-1 like any other ONS card. On the ONS 15454 SONET, the ML-Series card reports Telcordia GR-253 SONET alarms in the Alarms panel of CTC. For more information on alarms and alarm definitions, refer to the “Alarm Troubleshooting” chapter of the *Cisco ONS 15454 Troubleshooting Guide*, or the *Cisco ONS 15454 SDH Troubleshooting Guide*.

Configuring SONET/SDH Alarms

All SONET/SDH alarms are logged on the Cisco IOS CLI by default. But to provision or disable the reporting of specific SONET/SDH alarms on the Cisco IOS CLI, perform the following steps beginning in global configuration mode:

| | Command | Purpose |
|--------|--|--|
| Step 1 | Router(config)# interface pos number | Enters interface configuration mode and specifies the POS interface to configure. |
| Step 2 | Router(config-if)# pos report { all encap pais plop ppdi pplm prdi ptim puneq sd-ber-b3 sf-ber-b3 } | Permits logging of selected SONET/SDH alarms. Use the no form of the command to disable reporting of a specific alarm. The alarms are as follows: <ul style="list-style-type: none"> • all—All alarms/signals • encap—Path encapsulation mismatch • pais—Path alarm indication signal • plop—Path loss of pointer • ppdi—Path payload defect indication • pplm—Payload label, C2 mismatch • prdi—Path remote defect indication • ptim—Path trace identifier mismatch • puneq—Path label equivalent to zero • sd-ber-b3—PBIP BER in excess of SD threshold • sf-ber-b3—PBIP BER in excess of SF threshold |
| Step 3 | Router(config-if)# end | Returns to the privileged EXEC mode. |
| Step 4 | Router# copy running-config startup-config | (Optional) Saves configuration changes to NVRAM. |

To determine which alarms are reported on the POS interface and to display the bit error rate (BER) thresholds, use the **show controllers pos** command.



Note Cisco IOS alarm reporting commands apply only to the Cisco IOS CLI. SONET/SDH alarms reported to the CTC are not affected.

To configure path alarms as triggers and specify a delay, perform the following steps beginning in global configuration mode:

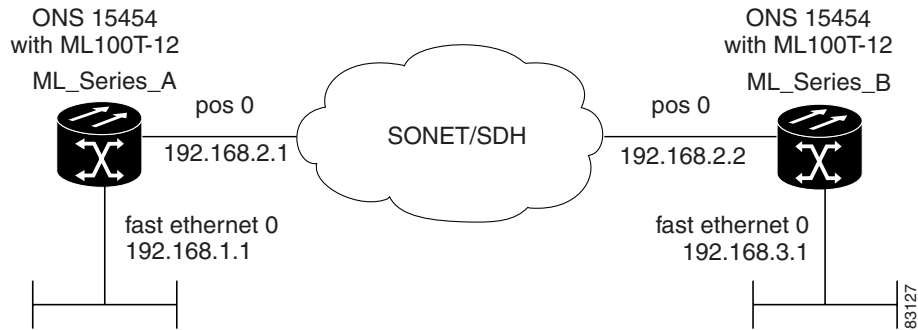
| | Command | Purpose |
|--------|--|--|
| Step 1 | Router(config)# interface pos number | Enters interface configuration mode and specifies the POS interface to configure. |
| Step 2 | Router(config-if)# pos trigger defect {all ber_sf_b3 encap pais plop ppdi pplm prdi ptim puneq} | Configures certain path defects as triggers to bring down the POS interface. The configurable triggers are as follows: <ul style="list-style-type: none"> • all—All link down alarm failures • ber_sd_b3—PBIP BER in excess of SD threshold failure • ber_sf_b3—PBIP BER in excess of SD threshold failure (default) • encap—Path Signal Label Encapsulation Mismatch failure (default) • pais—Path Alarm Indication Signal failure (default) • plop—Path Loss of Pointer failure (default) • ppdi—Path Payload Defect Indication failure (default) • pplm—Payload label mismatch path (default) • prdi—Path Remote Defect Indication failure (default) • ptim—Path Trace Indicator Mismatch failure (default) • puneq—Path Label Equivalent to Zero failure (default) |
| Step 3 | Router(config-if)# pos trigger delay millisecond | Sets waiting period before the line protocol of the interface goes down. Delay can be set from 200 to 2000 ms. If no time intervals are specified, the default delay is set to 200 ms. |
| Step 4 | Router(config-if)# end | Returns to the privileged EXEC mode. |
| Step 5 | Router# copy running-config startup-config | (Optional) Saves configuration changes to NVRAM. |

Common ML-Series POS Configurations

The following sections describe common ML-Series card POS configurations.

ML-Series Card to ML-Series Card

Figure 4-2 illustrates a POS configuration between two ML-Series cards.

Figure 4-2 ML-Series Card to ML-Series Card POS Configuration**Example 4-10 Router A Configuration**

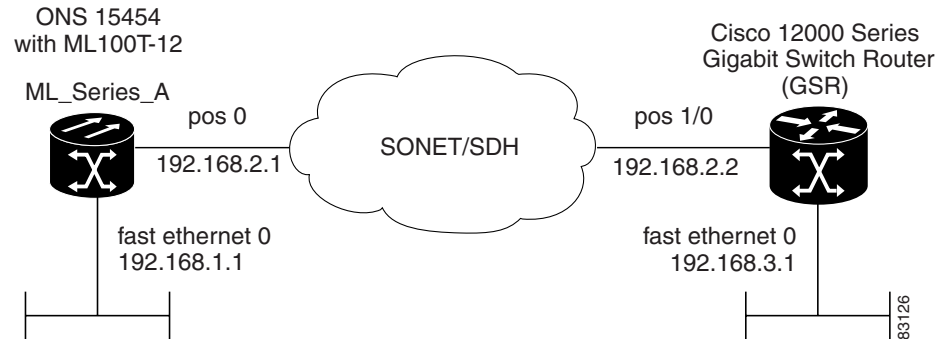
```
hostname Router_A
!
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
!
interface POS0
 ip address 192.168.2.1 255.255.255.0
 crc 32
 pos flag c2 1
!
router ospf 1
 log-adjacency-changes
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
```

Example 4-11 Router B Configuration

```
hostname Router_B
!
interface FastEthernet0
 ip address 192.168.3.1 255.255.255.0
!
interface POS0
 ip address 192.168.2.2 255.255.255.0
 crc 32
 pos flag c2 1
!
router ospf 1
 log-adjacency-changes
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.3.0 0.0.0.255 area 0
!
```

ML-Series Card to Cisco 12000 GSR-Series Router

Figure 4-3 illustrates a POS configuration between an ML-Series card and a Cisco 12000 GSR-Series router.

Figure 4-3 ML-Series Card to Cisco 12000 Series Gigabit Switch Router (GSR) POS Configuration**Example 4-12 Router A Configuration**

```
hostname Router_A
!
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
!
!
interface POS0
 ip address 192.168.2.1 255.255.255.0
 encapsulation ppp
 crc 32
!
router ospf 1
 log-adjacency-changes
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
```

Example 4-13 GSR-12000 Configuration

```
hostname GSR
!
interface FastEthernet1/0
 ip address 192.168.3.1 255.255.255.0
!
interface POS2/0
 ip address 192.168.2.2 255.255.255.0
 crc 32
 encapsulation PPP
 pos scramble-atm
!
router ospf 1
 log-adjacency-changes
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.3.0 0.0.0.255 area 0
!
```

**Note**

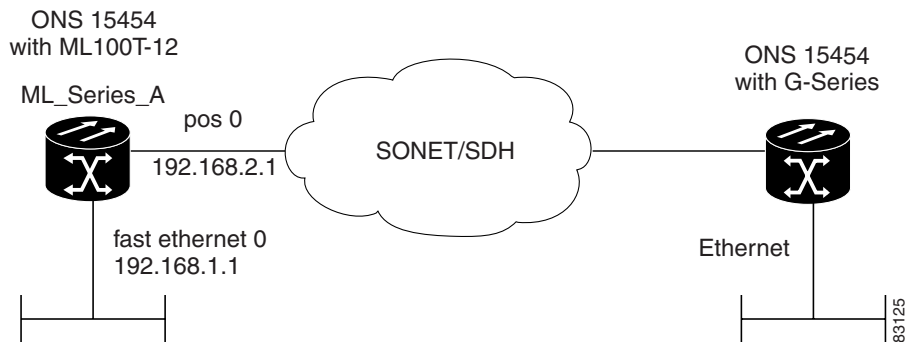
The default encapsulation for the ML-Series card is LEX and the corresponding default MTU is 1500 bytes. When connecting to an external POS device, it is important to ensure that both the ML-Series switch and the external device uses the same configuration for the parameters listed in [Table 4-4](#).

Table 4-4 ML-Series Parameter Configuration for Connection to a Cisco 12000 GSR-Series Router

| Command | Parameter |
|---|--|
| Router(config-if)# encapsulation ppp or Router(config-if)# encapsulation hdlc | Encapsulation—Default encapsulation is HDLC on GSR. Default encapsulation on ML-Series card is LEX. |
| Router(config-if)# show controller pos | C2 Byte—Use the show controller pos command to verify that the transmit and receive C2 values are the same. |
| Router(config-if)# pos flag c2 value | Sets the C2 byte value. Valid choices are 0 to 255 (decimal). The default value is 0x01 (hex) for LEX. |

ML-Series Card to G-Series Card

Figure 4-4 illustrates a POS configuration between an ML-Series card and a G-Series card.

Figure 4-4 ML-Series Card to G-Series Card POS Configuration

Example 4-14 Router A Configuration

```
hostname Router_A
!
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
!
interface POS0
 ip address 192.168.2.1 255.255.255.0
 crc 32
!
router ospf 1
 log-adjacency-changes
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
```




Configuring Bridging

This chapter describes how to configure bridging for the ML-Series card. For more information about the Cisco IOS commands used in this chapter, refer to the *Cisco IOS Command Reference* publication.

This chapter includes the following major sections:

- [Understanding Bridging, page 5-1](#)
- [Monitoring and Verifying Bridging, page 5-3](#)



Caution

Cisco Inter-Switch Link (ISL) and Cisco Dynamic Trunking Protocol (DTP) are not supported by the ML-Series, but the ML-Series broadcast forwards these formats. Using ISL or DTP on connecting devices is not recommended. Some Cisco devices attempt to use ISL or DTP by default.

Understanding Bridging

The ML-Series card can be configured to serve as an IP router and a bridge. Cisco IOS software supports transparent bridging for Fast Ethernet, Gigabit Ethernet, and POS. Cisco IOS software functionality combines the advantages of a spanning-tree bridge and a router. This combination provides the speed and protocol transparency of a spanning-tree bridge, along with the functionality, reliability, and security of a router.

To configure bridging, you must perform the following tasks in the modes indicated:

- In global configuration mode:
 - Enable bridging of IP packets.
 - Select the type of Spanning Tree Protocol.
- In interface configuration mode:
 - Determine which interfaces belong to the same bridge group.

These interfaces become part of the same spanning tree, allowing the ML-Series card to bridge all nonrouted traffic among the network interfaces comprising the bridge group. Interfaces not participating in a bridge group cannot forward bridged traffic.

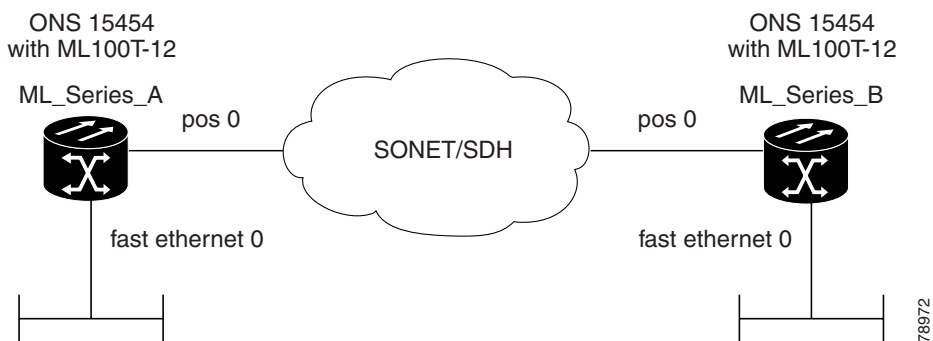
If the destination address of the packet is known in the bridge table, the packet is forwarded on a single interface in the bridge group. If the packet's destination is unknown in the bridge table, the packet is flooded on all forwarding interfaces in the bridge group. The bridge places source addresses in the bridge table as it learns them during the process of bridging.

A separate spanning-tree process runs for each configured bridge group. Each bridge group participates in a separate spanning tree. A bridge group establishes a spanning tree based on the bridge protocol data units (BPDUs) it receives on only its member interfaces.

Configuring Bridging

| | Command | Purpose |
|--------|--|---|
| Step 1 | Router(config)# no ip routing | Enables bridging of IP packets. This command needs to be executed once per card, not once per bridge-group. This step is not done for integrated routing and bridging (IRB). |
| Step 2 | Router(config)# bridge <i>bridge-group-number</i> protocol { <i>rstp</i> <i>ieee</i> } | Assigns a bridge group number and defines the appropriate spanning-tree type: either IEEE 802.1D Spanning Tree Protocol or IEEE 802.1W Rapid Spanning Tree. |
| Step 3 | Router(config)# bridge <i>bridge-group-number</i> priority <i>number</i> | (Optional) Assigns a specific priority to the bridge, to assist in the spanning-tree root definition. The lower the priority, the more likely the bridge is selected as the root. |
| Step 4 | Router(config)# interface <i>interface type interface number</i> | Enters interface configuration mode to configure the interface of the ML-Series card. |
| Step 5 | Router(config-if)# bridge-group <i>bridge-group-number</i> | Assigns a network interface to a bridge group. |
| Step 6 | Router(config-if)# no shutdown | Changes the shutdown state to up and enables the interface. |
| Step 7 | Router(config-if)# end | Returns to privileged EXEC mode. |
| Step 8 | Router# copy running-config startup-config | (Optional) Saves your entries in the configuration file. |

Example 5-1 Bridging Example



Example 5-2 Router A Configuration

```
bridge 1 protocol ieee
!
!
interface FastEthernet0
no ip address
bridge-group 1
```

```

!
interface POS0
  no ip address
  crc 32
  bridge-group 1
  pos flag c2 1

```

Example 5-3 Router B Configuration

```

bridge 1 protocol ieee
!
!
interface FastEthernet0
  no ip address
  bridge-group 1
!
interface POS0
  no ip address
  crc 32
  bridge-group 1
  pos flag c2 1

```

Monitoring and Verifying Bridging

After you have set up the ML-Series card for bridging, you can monitor and verify its operation by performing the following procedure, in privileged EXEC mode:

| | Command | Purpose |
|--------|--|---|
| Step 1 | Router# clear bridge <i>bridge-group-number</i> | Removes any learned entries from the forwarding database of a particular bridge group, clears the transmit, and receives counts for any statically configured forwarding entries. |
| Step 2 | Router# show bridge <i>bridge-group-number</i> <i>interface-address</i> | Displays classes of entries in the bridge forwarding database. |
| Step 3 | Router# show bridge verbose | Displays detailed information about configured bridge groups. |
| Step 4 | Router# show spanning-tree | Displays the spanning tree topology known to the ML-Series card. |

Example 5-4 Monitoring and Verifying Bridging

```

Router# show bridge

Total of 300 station blocks, 298 free
Codes: P - permanent, S - self

Bridge Group 1:

Maximum dynamic entries allowed: 1000
Current dynamic entry count: 2

      Address      Action  Interface
0000.0001.6000    forward FastEthernet0
0000.0001.6100    forward   POS0

```

Router# **show bridge verbose**

Total of 300 station blocks, 298 free
Codes: P - permanent, S - self

Maximum dynamic entries allowed: 1000
Current dynamic entry count: 2

| BG Hash | Address | Action | Interface | VC | Age | RX count | TX co |
|---------|----------------|---------|---------------|----|-----|----------|-------|
| 1 60/0 | 0000.0001.6000 | forward | FastEthernet0 | - | | | |
| 1 61/0 | 0000.0001.6100 | forward | POS0 | - | | | |

Flood ports
FastEthernet0
POS0

Router# **show spanning-tree**

Bridge group 1

Spanning tree enabled protocol ieee

Root ID Priority 32769
Address 0005.9a39.6634
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
Address 0005.9a39.6634
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300

| Interface | Role | Sts | Cost | Prio.Nbr | Type |
|-----------|------|-----|------|----------|------|
| Fa0 | Desg | FWD | 19 | 128.3 | P2p |
| PO0 | Desg | FWD | 9 | 128.20 | P2p |



Configuring STP and RSTP

This chapter describes the IEEE 802.1D Spanning Tree Protocol (STP) and the ML-Series implementation of the IEEE 802.1W Rapid Spanning Tree Protocol (RSTP). It also explains how to configure STP and RSTP on the ML-Series card.

This chapter consists of these sections:

- [STP Features, page 6-1](#)
- [RSTP, page 6-9](#)
- [Interoperability with 802.1D STP, page 6-14](#)
- [Configuring STP and RSTP Features, page 6-14](#)
- [Verifying and Monitoring STP and RSTP Status, page 6-19](#)

STP Features

These sections describe how these spanning-tree features work:

- [STP Overview, page 6-2](#)
- [Supported STP Instances, page 6-2](#)
- [Bridge Protocol Data Units, page 6-2](#)
- [Election of the Root Switch, page 6-3](#)
- [Bridge ID, Switch Priority, and Extended System ID, page 6-4](#)
- [Spanning-Tree Timers, page 6-4](#)
- [Creating the Spanning-Tree Topology, page 6-4](#)
- [Spanning-Tree Interface States, page 6-5](#)
- [Spanning-Tree Address Management, page 6-8](#)
- [STP and IEEE 802.1Q Trunks, page 6-8](#)
- [Spanning Tree and Redundant Connectivity, page 6-8](#)
- [Accelerated Aging to Retain Connectivity, page 6-9](#)

STP Overview

STP is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. Spanning-tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The spanning-tree algorithm calculates the best loop-free path throughout a switched Layer 2 network. Switches send and receive spanning-tree frames, called bridge protocol data units (BPDUs), at regular intervals. The switches do not forward these frames, but use the frames to construct a loop-free path.

Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages. Switches might also learn end-station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network.

Spanning tree defines a tree with a root switch and a loop-free path from the root to all switches in the Layer 2 network. Spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path.

When two interfaces on a switch are part of a loop, the spanning-tree port priority and path cost settings determine which interface is put in the forwarding state and which is put in the blocking state. The port priority value represents the location of an interface in the network topology and how well it is located to pass traffic. The path cost value represents media speed.

Supported STP Instances

The ML-Series card supports the per-VLAN spanning tree (PVST+) and a maximum of 255 spanning-tree instances.

Bridge Protocol Data Units

The stable, active, spanning-tree topology of a switched network is determined by these elements:

- Unique bridge ID (switch priority and MAC address) associated with each VLAN on each switch
- Spanning-tree path cost to the root switch
- Port identifier (port priority and MAC address) associated with each Layer 2 interface

When the switches in a network are powered up, each functions as the root switch. Each switch sends a configuration BPDU through all of its ports. The BPDUs communicate and compute the spanning-tree topology. Each configuration BPDU contains this information:

- Unique bridge ID of the switch that the sending switch identifies as the root switch
- Spanning-tree path cost to the root
- Bridge ID of the sending switch
- Message age
- Identifier of the sending interface
- Values for the hello, forward delay, and max-age protocol timers

When a switch receives a configuration BPDU that contains *superior* information (lower bridge ID, lower path cost, etc.), it stores the information for that port. If this BPDU is received on the root port of the switch, the switch also forwards it with an updated message to all attached LANs for which it is the designated switch.

If a switch receives a configuration BPDU that contains *inferior* information to that currently stored for that port, it discards the BPDU. If the switch is a designated switch for the LAN from which the inferior BPDU was received, it sends that LAN a BPDU containing the up-to-date information stored for that port. In this way, inferior information is discarded, and superior information is propagated on the network.

A BPDU exchange results in these actions:

- One switch in the network is elected as the root switch.
- A root port is selected for each switch (except the root switch). This port provides the best path (lowest cost) when the switch forwards packets to the root switch.
- The shortest distance to the root switch is calculated for each switch based on the path cost.
- A designated switch for each LAN segment is selected. The designated switch incurs the lowest path cost when forwarding packets from that LAN to the root switch. The port through which the designated switch is attached to the LAN is called the designated port.
- Interfaces included in the spanning-tree instance are selected. Root ports and designated ports are put in the forwarding state.
- All interfaces not included in the spanning tree are blocked.

Election of the Root Switch

All switches in the Layer 2 network participating in the spanning tree gather information about other switches in the network through an exchange of BPDU data messages. This exchange of messages results in these actions:

- Election of a unique root switch for each spanning-tree instance
- Election of a designated switch for every switched LAN segment
- Removal of loops in the switched network by blocking Layer 2 interfaces connected to redundant links

For each VLAN, the switch with the highest switch priority (the lowest numerical priority value) is elected as the root switch. If all switches are configured with the default priority (32768), the switch with the lowest MAC address in the VLAN becomes the root switch. The switch priority value occupies the most significant bits of the bridge ID.

When you change the switch priority value, you change the probability that the switch will be elected as the root switch. Configuring a higher value decreases the probability; a lower value increases the probability.

The root switch is the logical center of the spanning-tree topology in a switched network. All paths that are not needed to reach the root switch from anywhere in the switched network are placed in the spanning-tree blocking mode.

BPDU contains information about the sending switch and its ports, including switch and MAC addresses, switch priority, port priority, and path cost. Spanning tree uses this information to elect the root switch and root port for the switched network and the root port and designated port for each switched segment.

Bridge ID, Switch Priority, and Extended System ID

The IEEE 802.1D standard requires that each switch has a unique bridge identifier (bridge ID), which determines the selection of the root switch. Because each VLAN is considered as a different *logical bridge* with PVST+, the same switch must have as many different bridge IDs as VLANs configured on it. Each VLAN on the switch has a unique 8-byte bridge ID; the two most-significant bytes are used for the switch priority, and the remaining six bytes are derived from the switch MAC address.

The ML-Series card supports the 802.1T spanning-tree extensions, and some of the bits previously used for the switch priority are now used as the bridge ID. The result is that fewer MAC addresses are reserved for the switch, and a larger range of VLAN IDs can be supported, all while maintaining the uniqueness of the bridge ID. As shown in [Table 6-1](#), the two bytes previously used for the switch priority are reallocated into a 4-bit priority value and a 12-bit extended system ID value equal to the bridge ID. In earlier releases, the switch priority is a 16-bit value.

Table 6-1 Switch Priority Value and Extended System ID

| Switch Priority Value | | | | Extended System ID (Set Equal to the Bridge ID) | | | | | | | | | | | |
|-----------------------|--------|--------|--------|---|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit 16 | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 |
| 32768 | 16384 | 8192 | 4096 | 2048 | 1024 | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Spanning tree uses the extended system ID, the switch priority, and the allocated spanning-tree MAC address to make the bridge ID unique for each VLAN. With earlier releases, spanning tree used one MAC address per VLAN to make the bridge ID unique for each VLAN.

Spanning-Tree Timers

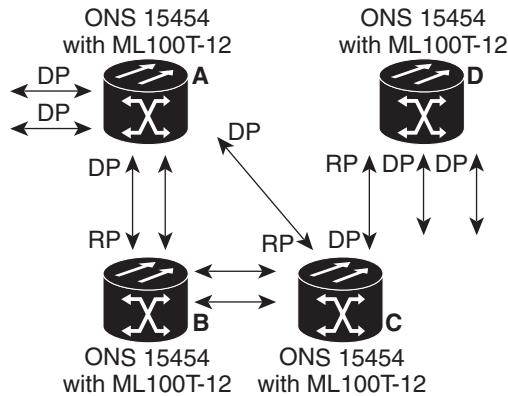
[Table 6-2](#) describes the timers that affect the entire spanning-tree performance.

Table 6-2 Spanning-Tree Timers

| Variable | Description |
|---------------------|--|
| Hello timer | When this timer expires, the interface sends out a Hello message to the neighboring nodes. |
| Forward-delay timer | Determines how long each of the listening and learning states last before the interface begins forwarding. |
| Maximum-age timer | Determines the amount of time the switch stores protocol information received on an interface. |

Creating the Spanning-Tree Topology

In [Figure 6-1](#), A is elected as the root switch because the switch priority of all the switches is set to the default (32768) and A has the lowest MAC address. However, because of traffic patterns, number of forwarding interfaces, or link types, A might not be the ideal root switch. By increasing the priority (lowering the numerical value) of the ideal switch so that it becomes the root switch, you force a spanning-tree recalculation to form a new topology with the ideal switch as the root.

Figure 6-1 Spanning-Tree Topology

RP = root port
DP = designated port

83803

When the spanning-tree topology is calculated based on default parameters, the path between source and destination end stations in a switched network might not be ideal. For instance, connecting higher-speed links to an interface that has a higher number than the root port can cause a root-port change. The goal is to make the fastest link the root port.

Spanning-Tree Interface States

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When an interface transitions directly from nonparticipation in the spanning-tree topology to the forwarding state, it can create temporary data loops. Interfaces must wait for new topology information to propagate through the switched LAN before starting to forward frames. They must allow the frame lifetime to expire for forwarded frames that have used the old topology.

Each Layer 2 interface on a switch using spanning tree exists in one of these states:

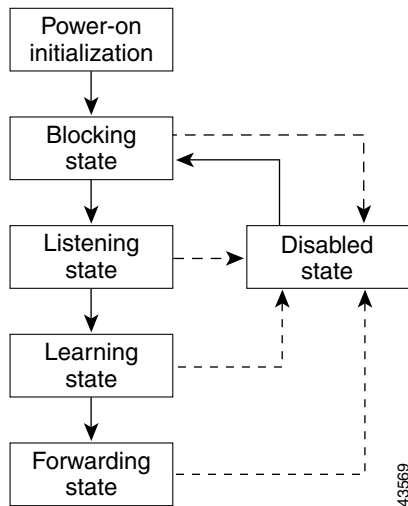
- Blocking—The interface does not participate in frame forwarding.
- Listening—The first transitional state after the blocking state when the spanning tree determines that the interface should participate in frame forwarding.
- Learning—The interface prepares to participate in frame forwarding.
- Forwarding—The interface forwards frames.
- Disabled—The interface is not participating in spanning tree because of a shutdown port, no link on the port, or no spanning-tree instance running on the port.

An interface moves through these states:

1. From initialization to blocking
2. From blocking to listening or to disabled
3. From listening to learning or to disabled
4. From learning to forwarding or to disabled
5. From forwarding to disabled

Figure 6-2 illustrates how an interface moves through the states.

Figure 6-2 Spanning-Tree Interface States



When you power up the switch, STP is enabled by default, and every interface in the switch, VLAN, or network goes through the blocking state and the transitory states of listening and learning. Spanning tree stabilizes each interface at the forwarding or blocking state.

When the spanning-tree algorithm places a Layer 2 interface in the forwarding state, this process occurs:

1. The interface is in the listening state while spanning tree waits for protocol information to transition the interface to the blocking state.
2. While spanning tree waits for the forward-delay timer to expire, it moves the interface to the learning state and resets the forward-delay timer.
3. In the learning state, the interface continues to block frame forwarding as the switch learns end-station location information for the forwarding database.
4. When the forward-delay timer expires, spanning tree moves the interface to the forwarding state, where both learning and frame forwarding are enabled.

Blocking State

A Layer 2 interface in the blocking state does not participate in frame forwarding. After initialization, a BPDU is sent to each interface in the switch. A switch initially functions as the root until it exchanges BPDUs with other switches. This exchange establishes which switch in the network is the root or root switch. If there is only one switch in the network, no exchange occurs, the forward-delay timer expires, and the interfaces move to the listening state. An interface always enters the blocking state after switch initialization.

An interface in the blocking state performs as follows:

- Discards frames received on the port
- Discards frames switched from another interface for forwarding
- Does not learn addresses
- Receives BPDUs

Listening State

The listening state is the first state a Layer 2 interface enters after the blocking state. The interface enters this state when the spanning tree determines that the interface should participate in frame forwarding.

An interface in the listening state performs as follows:

- Discards frames received on the port
- Discards frames switched from another interface for forwarding
- Does not learn addresses
- Receives BPDUs

Learning State

A Layer 2 interface in the learning state prepares to participate in frame forwarding. The interface enters the learning state from the listening state.

An interface in the learning state performs as follows:

- Discards frames received on the port
- Discards frames switched from another interface for forwarding
- Learns addresses
- Receives BPDUs

Forwarding State

A Layer 2 interface in the forwarding state forwards frames. The interface enters the forwarding state from the learning state.

An interface in the forwarding state performs as follows:

- Receives and forwards frames received on the port
- Forwards frames switched from another port
- Learns addresses
- Receives BPDUs

Disabled State

A Layer 2 interface in the disabled state does not participate in frame forwarding or in the spanning tree. An interface in the disabled state is nonoperational.

A disabled interface performs as follows:

- Forwards frames switched from another interface for forwarding
- Learns addresses
- Does not receive BPDUs

Spanning-Tree Address Management

IEEE 802.1D specifies 17 multicast addresses, ranging from 0x00180C2000000 to 0x0180C2000010, to be used by different bridge protocols. These addresses are static addresses that cannot be removed.

The ML-Series card switches supported BPDUs (0x0180C2000000 and 01000CCCCCD) when they are being tunneled via the protocol tunneling feature.

STP and IEEE 802.1Q Trunks

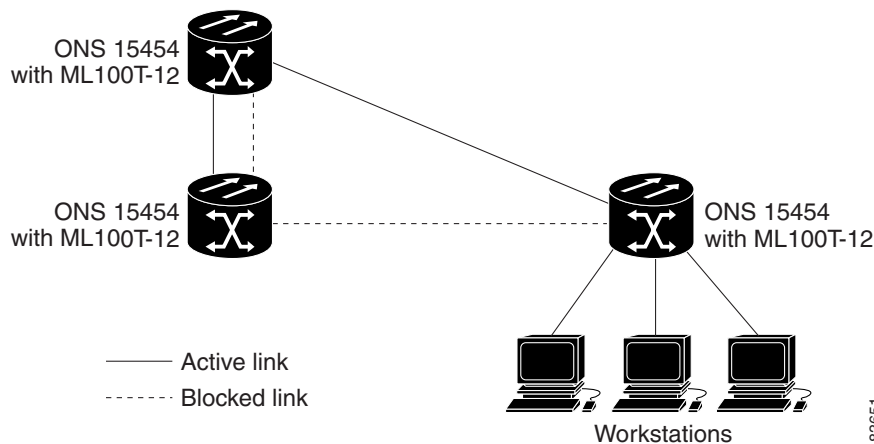
When you connect a Cisco switch to a non-Cisco device through an 802.1Q trunk, the Cisco switch uses PVST+ to provide spanning-tree interoperability. PVST+ is automatically enabled on 802.1Q trunks after users assign a protocol to a bridge group. The external spanning-tree behavior on access ports and Inter-Switch Link (ISL) trunk ports is not affected by PVST+.

For more information on 802.1Q trunks, see [Chapter 7, “Configuring VLANs.”](#)

Spanning Tree and Redundant Connectivity

You can create a redundant backbone with spanning tree by connecting two switch interfaces to another device or to two different devices. Spanning tree automatically disables one interface but enables it if the other one fails, as shown in Figure 6-3. If one link is high speed and the other is low speed, the low-speed link is always disabled. If the speeds are the same, the port priority and port ID are added together, and spanning tree disables the link with the lowest value.

Figure 6-3 Spanning Tree and Redundant Connectivity



You can also create redundant links between switches by using EtherChannel groups. For more information, see [Chapter 9, “Configuring Link Aggregation.”](#)

Accelerated Aging to Retain Connectivity

The default for aging dynamic addresses is 5 minutes, which is the default setting of the **bridge bridge-group-number aging-time** global configuration command. However, a spanning-tree reconfiguration can cause many station locations to change. Because these stations could be unreachable for 5 minutes or more during a reconfiguration, the address-aging time is accelerated so that station addresses can be dropped from the address table and then relearned.

Because each VLAN is a separate spanning-tree instance, the switch accelerates aging on a per-VLAN basis. A spanning-tree reconfiguration on one VLAN can cause the dynamic addresses learned on that VLAN to be subject to accelerated aging. Dynamic addresses on other VLANs can be unaffected and remain subject to the aging interval entered for the switch.

RSTP

RSTP provides rapid convergence of the spanning tree. It improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths). The most common initial deployment of RSTP is in the backbone and distribution layers of a Layer 2 switched network; this deployment provides the highly available network required in a service-provider environment.

RSTP improves the operation of the spanning tree while maintaining backward compatibility with equipment that is based on the (original) 802.1D spanning tree.

RSTP takes advantage of point-to-point wiring and provides rapid convergence of the spanning tree. Reconfiguration of the spanning tree can occur in less than 2 second (in contrast to 50 seconds with the default settings in the 802.1D spanning tree), which is critical for networks carrying delay-sensitive traffic such as voice and video.

These sections describe how RSTP works:

- [Port Roles and the Active Topology, page 6-9](#)
- [Rapid Convergence, page 6-10](#)
- [Synchronization of Port Roles, page 6-12](#)
- [Bridge Protocol Data Unit Format and Processing, page 6-12](#)
- [Topology Changes, page 6-13](#)

Supported RSTP Instances

The ML Series supports per-VLAN rapid spanning tree (PVRST) and a maximum of 255 rapid spanning-tree instances.

Port Roles and the Active Topology

The RSTP provides rapid convergence of the spanning tree by assigning port roles and by determining the active topology. The RSTP builds upon the IEEE 802.1D STP to select the switch with the highest switch priority (lowest numerical priority value) as the root switch as described in [“Election of the Root Switch” section on page 6-3](#). Then the RSTP assigns one of these port roles to individual ports:

- Root port—Provides the best path (lowest cost) when the switch forwards packets to the root switch.

- Designated port—Connects to the designated switch, which incurs the lowest path cost when forwarding packets from that LAN to the root switch. The port through which the designated switch is attached to the LAN is called the designated port.
- Alternate port—Offers an alternate path toward the root switch to that provided by the current root port.
- Backup port—Acts as a backup for the path provided by a designated port toward the leaves of the spanning tree. A backup port can exist only when two ports are connected together in a loopback by a point-to-point link or when a switch has two or more connections to a shared LAN segment.
- Disabled port—Has no role within the operation of the spanning tree.

A port with the root or a designated port role is included in the active topology. A port with the alternate or backup port role is excluded from the active topology.

In a stable topology with consistent port roles throughout the network, the RSTP ensures that every root port and designated port immediately transition to the forwarding state while all alternate and backup ports are always in the discarding state (equivalent to blocking in 802.1D). The port state controls the operation of the forwarding and learning processes. Table 6-3 provides a comparison of 802.1D and RSTP port states.

Table 6-3 Port State Comparison

| Operational Status | STP Port State | RSTP Port State | Is Port Included in the Active Topology? |
|--------------------|----------------|-----------------|--|
| Enabled | Blocking | Discarding | No |
| Enabled | Listening | Discarding | No |
| Enabled | Learning | Learning | Yes |
| Enabled | Forwarding | Forwarding | Yes |
| Disabled | Disabled | Discarding | No |



Caution

STP edge ports are bridge ports that do not need STP enabled, where loop protection is not needed out of that port or an STP neighbor does not exist out of that port. For RSTP, it is important to disable STP on edge ports, which are typically front-side Ethernet ports, using the command **bridge bridge-group-number spanning-disabled** on the appropriate interface. If RSTP is not disabled on edge ports, convergence times will be excessive for packets traversing those ports.



Note

To be consistent with Cisco STP implementations, Table 6-3 describes the port state as *blocking* instead of *discarding*. Designated ports start in the listening state.

Rapid Convergence

The RSTP provides for rapid recovery of connectivity following the failure of switch, a switch port, or a LAN. It provides rapid convergence for new root ports, and ports connected through point-to-point links as follows:

- Root ports—If the RSTP selects a new root port, it blocks the old root port and immediately transitions the new root port to the forwarding state.

- Point-to-point links—If you connect a port to another port through a point-to-point link and the local port becomes a designated port, it negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology.

As shown in Figure 6-4, Switch A is connected to Switch B through a point-to-point link, and all of the ports are in the blocking state. Assume that the priority of Switch A is a smaller numerical value than the priority of Switch B. Switch A sends a proposal message (a configuration BPDU with the proposal flag set) to Switch B, proposing itself as the designated switch.

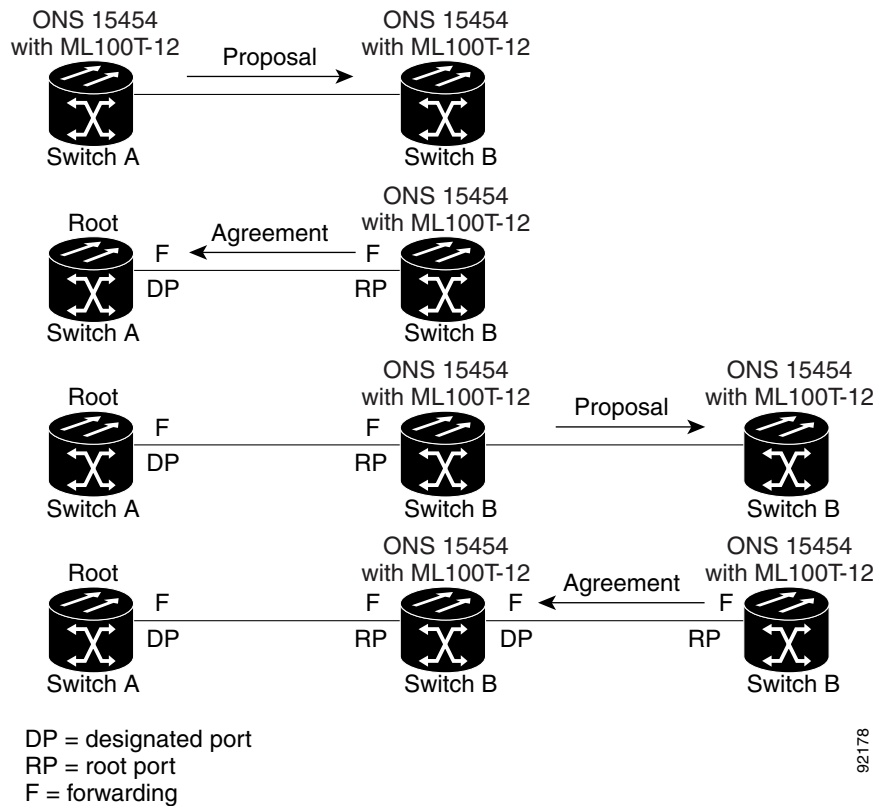
After receiving the proposal message, Switch B selects as its new root port the port from which the proposal message was received, forces all non-edge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

After receiving an agreement message from Switch B, Switch A also immediately transitions its designated port to the forwarding state. No loops in the network are formed because Switch B blocked all of its non-edge ports and because there is a point-to-point link between Switches A and B.

When Switch C is connected to Switch B, a similar set of handshaking messages are exchanged. Switch C selects the port connected to Switch B as its root port, and both ends immediately transition to the forwarding state. With each iteration of this handshaking process, one more switch joins the active topology. As the network converges, this proposal-agreement handshaking progresses from the root toward the leaves of the spanning tree.

The switch determines the link type from the port duplex mode: a full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

Figure 6-4 Proposal and Agreement Handshaking for Rapid Convergence



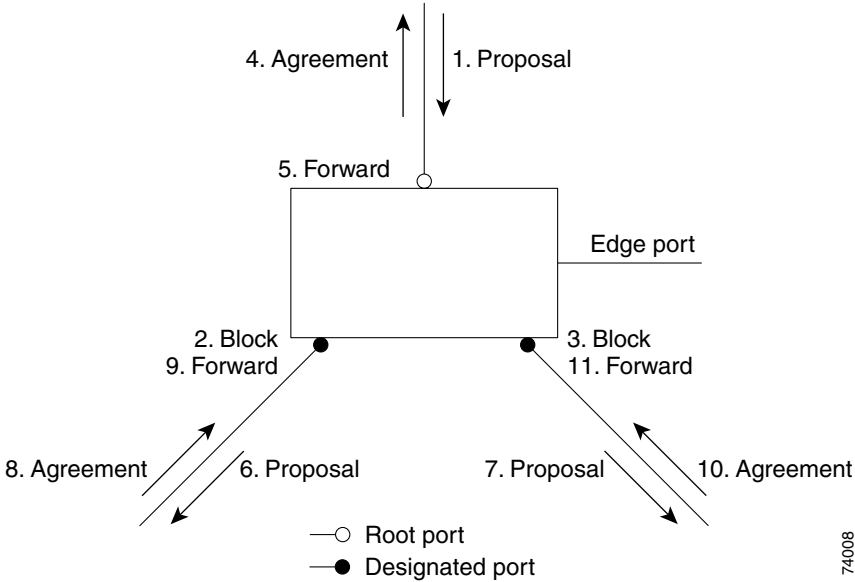
Synchronization of Port Roles

When the switch receives a proposal message on one of its ports and that port is selected as the new root port, the RSTP forces all other ports to synchronize with the new root information. The switch is synchronized with superior root information received on the root port if all other ports are synchronized.

If a designated port is in the forwarding state, it transitions to the blocking state when the RSTP forces it to synchronize with new root information. In general, when the RSTP forces a port to synchronize with root information and the port does not satisfy any of the above conditions, its port state is set to blocking.

After ensuring all of the ports are synchronized, the switch sends an agreement message to the designated switch corresponding to its root port. When the switches connected by a point-to-point link are in agreement about their port roles, the RSTP immediately transitions the port states to forwarding. The sequence of events is shown in Figure 6-5.

Figure 6-5 Sequence of Events During Rapid Convergence



Bridge Protocol Data Unit Format and Processing

The RSTP BPDU format is the same as the IEEE 802.1D BPDU format except that the protocol version is set to 2. A new Length field is set to zero, which means that no version 1 protocol information is present. Table 6-4 shows the RSTP flag fields.

Table 6-4 RSTP BPDU Flags

| Bit | Function |
|-----|----------------------|
| 0 | Topology change (TC) |
| 1 | Proposal |

Table 6-4 RSTP BPDU Flags (continued)

| Bit | Function |
|------|---------------------------------|
| 2–3: | Port role: |
| 00 | Unknown |
| 01 | Alternate port |
| 10 | Root port |
| 11 | Designated port |
| 4 | Learning |
| 5 | Forwarding |
| 6 | Agreement |
| 7 | Topology change acknowledgement |

The sending switch sets the proposal flag in the RSTP BPDU to propose itself as the designated switch on that LAN. The port role in the proposal message is always set to the designated port.

The sending switch sets the agreement flag in the RSTP BPDU to accept the previous proposal. The port role in the agreement message is always set to the root port.

The RSTP does not have a separate topology change notification (TCN) BPDU. It uses the topology change (TC) flag to show the topology changes. However, for interoperability with 802.1D switches, the RSTP switch processes and generates TCN BPDUs.

The learning and forwarding flags are set according to the state of the sending port.

Processing Superior BPDU Information

If a port receives superior root information (lower bridge ID, lower path cost, etc.) than currently stored for the port, the RSTP triggers a reconfiguration. If the port is proposed and is selected as the new root port, RSTP forces all the other ports to synchronize.

If the BPDU received is an RSTP BPDU with the proposal flag set, the switch sends an agreement message after all of the other ports are synchronized. If the BPDU is an 802.1D BPDU, the switch does not set the proposal flag and starts the forward-delay timer for the port. The new root port requires twice the forward-delay time to transition to the forwarding state.

If the superior information received on the port causes the port to become a backup or alternate port, RSTP sets the port to the blocking state but does not send the agreement message. The designated port continues sending BPDUs with the proposal flag set until the forward-delay timer expires, at which time the port transitions to the forwarding state.

Processing Inferior BPDU Information

If a designated port receives an inferior BPDU (higher bridge ID, higher path cost, etc.) than currently stored for the port with a designated port role, it immediately replies with its own information.

Topology Changes

This section describes the differences between the RSTP and the 802.1D in handling spanning-tree topology changes.

- **Detection**—Unlike 802.1D in which *any* transition between the blocking and the forwarding state causes a topology change, *only* transitions from the blocking to the forwarding state cause a topology change with RSTP. (Only an increase in connectivity is considered a topology change.) State changes on an edge port do not cause a topology change. When an RSTP switch detects a topology change, it flushes the learned information on all of its non-edge ports.
- **Notification**—Unlike 802.1D, which uses TCN BPDUs, the RSTP does not use them. However, for 802.1D interoperability, an RSTP switch processes and generates TCN BPDUs.
- **Acknowledgement**—When an RSTP switch receives a TCN message on a designated port from an 802.1D switch, it replies with an 802.1D configuration BPDU with the topology change acknowledgement bit set. However, if the TC-while timer (the same as the topology-change timer in 802.1D) is active on a root port connected to an 802.1D switch and a configuration BPDU with the topology change acknowledgement bit set is received, the TC-while timer is reset.

This behavior is only required to support 802.1D switches. The RSTP BPDUs never have the topology change acknowledgement bit set.

- **Propagation**—When an RSTP switch receives a TC message from another switch through a designated or root port, it propagates the topology change to all of its non-edge, edge, designated ports, and root port (excluding the port on which it is received). The switch starts the TC-while timer for all such ports and flushes the information learned on them.
- **Protocol migration**—For backward compatibility with 802.1D switches, RSTP selectively sends 802.1D configuration BPDUs and TCN BPDUs on a per-port basis.

When a port is initialized, the timer is started (which specifies the minimum time during which RSTP BPDUs are sent), and RSTP BPDUs are sent. While this timer is active, the switch processes all BPDUs received on that port and ignores the protocol type.

If the switch receives an 802.1D BPDU after the port's migration-delay timer has expired, it assumes that it is connected to an 802.1D switch and starts using only 802.1D BPDUs. However, if the RSTP switch is using 802.1D BPDUs on a port and receives an RSTP BPDU after the timer has expired, it restarts the timer and starts using RSTP BPDUs on that port.

Interoperability with 802.1D STP

A switch running RSTP supports a built-in protocol migration mechanism that enables it to interoperate with legacy 802.1D switches. If this switch receives a legacy 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port.

However, the switch does not automatically revert to the RSTP mode if it no longer receives 802.1D BPDUs because it cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. Also, a switch might continue to assign a boundary role to a port when the switch to which this switch is connected has joined the region.

Configuring STP and RSTP Features

These sections describe how to configure spanning-tree features:

- [Default STP and RSTP Configuration, page 6-15](#)
- [Disabling STP and RSTP, page 6-15](#)
- [Configuring the Root Switch, page 6-16](#)
- [Configuring the Port Priority, page 6-16](#)

- [Configuring the Path Cost, page 6-17](#)
- [Configuring the Switch Priority of a Bridge Group, page 6-18](#)
- [Configuring the Hello Time, page 6-18](#)
- [Configuring the Forwarding-Delay Time for a Bridge Group, page 6-18](#)
- [Configuring the Maximum-Aging Time for a Bridge Group, page 6-19](#)

Default STP and RSTP Configuration

Table 6-5 shows the default STP configuration.

Table 6-5 Default STP and RSTP Configuration

| Feature | Default Setting |
|---|---|
| Enable state | Up to 255 spanning-tree instances can be enabled. |
| Switch priority | 32768 + Bridge ID |
| Spanning-tree port priority (configurable on a per-interface basis—used on interfaces configured as Layer 2 access ports) | 128 |
| Spanning-tree port cost (configurable on a per-interface basis) | 1000 Mbps: 4 100 Mbps: 19 10 Mbps: 100 STS-1: 37 STS-3c: 14 STS-6c: 9 STS-9c: 7 STS-12c: 6 STS-24c: 3 |
| Hello time | 2 seconds |
| Forward-delay time | 15 seconds |
| Maximum-aging time | 20 seconds |

Disabling STP and RSTP

STP is enabled by default on VLAN 1 and on all newly created VLANs up to the specified spanning-tree limit of 255. Disable STP only if you are sure there are no loops in the network topology.



Caution

STP edge ports are bridge ports that do not need STP enabled, where loop protection is not needed out of that port or an STP neighbor does not exist out of that port. For RSTP, it is important to disable STP on edge ports, which are typically front-side Ethernet ports, using the command **bridge bridge-group-number spanning-disabled** on the appropriate interface. If RSTP is not disabled on edge ports, convergence times will be excessive for packets traversing those ports.

**Caution**

When STP is disabled and loops are present in the topology, excessive traffic and indefinite packet duplication can drastically reduce network performance.

Beginning in privileged EXEC mode, follow these steps to disable STP or RSTP on a per-VLAN basis:

| | Command | Purpose |
|--------|--|--|
| Step 1 | <code>Router# configure terminal</code> | Enters the global configuration mode. |
| Step 2 | <code>Router(config)# interface interface-id</code> | Enters the interface configuration mode. |
| Step 3 | <code>Router(config-if)# bridge-group bridge-group-number spanning disabled</code> | Disables STP or RSTP on a per-interface basis. |
| Step 4 | <code>Router(config-if)# end</code> | Returns to privileged EXEC mode. |

To re-enable STP, use the `no bridge-group bridge-group-number spanning disabled` interface-level configuration command.

Configuring the Root Switch

The switch maintains a separate spanning-tree instance for each active VLAN configured on it. A bridge ID, consisting of the switch priority and the switch MAC address, is associated with each instance. For each VLAN, the switch with the lowest bridge ID becomes the root switch for that VLAN.

**Note**

If your network consists of switches that both do and do not support the extended system ID, it is unlikely that the switch with the extended system ID support will become the root switch. The extended system ID increases the switch priority value every time the bridge ID is greater than the priority of the connected switches that are running older software.

Configuring the Port Priority

If a loop occurs, spanning tree uses the port priority when selecting an interface to put into the forwarding state. You can assign higher priority values (lower numerical values) to interfaces that you want selected first, and lower priority values (higher numerical values) that you want selected last. If all interfaces have the same priority value, spanning tree puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Beginning in privileged EXEC mode, follow these steps to configure the port priority of an interface:

| | Command | Purpose |
|--------|---|--|
| Step 1 | <code>Router# configure terminal</code> | Enters the global configuration mode. |
| Step 2 | <code>Router(config)# interface interface-id</code> | Enters the interface configuration mode, and specifies an interface to configure. Valid interfaces include physical interfaces and port-channel logical interfaces (port-channel port-channel-number). |

| | Command | Purpose |
|--------|---|--|
| Step 3 | <code>Router(config-if)# bridge-group bridge-group-number priority value</code> | Configures the port priority for an interface that is an access port. For the priority <i>value</i> , the range is 0 to 255; the default is 128 in increments of 16. The lower the number, the higher the priority. |
| Step 4 | <code>Router(config-if)# end</code> | Return to privileged EXEC mode. |

To return the interface to its default setting, use the **no bridge-group id bridge-group-number priority value** command.

Configuring the Path Cost

The spanning-tree path cost default value is derived from the media speed of an interface. If a loop occurs, spanning tree uses cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want selected first and higher cost values to interfaces that you want selected last. If all interfaces have the same cost value, spanning tree puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Beginning in privileged EXEC mode, follow these steps to configure the cost of an interface:

| | Command | Purpose |
|--------|---|--|
| Step 1 | <code>Router# configure terminal</code> | Enters the global configuration mode. |
| Step 2 | <code>Router(config)# interface interface-id</code> | Enters the interface configuration mode and specifies an interface to configure. Valid interfaces include physical interfaces and port-channel logical interfaces (port-channel <i>port-channel-number</i>). |
| Step 3 | <code>Router(config-if)# bridge-group bridge-group-number path-cost cost</code> | Configures the cost for an interface that is an access port. If a loop occurs, spanning tree uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission. For <i>cost</i> , the range is 0 to 65535; the default value is derived from the media speed of the interface. |
| Step 4 | <code>Router(config-if)# end</code> | Returns to the privileged EXEC mode. |



Note The **show spanning-tree interface** *interface-id* privileged EXEC command displays information only for ports that are in a link-up operative state. Otherwise, you can use the **show running-config** privileged EXEC command to confirm the configuration.

To return the interface to its default setting, use the **no bridge-group bridge-group-number path-cost cost** command.

Configuring the Switch Priority of a Bridge Group

You can configure the switch priority and make it more likely that the switch will be chosen as the root switch.

Beginning in privileged EXEC mode, follow these steps to configure the switch priority of a bridge group:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router# configure terminal | Enters the global configuration mode. |
| Step 2 | Router(config)# bridge bridge-group-number priority priority | Configures the switch priority of a bridge group. For <i>priority</i> , the range is 0 to 61440 in increments of 4096; the default is 32768. The lower the number, the more likely the switch will be chosen as the root switch. The value entered is rounded to the lower multiple of 4096. The actual number is computed by adding this number to the bridge group number. |
| Step 3 | Router(config)# end | Return to the privileged EXEC mode. |

To return the switch to its default setting, use the **no bridge bridge-group-number priority priority** command.

Configuring the Hello Time

You can configure the interval between the generation of configuration messages by the root switch by changing the hello time.

Beginning in privileged EXEC mode, follow these steps to configure the hello time of a bridge group:

| | Command | Purpose |
|--------|--|--|
| Step 1 | Router# configure terminal | Enters global configuration mode. |
| Step 2 | Router(config)# bridge bridge-group-number hello-time seconds | Configures the hello time of a bridge group. The hello time is the interval between the generation of configuration messages by the root switch. These messages mean that the switch is alive. For <i>seconds</i> , the range is 1 to 10; the default is 2. |
| Step 3 | Router(config)# end | Returns to privileged EXEC mode. |

To return the switch to its default setting, use the **no bridge bridge-group-number hello-time seconds** command.

Configuring the Forwarding-Delay Time for a Bridge Group

Beginning in privileged EXEC mode, follow these steps to configure the forwarding-delay time for a bridge group:

| | Command | Purpose |
|--------|--|--|
| Step 1 | Router# configure terminal | Enters global configuration mode. |
| Step 2 | Router(config)# bridge bridge-group-number forward-time seconds | Configures the forward time of a VLAN. The forward delay is the number of seconds a port waits before changing from its spanning-tree learning and listening states to the forwarding state. For <i>seconds</i> , the range is 4 to 200; the default is 15. |
| Step 3 | Router(config)# end | Returns to privileged EXEC mode. |

To return the switch to its default setting, use the **no bridge *bridge-group-number* forward-time *seconds*** command.

Configuring the Maximum-Aging Time for a Bridge Group

Beginning in privileged EXEC mode, follow these steps to configure the maximum-aging time for a bridge group:

| | Command | Purpose |
|--------|---|---|
| Step 1 | Router# configure terminal | Enters global configuration mode. |
| Step 2 | Router(config)# bridge bridge-group-number max-age seconds | Configures the maximum-aging time of a bridge group. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration. For <i>seconds</i> , the range is 6 to 200; the default is 20. |
| Step 3 | Router(config)# end | Returns to privileged EXEC mode. |

To return the switch to its default setting, use the **no bridge *bridge-group-number* max-age *seconds*** command.

Verifying and Monitoring STP and RSTP Status

To display the STP or RSTP status, use one or more of the privileged EXEC commands in [Table 6-6](#):

Table 6-6 Commands for Displaying Spanning-Tree Status

| Command | Purpose |
|---|---|
| Router# show spanning-tree active | Displays STP or RSTP information on active interfaces only. |
| Router# show spanning-tree detail | Displays a detailed summary of interface information. |
| Router# show spanning-tree interface <i>interface-id</i> | Displays STP or RSTP information for the specified interface. |
| Router# show spanning-tree summary [totals] | Displays a summary of port states or displays the total lines of the STP or RSTP state section. |

**Note**

The **show spanning-tree interface *interface-id*** privileged EXEC command displays information only if the port is in a link-up operative state. Otherwise, you can use the **show running-config interface** privileged EXEC command to confirm the configuration.

Examples of the **show spanning-tree** privileged EXEC command commands are shown here:

Example 6-1 show spanning-tree command

```
Router# show spanning-tree active
```

```
Bridge group 1
Spanning tree enabled protocol ieee
Root ID Priority 32769
Address 0005.9a39.6634
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
Address 0005.9a39.6634
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
```

| Interface | Role | Sts | Cost | Prio.Nbr | Type |
|-----------|------|-----|------|----------|------|
| Fa0 | Desg | FWD | 19 | 128.3 | P2p |
| PO0 | Desg | FWD | 3 | 128.20 | P2p |

```
Router# show spanning-tree detail
```

```
Bridge group 1 is executing the ieee compatible Spanning Tree protocol
Bridge Identifier has priority 32768, sysid 1, address 0005.9a39.6634
Configured hello time 2, max age 20, forward delay 15
We are the root of the spanning tree
Topology change flag not set, detected flag not set
Number of topology changes 2 last change occurred 00:16:45 ago
from POS0
Times: hold 1, topology change 35, notification 2
hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0, aging 300
```

```
Port 3 (FastEthernet0) of Bridge group 1 is forwarding
Port path cost 19, Port priority 128, Port Identifier 128.3.
Designated root has priority 32769, address 0005.9a39.6634
Designated bridge has priority 32769, address 0005.9a39.6634
Designated port id is 128.3, designated path cost 0
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 1
Link type is point-to-point by default
BPDU: sent 641, received 0
```

```
Port 20 (POS0) of Bridge group 1 is forwarding
Port path cost 3, Port priority 128, Port Identifier 128.20.
Designated root has priority 32769, address 0005.9a39.6634
Designated bridge has priority 32769, address 0005.9a39.6634
Designated port id is 128.20, designated path cost 0
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 6
Link type is point-to-point by default
BPDU: sent 582, received 15
```


Router# **show spanning-tree interface fast 0**

| Bridge Group | Role | Sts | Cost | Prio.Nbr | Type |
|----------------|------|-----|------|----------|------|
| Bridge group 1 | Desg | FWD | 19 | 128.3 | P2p |

Router# **show spanning-tree interface pos 0**

| Bridge Group | Role | Sts | Cost | Prio.Nbr | Type |
|----------------|------|-----|------|----------|------|
| Bridge group 1 | Desg | FWD | 3 | 128.20 | P2p |

Router# **show spanning-tree summary totals**

Switch is in pvst mode

Root bridge for: Bridge group 1

| Name | Blocking | Listening | Learning | Forwarding | STP Active |
|----------|----------|-----------|----------|------------|------------|
| 1 bridge | 0 | 0 | 0 | 2 | 2 |



Configuring VLANs

This chapter describes VLAN configurations for the ML-Series card. It describes how to configure IEEE 802.1Q VLAN encapsulation. For more information about the Cisco Internet Operating System (IOS) commands used in this chapter, refer to the *Cisco IOS Command Reference* publication.

This chapter contains the following major sections:

- [Understanding VLANs, page 7-1](#)
- [Configuring IEEE 802.1Q VLAN Encapsulation, page 7-2](#)
- [IEEE 802.1Q VLAN Configuration, page 7-3](#)
- [Monitoring and Verifying VLAN Operation, page 7-5](#)



Note

Configuring VLANs is optional. Complete general interface configurations before proceeding with configuring VLANs as an optional step.

Understanding VLANs

VLANs or bridge groups enable network managers to group users logically rather than by physical location. A VLAN is an emulation of a standard LAN that allows secure intra-group data transfer and communication to occur without the traditional restraints placed on the network. It can also be considered a broadcast domain set up within a switch. With VLANs, switches can support more than one subnet (or VLAN) on each switch and give routers and switches the opportunity to support multiple subnets on a single physical link. A group of devices that belong to the same VLAN, but are part of different LAN segments, are configured to communicate as if they were part of the same LAN segment.

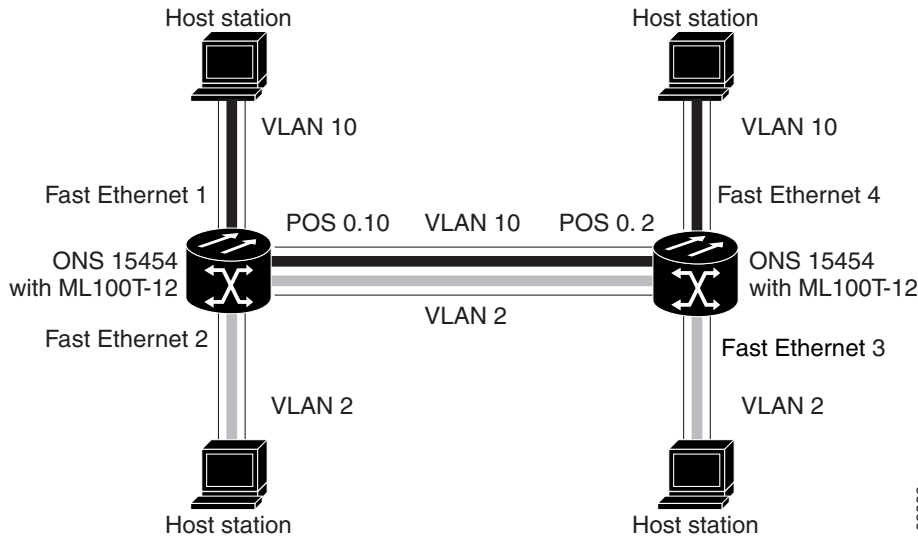
VLANs enable efficient traffic separation and provide excellent bandwidth utilization. VLANs also alleviate scaling issues by logically segmenting the physical LAN structure into different subnetworks so that packets are switched only between ports within the same VLAN. This can be very useful for security, broadcast containment, and accounting.

ML-Series software supports port-based VLANs and VLAN trunk ports, which are ports that carry the traffic of multiple VLANs. Each frame transmitted on a trunk link is tagged as belonging to only one VLAN.

ML-Series software supports VLAN frame encapsulation through the IEEE 802.1Q standard on both the ML100T-12 and the ML1000-2. The Cisco ISL VLAN frame encapsulation is not supported. ISL frames will be broadcast at Layer 2, or dropped at Layer 3.

ML-Series switching supports up to 900 VLAN subinterfaces per card (for example, 200 VLANs on 4 interfaces uses 800 VLAN subinterfaces). A maximum of 255 logical VLANs can be bridged per card (limited by the number of bridge-groups). Each VLAN subinterface can be configured for any VLAN ID in the full 1–4095 range. Figure 7-1 shows a network topology in which two VLANs span two ONS 15454s with ML-Series cards.

Figure 7-1 VLANs Spanning Devices in a Network



Configuring IEEE 802.1Q VLAN Encapsulation

On an IEEE 802.1Q trunk port, all transmitted and received frames are tagged except for those on the VLAN configured as the native VLAN for the port. Frames on the native VLAN are always transmitted untagged and are normally received untagged. You can configure VLAN encapsulation on both the ML100T-12 and the ML1000-2.

On an IEEE 802.1Q trunk port, all transmitted and received frames are tagged except for those on the VLAN configured as the native VLAN for the port. On ML-series cards, the native VLAN is always VLAN ID 1. Frames on the native VLAN are normally transmitted untagged and are normally received untagged. Tagging of transmitted native VLAN frames can be forced by the global configuration command `vlan dot1q tag native`. VLAN encapsulation is supported on both the ML100T-12 and the ML1000-2. VLAN encapsulation is supported for routing and bridging, and is supported on Ethernet interfaces and on POS interfaces with PPP and LEX encapsulation.

To configure VLANs using IEEE 802.1Q VLAN encapsulation, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router(config)# bridge <i>bridge-group-number</i> protocol <i>type</i> | Assigns a bridge group (VLAN) number and define the appropriate spanning tree type. See Chapter 5, “Configuring Bridging.” |
| Step 2 | Router(config)# interface <i>type number</i> | Enters interface configuration mode to configure the interface. |
| Step 3 | Router(config-if)# no ip address | Disables IP processing. |
| Step 4 | Router(config)# interface <i>type</i> <i>number.subinterface-number</i> | Enters subinterface configuration mode to configure the subinterface. |
| Step 5 | Router(config-subif)# encap dot1q <i>vlan-number</i> | Sets the encapsulation format on the VLAN to IEEE 802.1Q. |
| Step 6 | Router(config-subif)# bridge-group <i>bridge-group-number</i> | Assigns a network interface to a bridge group. |
| Step 7 | Router(config-subif)# end | Returns to privileged EXEC mode. |
| Step 8 | Router# copy running-config startup-config | (Optional) Saves your configuration changes to NVRAM. |

**Note**

In a bridge group on the ML-Series card, the VLAN ID does not have to be uniform across interfaces that belong to that bridge group. For example, a bridge-group can connect from a VLAN ID subinterface to a subinterface with a different VLAN ID, and then frames entering with one VLAN ID can be changed to exit with a different VLAN ID. This is known as VLAN translation.

**Note**

IP routing is enabled by default. To enable bridging, enter the **no ip routing** or **bridge IRB** command.

**Note**

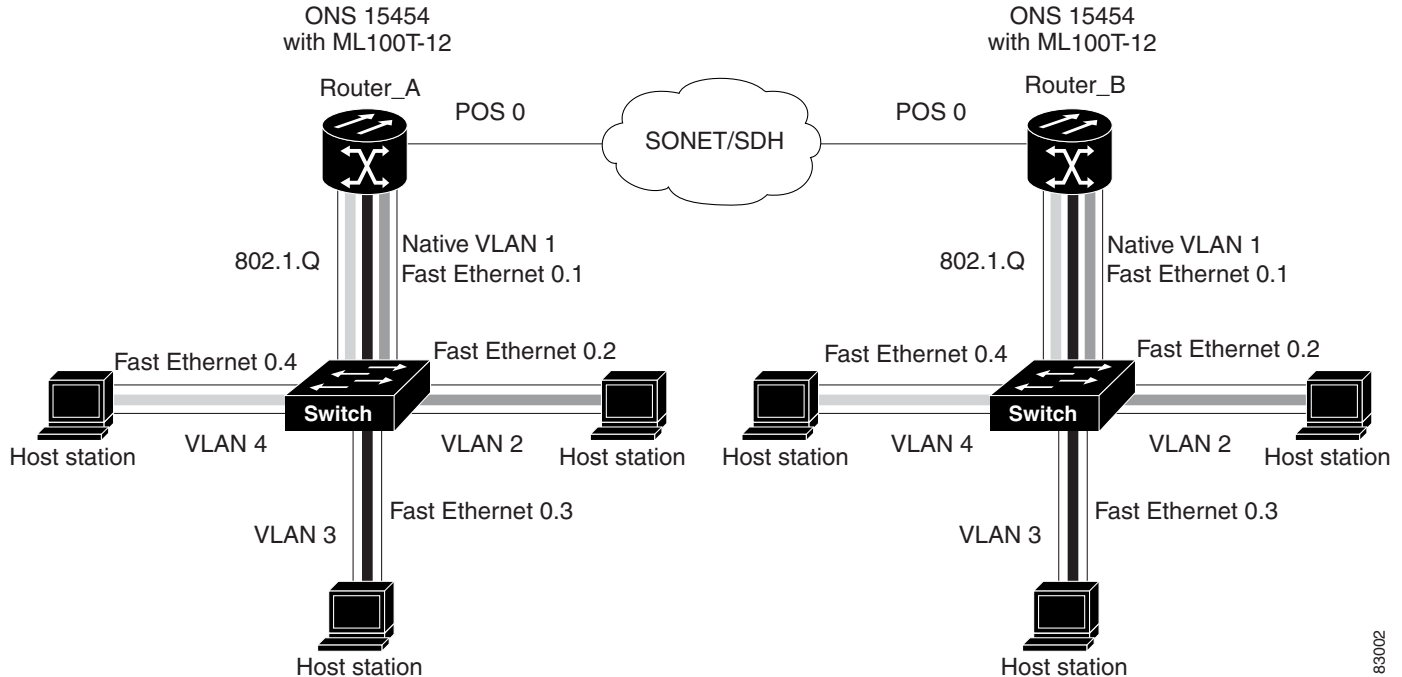
Native VLAN frames transmitted on the interface are normally untagged. All untagged frames received on the interface are associated with the native VLAN, which is always VLAN 1. Use the command **encapsulation dot1q 1 native**.

IEEE 802.1Q VLAN Configuration

The VLAN configuration example for the ML100T-12 shown in [Figure 7-2](#) depicts the following:

- Fast Ethernet subinterface 0.1 is in the IEEE 802.1Q native VLAN 1.
- Fast Ethernet subinterface 0.2 is in the IEEE 802.1Q VLAN 2.
- Fast Ethernet subinterface 0.3 is in the IEEE 802.1Q VLAN 3.
- Fast Ethernet subinterface 0.4 is in the IEEE 802.1Q VLAN 4.

Figure 7-2 Bridging IEEE 802.1Q VLANs



83002

The following shows how to configure VLANs for IEEE 802.1Q VLAN encapsulation. Use this configuration for both router A and router B. The example is shown in [Figure 7-2](#):

Example 7-1 Configure VLANs for IEEE8021Q VLAN Encapsulation

```

bridge 1 protocol ieee
bridge 2 protocol ieee
bridge 3 protocol ieee
bridge 4 protocol ieee
!
!
interface FastEthernet0
no ip address
!
interface FastEthernet0.1
encapsulation dot1Q 1 native
bridge-group 1
!
interface FastEthernet0.2
encapsulation dot1Q 2
bridge-group 2
!
interface FastEthernet0.3
encapsulation dot1Q 3
bridge-group 3
!
interface FastEthernet0.4
encapsulation dot1Q 4
bridge-group 4
!
interface POS0
no ip address
crc 32

```

```
pos flag c2 1
!
interface POS0.1
  encapsulation dot1Q 1 native
  bridge-group 1
!
interface POS0.2
  encapsulation dot1Q 2
  bridge-group 2
!
interface POS0.3
  encapsulation dot1Q 3
  bridge-group 3
!
interface POS0.4
  encapsulation dot1Q 4
  bridge-group 4
```

Monitoring and Verifying VLAN Operation

After the VLANs are configured on the ML-Series card, you can monitor their operation by entering the privileged EXEC command **show vlans *vlan-id***. This command displays information on all configured VLANs or on a specific VLAN (by VLAN ID number).



Caution

Two similar commands exist. The command **show vlans** gives information regarding IEEE 802.1Q VLANs configured on the ML-Series card. The command **show vlan** gives information regarding the VLAN tunnel. For more information on VLAN tunneling, see [Chapter 8, “Configuring IEEE 802.1Q and Layer 2 Protocol Tunneling.”](#)



Configuring IEEE 802.1Q and Layer 2 Protocol Tunneling

Virtual private networks (VPNs) provide enterprise-scale connectivity on a shared infrastructure, often Ethernet-based, with the same security, prioritization, reliability, and manageability requirements of private networks. Tunneling is a feature designed for service providers who carry traffic of multiple customers across their networks and are required to maintain the VLAN and Layer 2 protocol configurations of each customer without impacting the traffic of other customers. The ML-Series cards support IEEE 802.1Q tunneling and Layer 2 protocol tunneling.

This chapter contains these sections:

- [Understanding IEEE 802.1Q Tunneling, page 8-1](#)
- [Configuring IEEE 802.1Q Tunneling, page 8-4](#)
- [Understanding Layer 2 Protocol Tunneling, page 8-6](#)
- [Configuring Layer 2 Protocol Tunneling, page 8-7](#)
- [Monitoring and Verifying Tunneling Status, page 8-9](#)

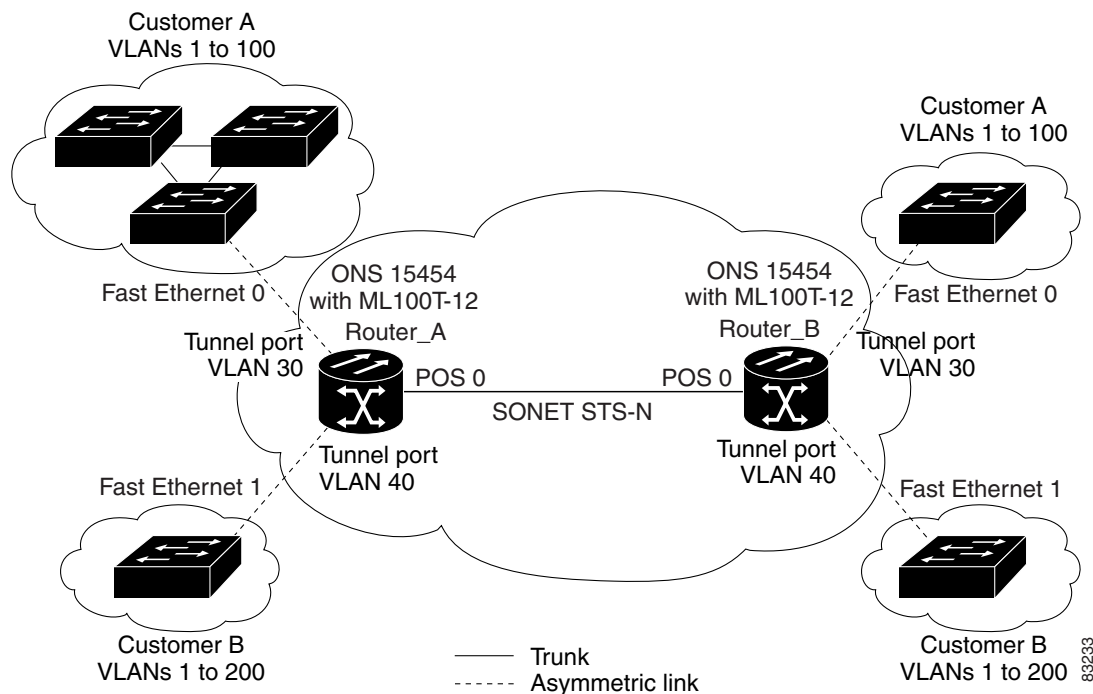
Understanding IEEE 802.1Q Tunneling

Business customers of service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and traffic of customers through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the IEEE 802.1Q VLAN limit of 4096.

Using the IEEE 802.1Q tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs are preserved and traffic from different customers is segregated within the service-provider infrastructure even when they appear to be on the same VLAN. The IEEE 802.1Q tunneling expands VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets. A port configured to support IEEE 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN that is dedicated to tunneling. Each customer requires a separate VLAN, but that VLAN supports all of the customer's VLANs.

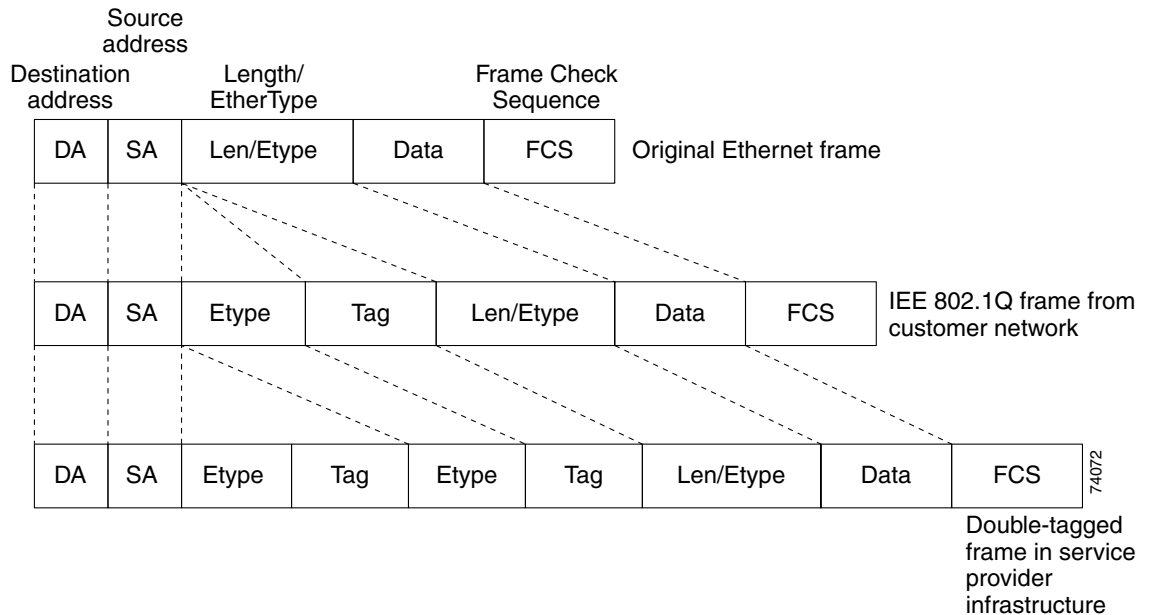
Customer traffic tagged in the normal way with appropriate VLAN IDs comes from an IEEE 802.1Q trunk port on the customer device and into a tunnel port on the ML-Series card. The link between the customer device and the ML-Series card is an asymmetric link because one end is configured as an IEEE 802.1Q trunk port and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID unique to each customer. See [Figure 8-1](#).

Figure 8-1 IEEE 802.1Q Tunnel Ports in a Service-Provider Network



Packets coming from the customer trunk port into the tunnel port on the ML-Series card are normally IEEE 802.1Q-tagged with an appropriate VLAN ID. The tagged packets remain intact inside the ML-Series card and, when they exit the trunk port into the service provider network, are encapsulated with another layer of an IEEE 802.1Q tag (called the *metro tag*) that contains the VLAN ID unique to the customer. The original IEEE 802.1Q tag from the customer is preserved in the encapsulated packet. Therefore, packets entering the service-provider infrastructure are double-tagged, with the outer tag containing the customer's access VLAN ID, and the inner VLAN ID being the VLAN of the incoming traffic.

When the double-tagged packet enters another trunk port in a service provider ML-Series card, the outer tag is stripped as the packet is processed inside the switch. When the packet exits another trunk port on the same core switch, the same metro tag is again added to the packet. Figure 8-2 shows the structure of the double-tagged packet.

Figure 8-2 Normal, IEEE 802.1Q, and 802.1Q Tunneled Ethernet Packet Formats

When the packet enters the trunk port of the service-provider egress switch, the outer tag is again stripped as the packet is processed internally on the switch. However, the metro tag is not added when it is sent out the tunnel port on the edge switch into the customer network, and the packet is sent as a normal IEEE 802.1Q-tagged frame to preserve the original VLAN numbers in the customer network.

In [Figure 8-1 on page 8-2](#), Customer A was assigned VLAN 30, and Customer B was assigned VLAN 40. Packets entering the ML-Series card tunnel ports with IEEE 802.1Q tags are double-tagged when they enter the service-provider network, with the outer tag containing VLAN ID 30 or 40, appropriately, and the inner tag containing the original VLAN number, for example, VLAN 100. Even if both Customers A and B have VLAN 100 in their networks, the traffic remains segregated within the service-provider network because the outer tag is different. With IEEE 802.1Q tunneling, each customer controls its own VLAN numbering space, which is independent of the VLAN numbering space used by other customers and the VLAN numbering space used by the service-provider network.

At the outbound tunnel port, the original VLAN numbers on the customer's network are recovered. If the traffic coming from a customer network is not tagged (native VLAN frames), these packets are bridged or routed as if they were normal packets, and the metro tag is added (as a single-level tag) when they exit toward the service provider network.

If using the native VLAN (VLAN 1) is used in the service provider network as a metro tag, this tag always be added to the customer traffic, even though the native VLAN ID is not normally added to transmitted frames. If the VLAN 1 metro tag is not added on frames entering the service provider network, then the customer VLAN tag appears to be the metro tag, with disastrous results. The global configuration command "vlan dot1q tag native" must be used to prevent this by forcing a tag to be added to VLAN 1. Avoiding the use of VLAN 1 as a metro tag transporting customer traffic is recommended to reduce the risk of misconfiguration. A best practice is to use VLAN 1 as a private management VLAN in the service provider network.

The IEEE 802.1Q class of service (COS) priority field on the added metro tag is set to zero by default, but can be modified by input or output policy maps.

Configuring IEEE 802.1Q Tunneling

This section includes this information about configuring IEEE 802.1Q tunneling:

- [IEEE 802.1Q Tunneling and Other Features, page 8-4](#)
- [Configuring an IEEE 802.1Q Tunneling Port, page 8-4](#)



Note

By default, IEEE 802.1Q tunneling is not configured on the ML-Series.

IEEE 802.1Q Tunneling and Other Features

Although IEEE 802.1Q tunneling works well for Layer 2 packet switching, there are incompatibilities with some Layer 2 features and with Layer 3 switching:

- A tunnel port cannot be a routed port.
- Tunnel ports do not support IP access control lists (ACLs).
- Layer 3 quality of service (QoS) ACLs and other QoS features related to Layer 3 information are not supported on tunnel ports. MAC-based QoS is supported on tunnel ports.
- EtherChannel port groups are compatible with tunnel ports as long as the IEEE 802.1Q configuration is consistent within an EtherChannel port group.
- Port Aggregation Protocol (PAgP) and Unidirectional Link Detection (UDLD) Protocol are not supported on IEEE 802.1Q tunnel ports.
- Dynamic Trunking Protocol (DTP) is not compatible with IEEE 802.1Q tunneling because you must manually configure asymmetric links with tunnel ports and trunk ports.
- Loopback detection is supported on IEEE 802.1Q tunnel ports.
- When a port is configured as an IEEE 802.1Q tunnel port, spanning tree bridge protocol data unit (BPDU) filtering is automatically disabled on the interface.

Configuring an IEEE 802.1Q Tunneling Port

Beginning in privileged EXEC mode, follow these steps to configure a port as an IEEE 802.1Q tunnel port:

| | Command | Purpose |
|--------|--|---|
| Step 1 | <code>Router# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# bridge <i>bridge-number</i> protocol <i>bridge-protocol</i></code> | Creates a bridge number and specifies a protocol. |
| Step 3 | <code>Router(config)# interface fastEthernet <i>number</i></code> | Enters the interface configuration mode and the interface to be configured as a tunnel port. This should be the edge port in the service-provider network that connects to the customer switch. Valid interfaces include physical interfaces and port-channel logical interfaces (port channels 1 to 64). |

| | Command | Purpose |
|--------|---|--|
| Step 4 | <code>Router(config-if)# bridge-group number</code> | Assigns the tunnel port to a bridge-group. All traffic from the port (tagged and untagged) will be switched based on this bridge-group. Other members of the bridge-group should be VLAN sub-interfaces on a provider trunk interface. |
| Step 5 | <code>Router(config-if)# mode dot1q-tunnel</code> | Sets the interface as an IEEE 802.1Q tunnel port. |
| Step 6 | <code>Router(config-if)# exit</code> | Returns to global configuration mode. |
| Step 7 | <code>Router(config-if)# end</code> | Returns to privileged EXEC mode. |
| Step 8 | <code>Router# show dot1q-tunnel</code> | Displays the tunnel ports on the switch. |
| Step 9 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

**Note**

The VLAN ID (VID) range of 2 to 4095 is recommended for IEEE 802.1Q tunneling on the ML-Series card.

**Note**

If VID 1 is required, use the following command:

```
Router (config)# VLAN dot1Q tag native
```

Use the **no mode dot1q-tunnel** interface configuration command to remove the IEEE 802.1Q tunnel from the interface.

The following sections show how to configure the example in [Figure 8-1 on page 8-2](#). The first section applies to Router A, and the second section applies to Router B.

Example 8-1 Router A Configuration

```
bridge 30 protocol ieee
bridge 40 protocol ieee
interface FastEthernet0
no ip routing
no ip address
mode dot1q-tunnel
bridge-group 30
!
interface FastEthernet1
no ip address
mode dot1q-tunnel
bridge-group 40
!
interface POS0
no ip address
crc 32
pos flag c2 1
!
interface POS0.1
encapsulation dot1Q 30
bridge-group 30
!
interface POS0.2
encapsulation dot1Q 40
bridge-group 40
```

Example 8-2 Router B Configuration

```

bridge 30 protocol ieee
bridge 40 protocol ieee
interface FastEthernet0
no ip routing
no ip address
mode dot1q-tunnel
bridge-group 30
!
interface FastEthernet1
no ip address
mode dot1q-tunnel
bridge-group 40
!
interface POS0
no ip address
crc 32
pos flag c2 1
!
interface POS0.1
encapsulation dot1Q 30
bridge-group 30
!
interface POS0.2
encapsulation dot1Q 40
bridge-group 40

```

Understanding Layer 2 Protocol Tunneling

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. STP must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure. Cisco Discovery Protocol (CDP) must discover neighboring Cisco devices from local and remote sites. VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

When protocol tunneling is enabled, edge switches on the inbound side of the service-provider infrastructure encapsulate Layer 2 protocol packets with a special MAC address and send them across the service-provider network. Core switches in the network do not process these packets, but forward them as normal packets. Layer 2 protocol data units (PDUs) for CDP, Spanning-Tree Protocol (STP), or Virtual terminal Protocol (VTP) cross the service-provider infrastructure and are delivered to customer switches on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs with the following results:

- Users on each of a customer's sites are able to properly run STP and every VLAN can build a correct spanning tree based on parameters from all sites and not just from the local site.
- CDP discovers and shows information about the other Cisco devices connected through the service-provider network.
- VTP provides consistent VLAN configuration throughout the customer network, propagating through the service provider to all switches.

Layer 2 protocol tunneling can be used independently or to enhance IEEE 802.1Q tunneling. If protocol tunneling is not enabled on IEEE 802.1Q tunneling ports, remote switches at the receiving end of the service-provider network do not receive the PDUs and cannot properly run STP, CDP, and VTP. When protocol tunneling *is* enabled, Layer 2 protocols within each customer's network are totally separate

from those running within the service-provider network. Customer switches on different sites that send traffic through the service-provider network with IEEE 802.1Q tunneling achieve complete knowledge of the customer's VLAN. If IEEE 802.1Q tunneling is not used, you can still enable Layer 2 protocol tunneling by connecting to the customer switch through access ports and enabling tunneling on the service-provider access port.

Configuring Layer 2 Protocol Tunneling

Layer 2 protocol tunneling (by protocol) is enabled on the tunnel ports that are connected to the customer in the edge switches of the service-provider network. ML-Series card tunnel ports are connected to customer IEEE 802.1Q trunk ports. The ML-Series card supports Layer 2 protocol tunneling for CDP, STP, and VTP. The ML-Series cards connected to the customer switch perform the tunneling process.

When the Layer 2 PDUs that entered the inbound ML-Series switch through the tunnel port exit the switch through the trunk port into the service-provider network, the switch overwrites the customer PDU-destination MAC address with a well-known Cisco proprietary multicast address (01-00-0c-cd-cd-d0). If IEEE 802.1Q tunneling is enabled, packets are also double-tagged; the outer tag is the customer metro tag and the inner tag is the customer VLAN tag. The core switches ignore the inner tags and forward the packet to all trunk ports in the same metro VLAN. The ML-Series switches on the outbound side restore the proper Layer 2 protocol and MAC address information and forward the packets. Therefore, the Layer 2 PDUs are kept intact and delivered across the service-provider infrastructure to the other side of the customer network.

This section contains this information about configuring Layer 2 protocol tunneling:

- [Default Layer 2 Protocol Tunneling Configuration, page 8-7](#)
- [Layer 2 Protocol Tunneling Configuration Guidelines, page 8-7](#)
- [Monitoring and Verifying Tunneling Status, page 8-9](#)

Default Layer 2 Protocol Tunneling Configuration

[Table 8-1](#) shows the default Layer 2 protocol tunneling configuration.

Table 8-1 Default Layer 2 Protocol Tunneling Configuration

| Feature | Default Setting |
|------------------------------|--|
| Layer 2 protocol tunneling | Disabled for CDP, STP, and VTP. |
| Class of service (CoS) value | If a CoS value is configured on the interface for data packets, that value is the default used for Layer 2 PDUs. If none is configured, there is no default. This allows existing CoS values to be maintained, unless the user configures otherwise. |

Layer 2 Protocol Tunneling Configuration Guidelines

These are some configuration guidelines and operating characteristics of Layer 2 protocol tunneling:

- The switch supports tunneling of CDP, STP (including multiple STP [MSTP]), and VTP protocols. Protocol tunneling is disabled by default but can be enabled for the individual protocols on IEEE 802.1Q tunnel ports.

- Tunneling is not supported on trunk ports. If you enter the **l2protocol-tunnel** interface configuration command on a trunk port, the command is accepted, but Layer 2 tunneling does not take effect unless you change the port to a tunnel port.
- EtherChannel port groups are compatible with tunnel ports as long as the IEEE 802.1Q configuration is consistent within an EtherChannel port group.
- If an encapsulated PDU (with the proprietary destination MAC address) is received from a tunnel port or access port with Layer 2 tunneling enabled, the tunnel port is shut down to prevent loops.
- Only decapsulated PDUs are forwarded to the customer network. The spanning tree instance running on the service-provider network does not forward BPDUs to tunnel ports. No CDP packets are forwarded from tunnel ports.
- Because tunneled PDUs (especially STP BPDUs) must be delivered to all remote sites for the customer virtual network to operate properly, you can give PDUs higher priority within the service-provider network than data packets received from the same tunnel port. By default, the PDUs use the same CoS value as data packets.
- Protocol tunneling has to be configured symmetrically at both the ingress and egress point. For example, if you configure the entry point to tunnel STP, CDP, VTP, then you must configure the egress point in the same way.

Configuring a Layer 2 Tunneling Port

Beginning in privileged EXEC mode, follow these steps to configure a port as a Layer 2 tunnel port:

| | Command | Purpose |
|---------|--|---|
| Step 1 | <code>Router# conf t</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# bridge <i>bridge-number</i> protocol <i>bridge-protocol</i></code> | Creates a bridge number and specifies a protocol. |
| Step 3 | <code>Router(config)# l2protocol-tunnel cos <i>cos-value</i></code> | Assigns a CoS value or values to associate with the Layer 2 tunneling port. The <i>cos-value</i> is a number from the 0 to 7 range. |
| Step 4 | <code>Router(config)# interface fastEthernet <i>number</i></code> | Enters the interface configuration mode and the interface to be configured as a tunnel port. This should be the edge port in the service-provider network that connects to the customer switch. Valid interfaces include physical interfaces and port-channel logical interfaces (port channels 1 to 64). |
| Step 5 | <code>Router(config-if)# bridge-group <i>bridge-group-number</i></code> | Specifies the default VLAN, which is used if the interface stops trunking. This is VLAN ID specific to the particular customer. |
| Step 6 | <code>Router(config-if)# mode dot1q-tunnel</code> | Sets the interface as an IEEE 802.1Q tunnel port. |
| Step 7 | <code>Router(config-if)# l2 protocol-tunnel [cdp] [stp] [vtp]</code> | Sets the interface as a Layer 2 protocol tunnel port and enables Cisco Discovery Protocol (CDP), Spanning Tree (STP) and VLAN Trunking Protocol (VTP), which are off by default. |
| Step 8 | <code>Router(config-if)# exit</code> | Returns to global configuration mode. |
| Step 9 | <code>Router(config-if)# end</code> | Returns to privileged EXEC mode. |
| Step 10 | <code>Router# show dot1q-tunnel</code> | Displays the tunnel ports on the switch. |
| Step 11 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

Monitoring and Verifying Tunneling Status

Table 8-2 shows the privileged EXEC commands for monitoring and maintaining IEEE 802.1Q and Layer 2 protocol tunneling.

Table 8-2 *Commands for Monitoring and Maintaining Tunneling*

| Command | Purpose |
|---|--|
| <code>show dot1q-tunnel</code> | Displays IEEE 802.1Q tunnel ports on the switch. |
| <code>show dot1q-tunnel interface interface-id</code> | Verifies if a specific interface is a tunnel port. |
| <code>show l2protocol-tunnel</code> | Displays information about Layer 2 protocol tunneling ports. |
| <code>show vlan</code> | Displays IEEE 802.1Q tunnel information. |



Configuring Link Aggregation

This chapter describes how to configure link aggregation for the ML-Series cards, both EtherChannel and Packet-over-SONET/SDH (POS) channel. For additional information about the Cisco IOS commands used in this chapter, refer to the *Cisco IOS Command Reference* publication.

This chapter contains the following major sections:

- [Understanding Link Aggregation, page 9-1](#)
- [Configuring EtherChannel, page 9-2](#)
- [EtherChannel Configuration Example, page 9-3](#)
- [Configuring POS Channel, page 9-4](#)
- [POS Channel Configuration Example, page 9-5](#)
- [Understanding Encapsulation over EtherChannel or POS Channel, page 9-7](#)
- [Configuring Encapsulation over EtherChannel or POS Channel, page 9-7](#)
- [Encapsulation over EtherChannel Example, page 9-7](#)
- [Monitoring and Verifying EtherChannel and POS, page 9-9](#)



Note

You might have already configured bridging, and you may now proceed with configuring link aggregation as an optional step. See [Chapter 5, “Configuring Bridging”](#) for more general bridging information.



Note

The ML-Series does not support the routing of Subnetwork Access Protocol (SNAP) or Inter-Switch Link (ISL) encapsulated frames.

Understanding Link Aggregation

The ML-Series card offers both EtherChannel and POS channel. Traditionally EtherChannel is a trunking technology that groups together multiple full-duplex 802.3 Ethernet interfaces to provide fault-tolerant high-speed links between switches, routers, and servers. EtherChannel is a logical aggregation of multiple Ethernet interfaces. EtherChannel forms a single higher bandwidth routing or bridging endpoint and was designed primarily for host-to-switch connectivity. The ML-Series card extends this link aggregation technology to bridged POS interfaces.

Link aggregation provides the following benefits:

- Logical aggregation of bandwidth
- Load balancing
- Fault tolerance

The EtherChannel interface, consisting of multiple Fast Ethernet, Gigabit Ethernet or POS interfaces, is treated as a single interface, which is called a *port channel*. You must perform all EtherChannel configurations on the EtherChannel interface (port channel) rather than on the individual member Ethernet interfaces. You can create the EtherChannel interface by entering the **interface port-channel** interface configuration command. Each ML100T-12 supports up to 7 Fast EtherChannel (FEC) interfaces or port channels (6 Fast Ethernet and 1 POS). Each ML1000-2 supports up to 2 Gigabit EtherChannel (GEC) interfaces or port channels (1 Gigabit Ethernet and 1 POS.)

EtherChannel connections are fully compatible with IEEE 802.1Q trunking and routing technologies. 802.1Q trunking can carry multiple VLANs across an EtherChannel.

Cisco's FEC technology builds upon standards-based 802.3 full-duplex Fast Ethernet to provide a reliable high-speed solution for the campus network backbone. FEC provides bandwidth scalability within the campus by providing up to 400-Mbps full-duplex Fast Ethernet on the ML100-12.

Cisco's GEC technology provides bandwidth scalability by providing 2-Gbps full-duplex aggregate capacity on the ML1000-2.

Cisco's POS channel technology provide bandwidth scalability by providing up to 48 STSs or VC4-16c of aggregate capacity on either the ML100-12 or the ML1000-2.



Note Link aggregation across multiple ML-Series cards is not supported.



Note Policing is not supported on port channel interfaces.

Configuring EtherChannel

You can configure an FEC or a GEC by creating an EtherChannel interface (port channel) and assigning a network IP address. All interfaces that are members of a FEC or a GEC should have the same link parameters, such as duplex and speed.

To create an EtherChannel interface, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|---------------|--|--|
| Step 1 | Router(config)# interface port-channel <i>channel-number</i> | Creates the EtherChannel interface. You can configure up to 6 FECs on the ML100T-12 and 1 GEC on the ML1000-2. |
| Step 2 | Router(config-if)# ip address <i>ip-address</i> <i>subnet-mask</i> | Assigns an IP address and subnet mask to the EtherChannel interface (required only for Layer 3 EtherChannel). |
| Step 3 | Router(config-if)# end | Exits to privileged EXEC mode. |
| Step 4 | Router# copy running-config startup-config | (Optional) Saves configuration changes to NVRAM. |

For information on other configuration tasks for the EtherChannel, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.

**Caution**

The EtherChannel interface is the Layer 2/Layer 3 interface. Do not enable Layer 3 addresses on the physical interfaces. Do not assign bridge groups on the physical interfaces because doing so creates loops.

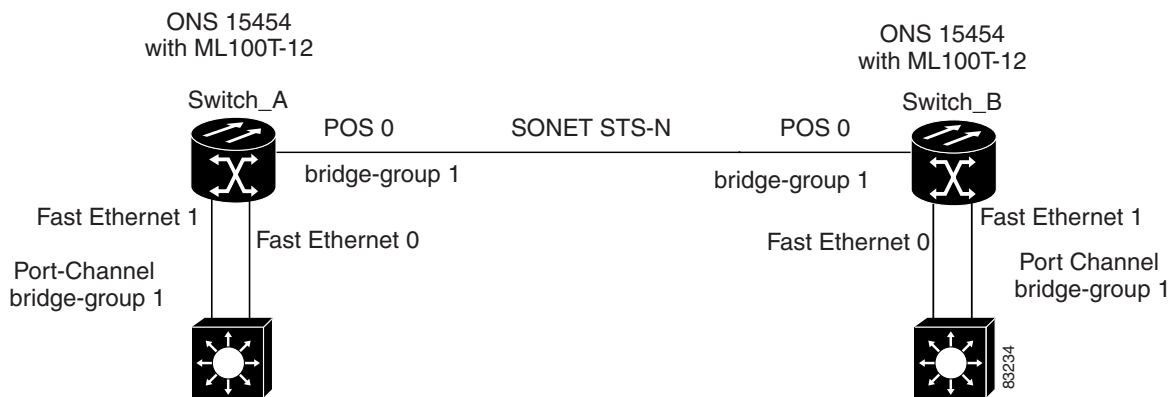
To assign Ethernet interfaces to the EtherChannel, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|---------------|---|--|
| Step 1 | Router(config)# interface fastethernet <i>number</i> or Router(config)# interface gigabitethernet <i>number</i> | Enters one of the interface configuration modes to configure the Fast Ethernet or Gigabit Ethernet interface that you want to assign to the EtherChannel. You can assign any interface on the system to the EtherChannel, but both interfaces must be either FEC or GEC. |
| Step 2 | Router(config-if)# channel-group <i>channel-number</i> | Assigns the Fast Ethernet or Gigabit Ethernet interfaces to the EtherChannel. The channel number must be the same channel number you assigned to the EtherChannel interface. |
| Step 3 | Router(config-if)# end | Exits to privileged EXEC mode. |
| Step 4 | Router# copy running-config startup-config | (Optional) Saves configuration changes to NVRAM. |

EtherChannel Configuration Example

shows an example of encapsulation over EtherChannel. The associated commands are provided in the sections that follow the figure.

Example 9-1 Encapsulation over EtherChannel Example



Example 9-2 Switch A Configuration

```

hostname Switch A
!
bridge 1 protocol ieee
!
interface Port-channel 1
 no ip address
 bridge-group 1
 hold-queue 150 in
!
interface FastEthernet 0
 no ip address
 channel-group 1
!
interface FastEthernet 1
 no ip address
 channel-group 1
!
interface POS 0
 no ip routing
 no ip address
  crc 32
 bridge-group 1
 pos flag c2 1

```

Example 9-3 Switch B Configuration

```

hostname Switch B
!
bridge 1 protocol ieee
!
interface Port-channel 1
 no ip routing
 no ip address
 bridge-group 1
 hold-queue 150 in
!
interface FastEthernet 0
 no ip address
 channel-group 1
!
interface FastEthernet 1
 no ip address
 channel-group 1
!
interface POS 0
 no ip address
  crc 32
 bridge-group 1
 pos flag c2 1
!

```

Configuring POS Channel

You can configure a POS channel by creating a POS channel interface (port channel) and optionally assigning an IP address. All POS interfaces that are members of a POS channel should have the same port properties and be on the same ML-Series card.



Note POS channel is only supported with G-Series card compatible (LEX) encapsulation.

To create a POS channel interface, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|---------------|--|---|
| Step 1 | Router(config)# interface port-channel <i>channel-number</i> | Creates the POS channel interface. You can configure one POS channel on the ML-Series card. |
| Step 2 | Router(config-if)# ip address <i>ip-address</i> <i>subnet-mask</i> | Assigns an IP address and subnet mask to the POS channel interface (required only for the Layer 3 POS channel). |
| Step 3 | Router(config-if)# end | Exits to privileged EXEC mode. |
| Step 4 | Router# copy running-config startup-config | (Optional) Saves configuration changes to NVRAM. |



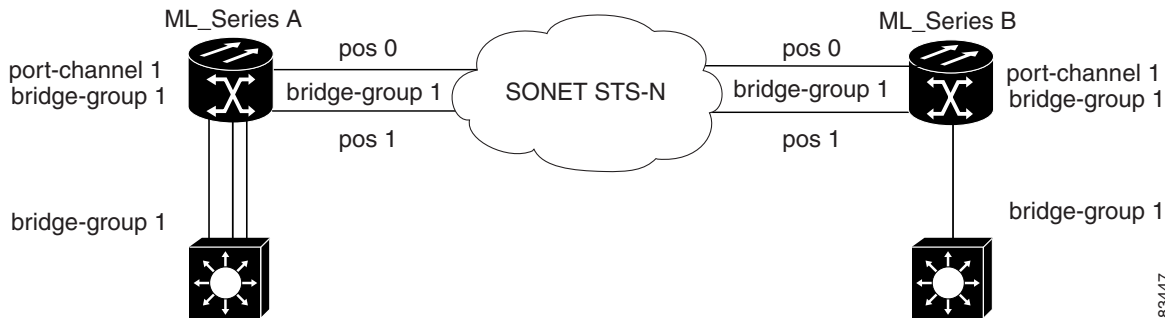
Caution The POS channel interface is the routed interface. Do not enable Layer 3 addresses on any physical interfaces. Do not assign bridge groups on any physical interfaces because doing so creates loops.

To assign POS interfaces to the POS channel, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|---------------|--|--|
| Step 1 | Router(config)# interface pos <i>number</i> | Enters the interface configuration mode to configure the POS interface that you want to assign to the POS channel. |
| Step 2 | Router(config-if)# channel-group <i>channel-number</i> | Assigns the POS interface to the POS channel. The channel number must be the same channel number that you assigned to the POS channel interface. |
| Step 3 | Router(config-if)# end | Exits to privileged EXEC mode. |
| Step 4 | Router# copy running-config startup-config | (Optional) Saves the configuration changes to NVRAM. |

POS Channel Configuration Example

[Example 9-4](#) shows an example of POS channel configuration. The associated code is provided in the sections that follow the figure.

Example 9-4 POS Channel Example

83447

Example 9-5 Switch A Configuration

```

bridge irb
bridge 1 protocol ieee
!
!
interface Port-channell
no ip address
no keepalive
bridge-group 1
!
interface FastEthernet0
no ip address
bridge-group 1
!
interface POS0
no ip address
channel-group 1
crc 32
pos flag c2 1
!
interface POS1
no ip address
channel-group 1
crc 32
pos flag c2 1

```

Example 9-6 Switch B Configuration

```

bridge irb
bridge 1 protocol ieee
!
!
interface Port-channell
no ip address
no keepalive
bridge-group 1
!
interface FastEthernet0
no ip address
bridge-group 1
!

```



```

interface POS0
  no ip address
  channel-group 1
  crc 32
  pos flag c2 1
  !
interface POS1
  no ip address
  channel-group 1
  crc 32
  pos flag c2 1

```

Understanding Encapsulation over EtherChannel or POS Channel

When configuring encapsulation over FEC, GEC, or POS, be sure to configure 802.1Q on the port-channel interface, not its member ports. However, certain attributes of port channel, such as duplex mode, need to be configured at the member port levels. Also make sure that you do not apply protocol-level configuration (such as an IP address or a bridge group assignment) to the member interfaces. All protocol-level configuration should be on the port channel or on its subinterface. You must configure 802.1Q encapsulation on the partner system of the EtherChannel as well.

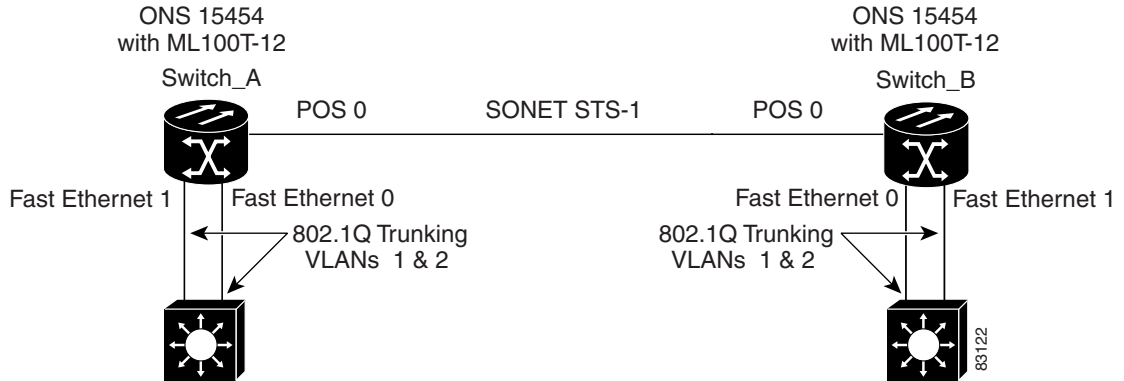
Configuring Encapsulation over EtherChannel or POS Channel

To configure encapsulation over the EtherChannel or POS channel, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|--|---|
| Step 1 | Router(config)# interface port-channel channel-number.subinterface-number | Configures the subinterface on the port channel. |
| Step 2 | Router(config-subif)# encapsulation dot1q vlan-id | Assigns the 802.1Q encapsulation to the subinterface. |
| Step 3 | Router(config-subif)# bridge-group bridge-group-number | Assigns an interface to a bridge group. |
| Step 4 | Router(config-subif)# end | Exits to privileged EXEC mode. Note Optionally, you can remain in interface configuration mode and enable other supported interface commands to meet your requirements. |
| Step 5 | Router# copy running-config startup-config | (Optional) Saves the configuration changes to NVRAM. |

Encapsulation over EtherChannel Example

Figure 9-1 shows an example of encapsulation over EtherChannel. The associated code is provided in Example 9-7 and Example 9-8.

Figure 9-1 Encapsulation over EtherChannel Example

This encapsulation over EtherChannel example shows how to set up two ONS 15454s with ML100T-12 cards (Switch A and Switch B) to interoperate with two switches that also support 802.1Q encapsulation over EtherChannel. To set up this example, use the configurations in the following sections for both Switch A and Switch B.

Example 9-7 Switch A Configuration

```
hostname Switch A
!
bridge irb
bridge 1 protocol ieee
bridge 2 protocol ieee
!
interface Port-channel1
 no ip address
 hold-queue 150 in
!
interface Port-channel1.1
 encapsulation dot1Q 1 native
bridge-group 1
!
interface Port-channel1.2
 encapsulation dot1Q 2
bridge-group 2

!
interface FastEthernet0
 no ip address
 channel-group 1
!
interface FastEthernet1
 no ip address
 channel-group 1
!
interface POS0
 no ip address
 crc 32
 pos flag c2 1
!
interface POS0.1
 encapsulation dot1Q 1 native
bridge-group 1
!
```

```
interface POS0.2
 encapsulation dot1Q 2
 bridge-group 2
```

Example 9-8 Switch B Configuration

```
hostname Switch B
!
bridge irb
bridge 1 protocol ieee
bridge 2 protocol ieee
!
interface Port-channel1
 no ip address
 hold-queue 150 in
!
interface Port-channel1.1
 encapsulation dot1Q 1 native
 bridge-group 1
!
interface Port-channel1.2
 encapsulation dot1Q 2
 bridge-group 2
!
interface FastEthernet0
 no ip address
 channel-group 1
!
interface FastEthernet1
 no ip address
 channel-group 1
!
interface POS0
 no ip address
 crc 32
 pos flag c2 1
!
interface POS0.1
 encapsulation dot1Q 1 native
 bridge-group 1
!
interface POS0.2
 encapsulation dot1Q 2
 bridge-group 2
!
```

Monitoring and Verifying EtherChannel and POS

After FEC, GEC, or POS is configured, you can monitor its status using the **show interfaces port-channel** command.

Example 9-9 show interfaces port-channel command

```
Router# show int port-channel 1
Port-channel1 is up, line protocol is up
 Hardware is FEChannel, address is 0005.9a39.6634 (bia 0000.0000.0000)
 MTU 1500 bytes, BW 200000 Kbit, DLY 100 usec,
 reliability 255/255, txload 1/255, rxload 1/255
```

```
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Unknown duplex, Unknown Speed
ARP type: ARPA, ARP Timeout 04:00:00
  No. of active members in this channel: 2
    Member 0 : FastEthernet0 , Full-duplex, Auto Speed
    Member 1 : FastEthernet1 , Full-duplex, Auto Speed
Last input 00:00:01, output 00:00:23, output hang never
Last clearing of "show interface" counters never
Input queue: 0/150/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue :0/80 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  820 packets input, 59968 bytes
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 watchdog, 0 multicast
  0 input packets with dribble condition detected
  32 packets output, 11264 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out.
```



Configuring Networking Protocols

This chapter describes networking protocol configurations for the ML-Series cards. It provides initial configuration information so you can get your ML-Series card up and running. For more information about the Cisco IOS commands used in this chapter, refer to the *Cisco IOS Command Reference* publication.

This chapter contains the following major sections:

- [Understanding IP Routing Protocols, page 10-1](#)
- [Basic IP Routing Protocol Configuration, page 10-2](#)
- [Configuring RIP, page 10-5](#)
- [Configuring OSPF, page 10-9](#)
- [Configuring EIGRP, page 10-21](#)
- [Configuring BGP, page 10-28](#)
- [Configuring IS-IS, page 10-30](#)
- [Configuring Static Routes, page 10-32](#)
- [Monitoring Static Routes, page 10-33](#)
- [Monitoring and Maintaining the IP Network, page 10-33](#)
- [Understanding IP Multicast Routing, page 10-34](#)
- [Configuring IP Multicast Routing, page 10-35](#)
- [Monitoring and Verifying IP Multicast Operation, page 10-35](#)



Note

Complete the general interface configuration in the “[Configuring Interfaces](#)” chapter before proceeding with configuring networking and routing protocols.

Understanding IP Routing Protocols

This section describes how to configure the ML-Series card for supported IP routing protocols. It is intended to provide enough information for a network administrator to get the protocols up and running. However, this section does not provide in-depth configuration detail for each protocol. For detailed information, refer to the *Cisco IOS IP and IP Routing Configuration Guide* and the *Cisco IOS IP and IP Routing Command Reference* publications.

IP routing is enabled by default on the ML-Series card.

For IP routing, you need the following to configure your interface:

- IP address
- IP subnet mask

You also need to do the following:

- Select a routing protocol.
- Assign IP network numbers to be advertised.

The ML Series supports the routing protocols listed and described in the following sections.

Basic IP Routing Protocol Configuration

To configure IP routing protocols to run on a Fast Ethernet, Gigabit Ethernet, or POS interface, perform one of the following procedures, depending on the protocol you are configuring.

RIP

To configure the RIP protocol, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|--|---|
| Step 1 | Router(config)# router rip | Enters router configuration mode, defines RIP as the routing protocol, and starts the RIP routing process. |
| Step 2 | Router(config-router)# network <i>net-number</i> | Specifies a directly connected network based on the Internet Network Information Center (InterNIC) network number—not a subnet number or individual address. The routing process associates interfaces with the appropriate addresses and begins processing packets on the specified network. |
| Step 3 | Router(config-router)# exit | Returns to global configuration mode. |

EIGRP

To configure the EIGRP protocol, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|--|---|
| Step 1 | Router(config)# router eigrp <i>autonomous-system-number</i> | Defines EIGRP as the IP routing protocol. The autonomous system number is the autonomous system to which this ML-Series card belongs. |
| Step 2 | Router(config-router)# network <i>net-number</i> | Defines the directly connected networks that run EIGRP. The network number is the number of the network that is advertised by this ML-Series card. |
| Step 3 | Router(config-router)# exit | Returns to global configuration mode. |

OSPF

To configure the OSPF protocol, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|--|---|
| Step 1 | Router(config)# router ospf <i>process-ID</i> | Defines OSPF as the IP routing protocol. The process ID identifies a unique OSPF router process. This number is internal to the ML-Series card only; the process ID here does not have to match the process IDs on other routers. |
| Step 2 | Router(config-router)# network <i>net-address wildcard-mask area area-ID</i> | Assigns an interface to a specific area. <ul style="list-style-type: none"> • The network address is the address of directly connected networks or subnets. • The wildcard mask is an inverse mask that compares a given address with interface addressing to determine whether OSPF uses this interface. • The area parameter identifies the interface as belonging to an area. • The area ID specifies the area associated with the network address. |
| Step 3 | Router(config-router)# end | Returns to privileged EXEC mode. |

BGP

To configure the BGP protocol, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|--|---|
| Step 1 | Router(config)# router bgp <i>autonomous-system-number</i> | Defines BGP as the IP routing protocol. The autonomous system number is the autonomous system to which this ML-Series card belongs. |
| Step 2 | Router(config-router) # network <i>net-number</i> | Defines the directly connected networks that run BGP. The network number is the number of the network that is advertised by this ML-Series card. |
| Step 3 | Router(config-router)# exit | Returns to global configuration mode. |

Enabling IP Routing

By default, IP routing is enabled. Beginning in privileged EXEC mode, follow this procedure to enable IP routing:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router# configure terminal | Enters global configuration mode. |
| Step 2 | Router(config)# ip routing | Enables IP routing (default). |
| Step 3 | Router(config)# router <i>ip-routing-protocol</i> | Specifies an IP routing protocol. This step might include other commands, such as specifying the networks to route with the network (RIP) router configuration command. For information about specific protocols, refer to sections later in this chapter and to the <i>Cisco IOS IP and IP Routing Configuration Guide</i> . |
| Step 4 | Router(config-router)# end | Returns to privileged EXEC mode. |
| Step 5 | Router(config)# show running-config | Verifies your entries. |
| Step 6 | Router(config)# copy running-config startup-config | (Optional) Saves your entries in the configuration file. |

Use the **no ip routing** global configuration command to disable routing.

Example 10-1 Enabling IP Routing Using RIP as the Routing Protocol:

```
Router# configure terminal
Router(config)# ip routing
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# end
```


Configuring IP Routing

You can now set up parameters for the selected routing protocols as described in these sections:

- [Configuring RIP, page 10-5](#)
- [Configuring OSPF, page 10-9](#)
- [Configuring EIGRP, page 10-21](#)
- [Configuring BGP, page 10-28](#)
- [Configuring IS-IS, page 10-30](#)
- [Configuring Static Routes, page 10-32](#)

Configuring RIP

The Routing Information Protocol (RIP) is an Interior Gateway Protocol (IGP) created for use in small, homogeneous networks. It is a distance-vector routing protocol that uses broadcast User Datagram Protocol (UDP) data packets to exchange routing information. The protocol is documented in RFC 1058. You can find detailed information about RIP in *IP Routing Fundamentals*, published by Cisco Press.

Using RIP, the switch sends routing information updates (advertisements) every 30 seconds. If a router does not receive an update from another router for 180 seconds or more, it marks the routes served by that router as unusable. If there is still no update after 240 seconds, the router removes all routing table entries for the non-updating router.

RIP uses hop counts to rate the value of different routes. The hop count is the number of routers that can be traversed in a route. A directly connected network has a hop count of zero; a network with a hop count of 16 is unreachable. This small range (0 to 15) makes RIP unsuitable for large networks.

If the router has a default network path, RIP advertises a route that links the router to the pseudo network 0.0.0.0. The 0.0.0.0 network does not exist; it is treated by RIP as a network to implement the default routing feature. The switch advertises the default network if a default was learned by RIP or if the router has a gateway of last resort and RIP is configured with a default metric. RIP sends updates to the interfaces in specified networks. If an interface's network is not specified, it is not advertised in any RIP update.

[Table 10-1](#) shows the default RIP configuration.

Table 10-1 Default RIP Configuration

| Feature | Default Setting |
|---------------------------------|--|
| Auto summary | Enabled |
| Default-information originate | Disabled |
| Default metric | Built-in; automatic metric translations |
| IP RIP authentication key-chain | No authentication Authentication mode: clear text |
| IP RIP receive version | According to the version router configuration command |
| IP RIP send version | According to the version router configuration command |
| IP RIP triggered | According to the version router configuration command |
| IP split horizon | Varies with media |

Table 10-1 Default RIP Configuration (continued)

| Feature | Default Setting |
|------------------------|--|
| Neighbor | None defined |
| Network | None specified |
| Offset list | Disabled |
| Output delay | 0 milliseconds |
| Timers basic | Update: 30 seconds Invalid: 180 seconds Hold-down: 180 seconds Flush: 240 seconds |
| Validate-update-source | Enabled |
| Version | Receives RIP Version 1 and Version 2 packets; sends Version 1 packets |

To configure RIP, enable RIP routing for a network and optionally configure other parameters. Beginning in privileged EXEC mode, follow this procedure to enable and configure RIP:

| | Command | Purpose |
|--------|---|---|
| Step 1 | <code>Router# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# ip routing</code> | Enables IP routing. (Required only if IP routing is disabled.) |
| Step 3 | <code>Router(config)# router rip</code> | Enables a RIP routing process, and enters router configuration mode. |
| Step 4 | <code>Router(config-router)# network network number</code> | Associates a network with a RIP routing process. You can specify multiple network commands. RIP routing updates are sent and received through interfaces only on these networks. |
| Step 5 | <code>Router(config-router)# neighbor ip-address</code> | (Optional) Defines a neighboring router with which to exchange routing information. This step allows routing updates from RIP (normally a broadcast protocol) to reach nonbroadcast networks. |
| Step 6 | <code>Router(config-router)# offset list [access-list number name] {in out} offset [type number]</code> | (Optional) Applies an offset list to routing metrics to increase incoming and outgoing metrics to routes learned through RIP. You can limit the offset list with an access list or an interface. |
| Step 7 | <code>Router(config-router)# timers basic update invalid holddown flush</code> | (Optional) Adjusts routing protocol timers. Valid ranges for all timers are 0 to 4294967295 seconds. <ul style="list-style-type: none"> <i>update</i>—The time (in seconds) between sending of routing updates. The default is 30 seconds. <i>invalid</i>—The timer interval (in seconds) after which a route is declared invalid. The default is 180 seconds. <i>holddown</i>—The time (in seconds) that must pass before a route is removed from the routing table. The default is 180 seconds. <i>flush</i>—The amount of time (in seconds) for which routing updates are postponed. The default is 240 seconds. |

| | Command | Purpose |
|---------|---|---|
| Step 8 | <code>Router(config-router)# version {1 2}</code> | (Optional) Configures the switch to receive and send only RIP Version 1 or RIP Version 2 packets. By default, the switch receives Version 1 and 2 but sends only Version 1. You can also use the interface commands <code>ip rip {send receive} version {1 2 1 2}</code> to control what versions are used for sending and receiving on interfaces. |
| Step 9 | <code>Router(config-router)# no auto summary</code> | (Optional) Disables automatic summarization. By default, the switch summarizes subprefixes when crossing classful network boundaries. Disables summarization (RIP Version 2 only) to advertise subnet and host routing information to classful network boundaries. |
| Step 10 | <code>Router(config-router)# no validate-update-source</code> | (Optional) Disables validation of the source IP address of incoming RIP routing updates. By default, the switch validates the source IP address of incoming RIP routing updates and discards the update if the source address is not valid. Under normal circumstances, disabling this feature is not recommended. However, if you have a router that is off-network and you want to receive its updates, you can use this command. |
| Step 11 | <code>Router(config-router)# output-delay delay</code> | (Optional) Adds interpacket delay for RIP updates sent. By default, packets in a multiple-packet RIP update have no delay added between packets. If you are sending packets to a lower-speed device, you can add an interpacket delay in the range of 8 to 50 milliseconds. |
| Step 12 | <code>Router(config-router)# end</code> | Returns to privileged EXEC mode. |
| Step 13 | <code>Router# show ip protocols</code> | Verifies your entries. |
| Step 14 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

To turn off the RIP routing process, use the **no router rip** global configuration command.

To display the parameters and current state of the active routing protocol process, use the **show ip protocols** privileged EXEC command.

Example 10-2 show ip protocols command output (showing RIP processes)

```
Router# show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 15 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface          Send Recv Triggered RIP Key-chain
  FastEthernet0       1    1 2
  POS0                 1    1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    192.168.2.0
    192.168.3.0
  Routing Information Sources:
    Gateway           Distance      Last Update
  192.168.2.1         120          00:00:23
  Distance: (default is 120)
```

Use the **show ip rip database** privileged EXEC command to display summary address entries in the RIP database.

```
Router# show ip rip database
192.168.1.0/24    auto-summary
192.168.1.0/24
    [1] via 192.168.2.1, 00:00:24, POS0
192.168.2.0/24    auto-summary
192.168.2.0/24    directly connected, POS0
192.168.3.0/24    auto-summary
192.168.3.0/24    directly connected, FastEthernet0
```

RIP Authentication

RIP Version 1 does not support authentication. If you are sending and receiving RIP Version 2 packets, you can enable RIP authentication on an interface. The key chain determines the set of keys that can be used on the interface. If a key chain is not configured, no authentication is performed, not even the default.

The switch supports two modes of authentication on interfaces for which RIP authentication is enabled: plain text and MD5. The default is plain text.

Beginning in privileged EXEC mode, follow this procedure to configure RIP authentication on an interface:

| | Command | Purpose |
|--------|--|---|
| Step 1 | Router# configure terminal | Enters global configuration mode. |
| Step 2 | Router(config)# interface <i>interface-id</i> | Enters interface configuration mode, and specifies the interface to configure. |
| Step 3 | Router(config-if)# ip rip authentication key-chain <i>name-of-chain</i> | Enables RIP authentication. |
| Step 4 | Router(config-if)# ip rip authentication mode {text md5} | Configures the interface to use plain text authentication (the default) or MD5 digest authentication. |
| Step 5 | Router(config-if)# end | Returns to privileged EXEC mode. |
| Step 6 | Router# show running-config interface [<i>interface-id</i>] | Verifies your entries. |
| Step 7 | Router# copy running-config startup-config | (Optional) Saves your entries in the configuration file. |

To restore clear text authentication, use the **no ip rip authentication mode** interface configuration command. To prevent authentication, use the **no ip rip authentication key-chain** interface configuration command.

Summary Addresses and Split Horizon

Routers connected to broadcast-type IP networks and using distance-vector routing protocols normally use the split-horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router on any interface from which that information originated. This feature usually optimizes communication among multiple routers, especially when links are broken.

**Note**

In general, disabling split horizon is not recommended unless you are certain that your application requires it to properly advertise routes.

If you want to configure an interface running RIP to advertise a summarized local IP address pool on a network access server for dial-up clients, use the **ip summary-address rip** interface configuration command.

Beginning in privileged EXEC mode, follow these steps to set an interface to advertise a summarized local IP address pool and to disable split horizon on the interface:

| | Command | Purpose |
|--------|--|--|
| Step 1 | Router# configure terminal | Enters global configuration mode. |
| Step 2 | Router(config)# interface <i>interface-id</i> | Enters interface configuration mode, and specifies the Layer 3 interface to configure. |
| Step 3 | Router(config-if)# ip address <i>ip-address subnet-mask</i> | Configures the IP address and IP subnet. |
| Step 4 | Router(config-if)# ip summary-address rip <i>ip address</i> <i>ip-network mask</i> | Configures the IP address to be summarized and the IP network mask. |
| Step 5 | Router(config-if)# no ip split horizon | Disables split horizon on the interface. |
| Step 6 | Router(config-if)# end | Returns to privileged EXEC mode. |
| Step 7 | Router# show ip interface <i>interface-id</i> | Verifies your entries. |
| Step 8 | Router# copy running-config startup-config | (Optional) Saves your entries in the configuration file. |

To disable IP summarization, use the **no ip summary-address rip** router configuration command.

**Note**

If split horizon is enabled, neither autosummary nor interface summary addresses (those configured with the **ip summary-address rip** router configuration command) are advertised.

Configuring OSPF

This section briefly describes how to configure Open Shortest Path First (OSPF) Protocol. For a complete description of the OSPF commands, refer to the “OSPF Commands” chapter of the *Cisco IOS IP and IP Routing Command Reference publication*.

OSPF is an IGP designed expressly for IP networks, supporting IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets. The Cisco implementation supports RFC 1253, the OSPF Management Information Base (MIB).

The Cisco implementation conforms to the OSPF Version 2 specifications with these key features:

- Stub areas—Definition of stub areas is supported.

- Route redistribution—Routes learned through any IP routing protocol can be redistributed into another IP routing protocol. At the intradomain level, this means that OSPF can import and export routes learned through protocols such as EIGRP and RIP.
- Authentication—Plain text and message-digest key (MD5) authentication among neighboring routers within an area are supported.
- Routing interface parameter—Configurable parameters supported include interface output cost, retransmission interval, interface transmit delay, router priority, router dead and hello intervals, and authentication key.
- Virtual links—Virtual links are supported.
- Not-so-stubby-area (NSSA)—RFC 1587.

OSPF typically requires coordination among many internal routers, *area border routers* (ABRs) connected to multiple areas, and *autonomous system boundary routers* (ASBRs). The minimum configuration would use all default parameter values, no authentication, and interfaces assigned to areas. If you customize your environment, you must ensure coordinated configuration of all routers.

Table 10-2 shows the default OSPF configuration.

Table 10-2 Default OSPF Configuration

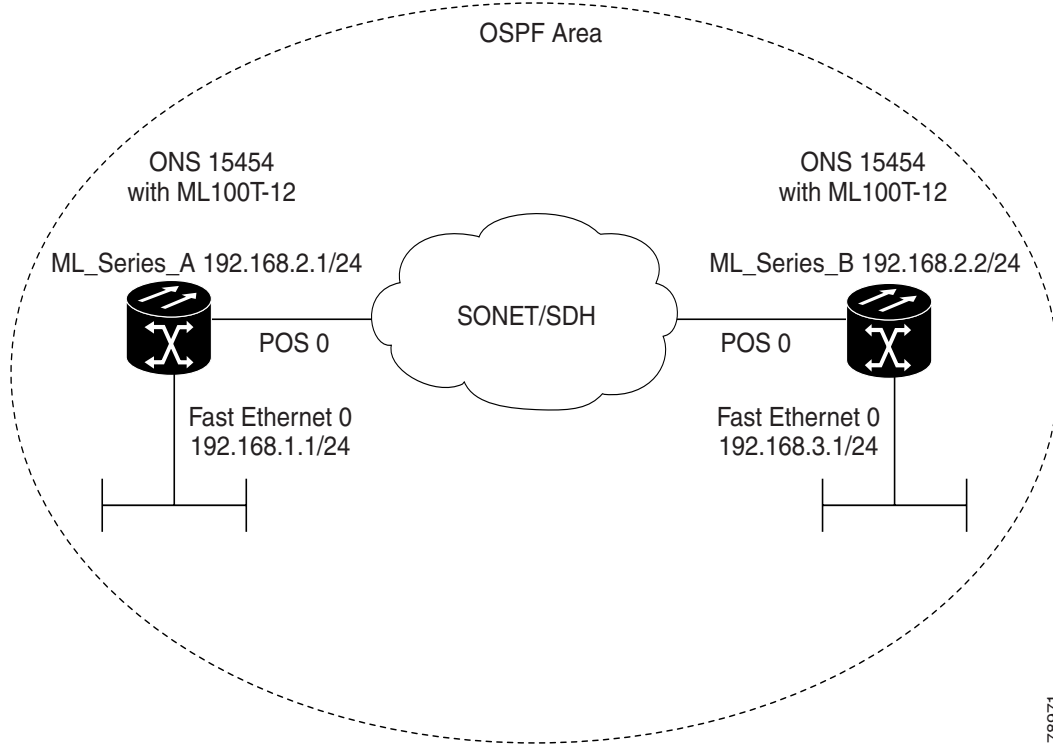
| Feature | Default Setting |
|-------------------------------|--|
| Interface parameters | Cost: No default cost predefined. Retransmit interval: 5 seconds. Transmit delay: 1 second. Priority: 1. Hello interval: 10 seconds. Dead interval: 4 times the hello interval. No authentication. No password specified. MD5 authentication disabled. |
| Area | Authentication type: 0 (no authentication). Default cost: 1. Range: Disabled. Stub: No stub area defined. NSSA: No NSSA area defined. |
| Auto cost | 100 Mbps. |
| Default-information originate | Disabled. When enabled, the default metric setting is 10, and the external route type default is Type 2. |
| Default metric | Built-in, automatic metric translation, as appropriate for each routing protocol. |
| Distance OSPF | dist1 (all routes within an area): 110. dist2 (all routes from one area to another): 110. and dist3 (routes from other routing domains): 110. |
| OSPF database filter | Disabled. All outgoing link-state advertisements (LSAs) are flooded to the interface. |

Table 10-2 Default OSPF Configuration (continued)

| Feature | Default Setting |
|----------------------------------|--|
| IP OSPF name lookup | Disabled. |
| Log adjacency changes | Enabled. |
| Neighbor | None specified. |
| Neighbor database filter | Disabled. All outgoing LSAs are flooded to the neighbor. |
| Network area | Disabled. |
| Router ID | No OSPF routing process defined. |
| Summary address | Disabled. |
| Timers LSA group pacing | 240 seconds. |
| Timers shortest path first (spf) | spf delay: 5 seconds. spf-holdtime: 10 seconds. |
| Virtual link | No area ID or router ID defined. Hello interval: 10 seconds. Retransmit interval: 5 seconds. Transmit delay: 1 second. Dead interval: 40 seconds. Authentication key: No key predefined. MD5: No key predefined. |

Figure 10-1 shows an example of an IP routing protocol using OSPF.

Figure 10-1 IP Routing Protocol Example Using OSPF



78971

Enabling OSPF requires that you create an OSPF routing process, specify the range of IP addresses to be associated with the routing process, and assign area IDs to be associated with that range.

Beginning in privileged EXEC mode, follow this procedure to enable OSPF:

| | Command | Purpose |
|--------|---|--|
| Step 1 | <code>Router#configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# router ospf process-id</code> | Enables OSPF routing, and enters router configuration mode. The process ID is an internally used identification parameter that is locally assigned and can be any positive integer. Each OSPF routing process has a unique value. |
| Step 3 | <code>Router(config)# network address wildcard-mask area area-id</code> | Defines an interface on which OSPF runs and the area ID for that interface. Use the wildcard-mask to use a single command to define one or more multiple interfaces to be associated with a specific OSPF area. The area ID can be a decimal value or an IP address. |
| Step 4 | <code>Router(config)# end</code> | Returns to privileged EXEC mode. |
| Step 5 | <code>Router# show ip protocols</code> | Verifies your entries. |
| Step 6 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

To terminate an OSPF routing process, use the `no router ospf process-id` global configuration command.

Example 10-3 Configuring an OSPF Routing Process and Assigning a Process Number of 1

```
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 0
```

Example 10-4 show ip protocols privileged EXEC command output (verifying the OSPF process ID)

```
Router# show ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.3.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    192.168.2.0 0.0.0.255 area 0
    192.168.3.0 0.0.0.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.3.1         110          00:03:34
    192.168.2.1         110          00:03:34
  Distance: (default is 110)
```

OSPF Interface Parameters

You can use the **ip ospf** interface configuration commands to modify interface-specific OSPF parameters. You are not required to modify any of these parameters, but some interface parameters (hello interval, dead interval, and authentication key) must be consistent across all routers in an attached network. If you modify these parameters, be sure all routers in the network have compatible values.

**Note**

The **ip ospf** interface configuration commands are all optional.

Beginning in privileged EXEC mode, follow these steps to modify OSPF interface parameters:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router# configure terminal | Enters global configuration mode. |
| Step 2 | Router(config)# interface interface-id | Enters interface configuration mode, and specifies the Layer 3 interface to configure. |
| Step 3 | Router(config-if)# ip ospf cost | (Optional) Explicitly specifies the cost of sending a packet on the interface. |
| Step 4 | Router(config-if)# ip ospf retransmit-interval seconds | (Optional) Specifies the number of seconds between link state advertisement transmissions. The range is 1 to 65535 seconds. The default is 5 seconds. |
| Step 5 | Router(config-if)# ip ospf transmit-delay seconds | (Optional) Sets the estimated number of seconds to wait before sending a link state update packet. The range is 1 to 65535 seconds. The default is 1 second. |
| Step 6 | Router(config-if)# ip ospf priority number | (Optional) Sets priority to help determine the OSPF designated router for a network. The range is from 0 to 255. The default is 1. |

| | Command | Purpose |
|---------|--|---|
| Step 7 | <code>Router(config-if)# ip ospf hello-interval seconds</code> | (Optional) Sets the number of seconds between hello packets sent on an OSPF interface. The value must be the same for all nodes on a network. The range is 1 to 65535 seconds. The default is 10 seconds. |
| Step 8 | <code>Router(config-if)# ip ospf dead-interval seconds</code> | (Optional) Sets the number of seconds after the last device hello packet was seen before its neighbors declare the OSPF router to be down. The value must be the same for all nodes on a network. The range is 1 to 65535 seconds. The default is 4 times the hello interval. |
| Step 9 | <code>Router(config-if)# ip ospf authentication-key key</code> | (Optional) Assigns a password to be used by neighboring OSPF routers. The password can be any string of keyboard-entered characters up to 8 bytes in length. All neighboring routers on the same network must have the same password to exchange OSPF information. |
| Step 10 | <code>Router(config-if)# ip ospf message-digest-key keyid md5 key</code> | (Optional) Enables MDS authentication. <ul style="list-style-type: none"> • <i>keyid</i>—Identifier from 1 to 255. • <i>key</i>—Alphanumeric password of up to 16 bytes. |
| Step 11 | <code>Router(config-if)# ip ospf database-filter all out</code> | (Optional) Blocks flooding of OSPF LSA packets to the interface. By default, OSPF floods new LSAs over all interfaces in the same area, except the interface on which the LSA arrives. |
| Step 12 | <code>Router(config-if)# end</code> | Returns to privileged EXEC mode. |
| Step 13 | <code>Router# show ip ospf interface [interface-name]</code> | Displays OSPF-related interface information. |
| Step 14 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

Use the **no** form of these commands to remove the configured parameter value or return to the default value.

Example 10-5 show ip ospf interface privileged EXEC command output

```
Router# show ip ospf interface
FastEthernet0 is up, line protocol is up
  Internet Address 192.168.3.1/24, Area 0
  Process ID 1, Router ID 192.168.3.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.3.1, Interface address 192.168.3.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
POS0 is up, line protocol is up
  Internet Address 192.168.2.2/24, Area 0
  Process ID 1, Router ID 192.168.3.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
```

```
Designated Router (ID) 192.168.3.1, Interface address 192.168.2.2
Backup Designated router (ID) 192.168.2.1, Interface address 192.168.2.1
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Hello due in 00:00:05
Index 1/1, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 2, maximum is 2
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 192.168.2.1 (Backup Designated Router)
Suppress hello for 0 neighbor(s)
```

OSPF Area Parameters

You can optionally configure several OSPF area parameters. These parameters include authentication for password-based protection against unauthorized access to an area, stub areas, and not-so-stubby-areas (NSSAs). *Stub areas* are areas into which information about external routes is not sent. Instead, the area border router (ABR) generates a default external route into the stub area for destinations outside the autonomous system (AS). An NSSA does not flood all LSAs from the core into the area, but can import AS external routes within the area by redistribution.

Route summarization is the consolidation of advertised addresses into a single summary route to be advertised by other areas. If network numbers are contiguous, you can use the **area range** router configuration command to configure the ABR to advertise a summary route that covers all networks in the range.



Note

The OSPF **area** router configuration commands are all optional.

Beginning in privileged EXEC mode, follow these steps to configure area parameters:

| | Command | Purpose |
|---------|--|--|
| Step 1 | <code>Router# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)#router ospf process-id</code> | Enables OSPF routing, and enters router configuration mode. |
| Step 3 | <code>Router(config)# area area-id authentication</code> | (Optional) Allows password-based protection against unauthorized access to the identified area. The identifier can be either a decimal value or an IP address. |
| Step 4 | <code>Router(config)# area area-id authentication message-digest</code> | (Optional) Enables MD5 authentication on the area. |
| Step 5 | <code>Router(config)# area area-id stub [no-summary]</code> | (Optional) Defines an area as a stub area. The no-summary keyword prevents an ABR from sending summary link advertisements into the stub area. |
| Step 6 | <code>Router(config)# area area-id nssa {no-redistribution default-informati on-originate no-summary}</code> | (Optional) Defines an area as a not-so-stubby-area. Every router within the same area must agree that the area is NSSA. Select one of these keywords: <ul style="list-style-type: none"> • no-redistribution—Select when the router is an NSSA ABR and you want the redistribute command to import routes into normal areas, but not into the NSSA. • default-information-originate—Select on an ABR to allow importing type 7 LSAs into the NSSA. • no-redistribution—Select to not send summary LSAs into the NSSA. |
| Step 7 | <code>Router(config)# area area-id range address-mask</code> | (Optional) Specifies an address range for which a single route is advertised. Use this command only with area border routers. |
| Step 8 | <code>Router(config)# end</code> | Returns to privileged EXEC mode. |
| Step 9 | <code>Router# show ip ospf [process-id]</code> <code>show ip ospf [process-id [area-id]] database</code> | Displays information about the OSPF routing process in general or for a specific process ID to verify configuration. Displays lists of information related to the OSPF database for a specific router. |
| Step 10 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

Use the **no** form of these commands to remove the configured parameter value or to return to the default value.

Example 10-6 show ip ospf database and show ip ospf privileged EXEC command outputs

```
Router# show ip ospf database
```

```
OSPF Router with ID (192.168.3.1) (Process ID 1)
```

```
Router Link States (Area 0)
```

| Link ID | ADV Router | Age | Seq# | Checksum | Link count |
|-------------|-------------|-----|------------|----------|------------|
| 192.168.2.1 | 192.168.2.1 | 428 | 0x80000003 | 0x004AB8 | 2 |
| 192.168.3.1 | 192.168.3.1 | 428 | 0x80000003 | 0x006499 | 2 |

Net Link States (Area 0)

| Link ID | ADV Router | Age | Seq# | Checksum |
|-------------|-------------|-----|------------|----------|
| 192.168.2.2 | 192.168.3.1 | 428 | 0x80000001 | 0x00A4E0 |

Router# **show ip ospf**

```

Routing Process "ospf 1" with ID 192.168.3.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x000000
Number of opaque AS LSA 0. Checksum Sum 0x000000
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
  Area BACKBONE(0)
    Number of interfaces in this area is 2
    Area has no authentication
    SPF algorithm executed 4 times
    Area ranges are
    Number of LSA 3. Checksum Sum 0x015431
    Number of opaque link LSA 0. Checksum Sum 0x000000
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0

```

Other OSPF Behavior Parameters

You can optionally configure other OSPF parameters in router configuration mode:

- **Route summarization:** When redistributing routes from other protocols, each route is advertised individually in an external LSA. To help decrease the size of the OSPF link state database, you can use the **summary-address** router configuration command to advertise a single router for all the redistributed routes included in a specified network address and mask.
- **Virtual links:** In OSPF, all areas must be connected to a backbone area. You can establish a virtual link in case of a backbone-continuity break by configuring two ABRs as endpoints of a virtual link. Configuration information includes the identity of the other virtual endpoint (the other ABR) and the nonbackbone link that the two routers have in common (the transit area). Virtual links cannot be configured through a stub area.
- **Default route:** When you specifically configure redistribution of routes into an OSPF routing domain, the route automatically becomes an autonomous system boundary router (ASBR). You can force the ASBR to generate a default route into the OSPF routing domain.
- **Domain Name Server (DNS) names for use in all OSPF **show** privileged EXEC command displays** make it easier to identify a router than displaying it by router ID or neighbor ID.
- **Default Metrics:** OSPF calculates the OSPF metric for an interface according to the bandwidth of the interface. The metric is calculated as *ref-bw* divided by bandwidth, where *ref* is 10 by default, and bandwidth (*bw*) is determined by the **bandwidth** interface configuration command. For multiple links with high bandwidth, you can specify a larger number to differentiate the cost on those links.

- **Administrative distance:** This is a rating of the trustworthiness of a routing information source, an integer between 0 and 255, with a higher value meaning a lower trust rating. An administrative distance of 255 means that the routing information source cannot be trusted at all and should be ignored. OSPF uses three different administrative distances: routes within an area (interarea), routes to another area (interarea), and routes from another routing domain learned through redistribution (external). You can change any of the distance values.
- **Passive interfaces:** Because interfaces between two devices on an Ethernet represent only one network segment, to prevent OSPF from sending hello packets for the sending interface, you must configure the sending device to be a passive interface. Both devices can identify each other through the hello packet for the receiving interface.
- **Route calculation timers:** You can configure the delay time between when OSPF receives a topology change and when it starts the shortest path first (SPF) calculation. You can also configure the hold time between two SPF calculations.
- **Log neighbor changes:** You can configure the router to send a syslog message when an OSPF neighbor state changes, providing a high-level view of changes in the router.

Beginning in privileged EXEC mode, follow this procedure to configure these OSPF parameters:

| | Command | Purpose |
|---------------|--|---|
| Step 1 | <code>Router# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# router ospf process-id</code> | Enables OSPF routing, and enters router configuration mode. |
| Step 3 | <code>Router(config)# summary-address address-mask</code> | (Optional) Specifies an address and IP subnet mask for redistributed routes so that only one summary route is advertised. |
| Step 4 | <code>area area-id virtual-link router-id [hello-interval seconds] [retransmit-interval seconds] [trans] [[authentication-key key] [message-digest-key key-id md5 key]]</code> | (Optional) Establishes a virtual link and set its parameters. See the “ OSPF Interface Parameters ” section on page 10-13 for parameter definitions and Table 10-2 on page 10-10 for virtual link defaults. |
| Step 5 | <code>Router(config)# default-information originate [always] [metric metric-value] [metric-type type-value] [route-map map-name]</code> | (Optional) Forces the ASBR to generate a default route into the OSPF routing domain. Parameters are all optional. |
| Step 6 | <code>Router(config)# ip ospf name-lookup</code> | (Optional) Configures DNS name lookup. The default is disabled. |
| Step 7 | <code>Router(config)# ip auto-cost reference-bandwidth ref-bw</code> | (Optional) Specifies an address range for which a single route will be advertised. Use this command only with area border routers. |
| Step 8 | <code>Router(config)# distance ospf {[inter-area dist1] [inter-area dist2] [external dist3]}</code> | (Optional) Changes the OSPF distance values. The default distance for each type of route is 110. The range is 1 to 255. |
| Step 9 | <code>Router(config)# passive-interface type number</code> | (Optional) Suppresses the sending of hello packets through the specified interface. |

| | Command | Purpose |
|---------|---|--|
| Step 10 | <code>Router(config)# timers spf <i>spf-delay</i> <i>spf-holdtime</i></code> | (Optional) Configures route calculation timers. <ul style="list-style-type: none"> <i>spf-delay</i>—Enter an integer from 0 to 65535. The default is 5 seconds; 0 means no delay. <i>spf-holdtime</i>—Enter an integer from 0 to 65535. The default is 10 seconds; 0 means no delay. |
| Step 11 | <code>Router(config)# ospf log-adj-changes</code> | (Optional) Sends syslog message when a neighbor state changes. |
| Step 12 | <code>Router(config)# end</code> | Returns to privileged EXEC mode. |
| Step 13 | <code>Router# show ip ospf [<i>process-id</i> <i>area-id</i>] database</code> | Displays lists of information related to the OSPF database for a specific router. For some of the keyword options, see to the “Monitoring OSPF” section on page 10-20. |
| Step 14 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

Change LSA Group Pacing

The OSPF LSA group pacing feature allows the router to group OSPF LSAs and pace the refreshing, check-summing, and aging functions for more efficient router use. This feature is enabled by default with a 4-minute default pacing interval, and you will not usually need to modify this parameter. The optimum group pacing interval is inversely proportional to the number of LSAs the router is refreshing, check-summing, and aging. For example, if you have approximately 10,000 LSAs in the database, decreasing the pacing interval would benefit you. If you have a very small database (40 to 100 LSAs), increasing the pacing interval to 10 to 20 minutes might benefit you slightly.

Beginning in privileged EXEC mode, follow this procedure to configure OSPF LSA pacing:

| | Command | Purpose |
|--------|---|---|
| Step 1 | <code>Router# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# router ospf <i>process-id</i></code> | Enables OSPF routing, and enters router configuration mode. |
| Step 3 | <code>Router(config)# timers lsa-group-pacing <i>seconds</i></code> | Changes the group pacing of LSAs. |
| Step 4 | <code>Router(config)# end</code> | Returns to privileged EXEC mode. |
| Step 5 | <code>Router# show running-config</code> | Verifies your entries. |
| Step 6 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

To return to the default value, use the **no timers lsa-group-pacing** router configuration command.

Loopback Interface

OSPF uses the highest IP address configured on the interfaces as its router ID. If this interface is down or removed, the OSPF process must recalculate a new router ID and resend all its routing information out of its interfaces. If a loopback interface is configured with an IP address, OSPF uses this IP address

as its router ID, even if other interfaces have higher IP addresses. Because loopback interfaces never fail, this provides greater stability. OSPF automatically prefers a loopback interface over other interfaces, and it chooses the highest IP address among all loopback interfaces.

Beginning in privileged EXEC mode, follow this procedure to configure a loopback interface:

| | Command | Purpose |
|--------|--|--|
| Step 1 | <code>Router# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# interface loopback interface-id</code> | Creates a loopback interface, and enters interface configuration mode. |
| Step 3 | <code>Router(config)# ip address address mask</code> | Assigns an IP address to this interface. |
| Step 4 | <code>Router(config)# end</code> | Returns to privileged EXEC mode. |
| Step 5 | <code>Router# show ip interface</code> | Verifies your entries. |
| Step 6 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

Use the `no interface loopback 0` global configuration command to disable the loopback interface.

Monitoring OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases.

Table 10-3 lists some of the privileged EXEC commands for displaying statistics. For more `show ip ospf database` privileged EXEC command options and for explanations of fields in the resulting display, refer to the *Cisco IOS IP and IP Routing Command Reference*.

Table 10-3 Show IP OSPF Statistics Commands

| Command | Purpose |
|---|--|
| <code>Router(config)# show ip ospf [process-id]</code> | Displays general information about OSPF routing processes. |
| <code>Router(config)# show ip ospf [process-id] database [router] [link-state-id]</code> <code>show ip ospf [process-id] database [router] [self-originate]</code> <code>show ip ospf [process-id] database [router] [adv-router ip-address]</code> <code>show ip ospf [process-id] database [network] [link-state-id]</code> <code>show ip ospf [process-id] database [summary] [link-state-id]</code> <code>show ip ospf [process-id] database [asbr-summary] [link-state-id]</code> <code>show ip ospf [process-id] database [external] [link-state-id]</code> <code>show ip ospf [process-id area-id] database [database-summary]</code> | Displays lists of information related to the OSPF database. |
| <code>Router(config)# show ip ospf border-routes</code> | Displays the internal OSPF routing ABR and ASBR table entries. |
| <code>Router(config)# show ip ospf interface [interface-name]</code> | Displays OSPF-related interface information. |
| <code>Router(config)# show ip ospf neighbor [interface-name] [neighbor-id] detail</code> | Displays OSPF interface neighbor information. |
| <code>Router(config)# show ip ospf virtual-links</code> | Displays OSPF-related virtual links information. |

Configuring EIGRP

Enhanced IGRP (EIGRP) is a Cisco proprietary enhanced version of the IGRP. Enhanced IGRP uses the same distance vector algorithm and distance information as IGRP; however, the convergence properties and the operating efficiency of Enhanced IGRP are significantly improved.

The convergence technology employs an algorithm referred to as the Diffusing Update Algorithm (DUAL), which guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize at the same time. Routers that are not affected by topology changes are not involved in recomputations.

IP EIGRP provides increased network width. With RIP, the largest possible width of your network is 15 hops. When IGRP is enabled, the largest possible width is 224 hops. Because the EIGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport-layer hop counter. EIGRP increments the transport control field only when an IP packet has traversed 15 routers and the next hop to the destination was learned through EIGRP. When a RIP route is used as the next hop to the destination, the transport control field is incremented as usual.

EIGRP offers these features:

- Fast convergence.
- Incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table, minimizing the bandwidth required for EIGRP packets.
- Less CPU usage than IGRP because full update packets need not be processed each time they are received.
- Protocol-independent neighbor discovery mechanism to learn about neighboring routers.
- Variable-length subnet masks (VLSMs).
- Arbitrary route summarization.
- EIGRP scales to large networks.

EIGRP has these four basic components:

- *Neighbor discovery and recovery* is the process that routers use to dynamically learn of other routers on their directly attached networks. Routers must also discover when their neighbors become unreachable or inoperative. Neighbor discovery and recovery is achieved with low overhead by periodically sending small hello packets. As long as hello packets are received, the Cisco IOS software can determine that a neighbor is alive and functioning. When this status is determined, the neighboring routers can exchange routing information.
- The *reliable transport protocol* is responsible for guaranteed, ordered delivery of EIGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some EIGRP packets must be sent reliably, and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities (such as Ethernet), it is not necessary to send hellos reliably to all neighbors individually. Therefore, EIGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets (such as updates) require acknowledgment, which is shown in the packet. The reliable transport has a provision to send multicast packets quickly when there are unacknowledged packets pending. Doing so helps ensure that convergence time remains low in the presence of varying speed links.
- The *DUAL finite state machine* embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses the distance information (known as a metric) to select efficient, loop-free paths. DUAL selects routes to be inserted into a routing table based on feasible successors. A successor is a neighboring router used for packet forwarding that has a least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no

feasible successors, but there are neighbors advertising the destination, a recomputation must occur. This is the process whereby a new successor is determined. The amount of time it takes to recompute the route affects the convergence time. Recomputation is processor-intensive; it is advantageous to avoid recomputation if it is not necessary. When a topology change occurs, DUAL tests for feasible successors. If there are feasible successors, it uses any it finds to avoid unnecessary recomputation.

- The *protocol-dependent modules* are responsible for network layer protocol-specific tasks. An example is the IP EIGRP module, which is responsible for sending and receiving EIGRP packets that are encapsulated in IP. It is also responsible for parsing EIGRP packets and informing DUAL of the new information received. EIGRP asks DUAL to make routing decisions, but the results are stored in the IP routing table. EIGRP is also responsible for redistributing routes learned by other IP routing protocols.

Table 10-4 shows the default EIGRP configuration.

Table 10-4 Default EIGRP Configuration

| Feature | Default Setting |
|-----------------------------|--|
| Auto summary | Enabled. Subprefixes are summarized to the classful network boundary when crossing classful network boundaries. |
| Default-information | Exterior routes are accepted and default information is passed between IGRP or EIGRP processes when doing redistribution. |
| Default metric | Only connected routes and interface static routes can be redistributed without a default metric. The metric includes: <ul style="list-style-type: none"> • Bandwidth: 0 or greater kbps. • Delay (tens of microseconds): 0 or any positive number that is a multiple of 39.1 nanoseconds. • Reliability: Any number between 0 and 255 (255 means 100 percent reliability). • Loading: Effective bandwidth as a number between 0 and 255 (255 is 100 percent loading). • MTU: Maximum transmission unit size of the route in bytes. 0 or any positive integer. |
| Distance | Internal distance: 90. External distance: 170. |
| EIGRP log-neighbor changes | Disabled. No adjacency changes logged. |
| IP authentication key-chain | No authentication provided. |
| IP authentication mode | No authentication provided. |
| IP bandwidth-percent | 50 percent. |
| IP hello interval | For low-speed nonbroadcast multiaccess (NBMA) networks: 60 seconds; all other networks: 5 seconds. |
| IP hold-time | For low-speed NBMA networks: 180 seconds; all other networks: 15 seconds. |
| IP split-horizon | Enabled. |
| IP summary address | No summary aggregate addresses are predefined. |


Table 10-4 Default EIGRP Configuration (continued)

| Feature | Default Setting |
|----------------|---|
| Metric weights | tos: 0 k1 and k3: 1 k2, k4, and k5: 0 |
| Network | None specified. |
| Offset-list | Disabled. |
| Router EIGRP | Disabled. |
| Set metric | No metric set in the route map. |
| Traffic-share | Distributed proportionately to the ratios of the metrics. |
| Variance | 1 (equal-cost load balancing). |

To create an EIGRP routing process, you must enable EIGRP and associate networks. EIGRP sends updates to the interfaces in the specified networks. If you do not specify an interface network, it is not advertised in any EIGRP update.

EIGRP Router Mode Commands

Beginning in privileged EXEC mode, follow these steps to configure EIGRP. Configuring the routing process is required; other steps are optional:

| | Command | Purpose |
|--------|--|--|
| Step 1 | <code>Router# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# router eigrp autonomous-system</code> | Enables an EIGRP routing process, and enters router configuration mode. The autonomous system number identifies the routes to other EIGRP routers and is used to tag routing information. |
| Step 3 | <code>Router(config)# network network-number</code> | Associates networks with an EIGRP routing process. EIGRP sends updates to the interfaces in the specified networks. If an interface's network is not specified, it is not advertised in any IGRP or EIGRP update. |
| Step 4 | <code>Router(config)# eigrp log-neighbor-changes</code> | (Optional) Enables logging of EIGRP neighbor changes to monitor routing system stability. |
| Step 5 | <code>Router(config)# metric weights tos k1 k2 k3 k4 k5</code> | (Optional) Adjusts the EIGRP metric. Although the defaults have been carefully determined to provide excellent operation in most networks, you can adjust them. |
| | |  Caution Determining metrics is complex and is not recommended without guidance from an experienced network designer. |

| | Command | Purpose |
|---------|---|--|
| Step 6 | <code>Router(config)# offset list</code> [[<i>access-list-number</i> <i>name</i>]] { <i>in</i> <i>out</i> } <i>offset</i> [<i>type-number</i>] | (Optional) Applies an offset list to routing metrics to increase incoming and outgoing metrics to routes learned through EIGRP. You can limit the offset list with an access list or an interface. |
| Step 7 | <code>Router(config)# no auto-summary</code> | (Optional) Disables automatic summarization of subnet routes into network-level routes. |
| Step 8 | <code>Router(config)# ip summary-address</code> <code>eigrp</code> <i>autonomous-system-number</i> <i>address-mask</i> | (Optional) Configures a summary aggregate. |
| Step 9 | <code>Router(config)# end</code> | Returns to privileged EXEC mode. |
| Step 10 | <code>Router# show ip protocols</code> | Verifies your entries. |
| Step 11 | <code>Router# copy running-config</code> <code>startup-config</code> | (Optional) Saves your entries in the configuration file. |

Use the **no** forms of these commands to disable the feature or return the setting to the default value.

Example 10-7 show ip protocols privileged EXEC command output (for EIGRP)


```
Router# show ip protocols
Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  Automatic network summarization is in effect
  Automatic address summarization:
    192.168.3.0/24 for POS0
    192.168.2.0/24 for FastEthernet0
  Maximum path: 4
  Routing for Networks:
    192.168.2.0
    192.168.3.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.1         90          00:03:16
  Distance: internal 90 external 170
```

EIGRP Interface Mode Commands

Other optional EIGRP parameters can be configured on an interface basis.

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|--------|--|--|
| Step 1 | <code>Router# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# interface</code> <i>interface-id</i> | Enters interface configuration mode, and specifies the Layer 3 interface to configure. |

| | Command | Purpose |
|---------|---|--|
| Step 3 | <code>Router(config)# ip bandwidth-percent eigrp percent</code> | (Optional) Configures the percentage of bandwidth that can be used by EIGRP on an interface. The default is 50 percent. |
| Step 4 | <code>Router(config)# ip summary-address eigrp autonomous-system-number address mask</code> | (Optional) Configures a summary aggregate address for a specified interface (not usually necessary if autosummary is enabled). |
| Step 5 | <code>Router(config)# ip hello-interval eigrp autonomous-system-number seconds</code> | (Optional) Changes the hello time interval for an EIGRP routing process. The range is 1 to 65535 seconds. The default is 60 seconds for low-speed NBMA networks and 5 seconds for all other networks. |
| Step 6 | <code>Router(config)# ip hold-time eigrp autonomous-system-number seconds</code> | (Optional) Changes the hold time interval for an EIGRP routing process. The range is 1 to 65535 seconds. The default is 180 seconds for low-speed NBMA networks and 15 seconds for all other networks. |
| | |  |
| | | Caution Do not adjust the hold time without consulting Cisco technical support. |
| Step 7 | <code>Router(config)# no ip split-horizon eigrp autonomous-system-number</code> | (Optional) Disables split horizon to allow route information to be advertised by a router out any interface from which that information originated. |
| Step 8 | <code>Router# end</code> | Returns to privileged EXEC mode. |
| Step 9 | <code>Router# show ip eigrp interface</code> | Displays the interfaces that EIGRP is active on and information about EIGRP relating to those interfaces. |
| Step 10 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

Use the **no** forms of these commands to disable the feature or return the setting to the default value.

Example 10-8 show ip eigrp interface privileged EXEC command output

```
Router# show ip eigrp interface
IP-EIGRP interfaces for process 1
```

| Interface | Peers | Xmit Queue Un/Reliable | Mean SRTT | Pacing Time Un/Reliable | Multicast Flow Timer | Pending Routes |
|-----------|-------|---------------------------|--------------|----------------------------|-------------------------|-------------------|
| PO0 | 1 | 0/0 | 20 | 0/10 | 50 | 0 |
| Fa0 | 0 | 0/0 | 0 | 0/10 | 0 | 0 |

Configure EIGRP Route Authentication

EIGRP route authentication provides MD5 authentication of routing updates from the EIGRP routing protocol to prevent the introduction of unauthorized or false routing messages from unapproved sources.

Beginning in privileged EXEC mode, follow these steps to enable authentication:

| | Command | Purpose |
|---------|--|---|
| Step 1 | <code>Router# configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>Router(config)# interface interface-id</code> | Enters interface configuration mode, and specifies the Layer 3 interface to configure. |
| Step 3 | <code>Router(config-if)# ip authentication mode eigrp autonomous-system md5</code> | Enables MD5 authentication in IP EIGRP packets. |
| Step 4 | <code>Router(config-if)# ip authentication key-chain eigrp autonomous-system key-chain</code> | Enables authentication of IP EIGRP packets. |
| Step 5 | <code>Router(config-if)# exit</code> | Returns to global configuration mode. |
| Step 6 | <code>Router(config)# key chain name-of-chain</code> | Identifies a key chain and enter key-chain configuration mode. Match the name configured in Step 4. |
| Step 7 | <code>Router(config)# key number</code> | In key-chain configuration mode, identifies the key number. |
| Step 8 | <code>Router(config)# key-string text</code> | In key-chain key configuration mode, identifies the key string. |
| Step 9 | <code>Router(config)# accept-lifetime start-time {infinite end-time duration seconds}</code> | (Optional) Specifies the time period during which the key can be received. The <i>start-time</i> and <i>end-time</i> syntax can be either <i>hh:mm:ss Month date year</i> or <i>hh:mm:ss date Month year</i> . The default is forever with the default <i>start-time</i> and the earliest acceptable date as January 1, 1993. The default <i>end-time</i> and duration is infinite . |
| Step 10 | <code>Router(config)# send-lifetime start-time {infinite end-time duration seconds}</code> | (Optional) Specifies the time period during which the key can be sent. The <i>start-time</i> and <i>end-time</i> syntax can be either <i>hh:mm:ss Month date year</i> or <i>hh:mm:ss date Month year</i> . The default is forever with the default <i>start-time</i> and the earliest acceptable date as January 1, 1993. The default <i>end-time</i> and duration is infinite . |
| Step 11 | <code>Router(config)# end</code> | Returns to privileged EXEC mode. |
| Step 12 | <code>Router# show key chain</code> | Displays authentication key information. |
| Step 13 | <code>Router# copy running-config startup-config</code> | (Optional) Saves your entries in the configuration file. |

Use the **no** forms of these commands to disable the feature or to return the setting to the default value.

Monitoring and Maintaining EIGRP

You can delete neighbors from the neighbor table. You can also display various EIGRP routing statistics. [Table 10-5](#) lists the privileged EXEC commands for deleting neighbors and displaying statistics. For explanations of fields in the resulting display, refer to the *Cisco IOS IP and IP Routing Command Reference* publication.

Table 10-5 IP EIGRP Clear and Show Commands

| Command | Purpose |
|--|--|
| Router# clear ip eigrp neighbors {ip-address interface} | Deletes neighbors from the neighbor table. |
| Router# show ip eigrp interface [interface] [as-number] | Displays information about interfaces configured for EIGRP. |
| Router# show ip eigrp neighbors [type-number] | Displays EIGRP discovered neighbors. |
| Router# show ip eigrp topology {autonomous-system-number} [[ip-address] mask] | Displays the EIGRP topology table for a given process. |
| Router# show ip eigrp traffic [autonomous-system-number] | Displays the number of packets sent and received for all or a specified EIGRP process. |

Example 10-9 show ip eigrp interface privileged EXEC command output

```
Router# show ip eigrp interface
IP-EIGRP interfaces for process 1

          Xmit Queue   Mean   Pacing Time   Multicast   Pending
Interface  Peers  Un/Reliable  SRTT  Un/Reliable  Flow Timer  Routes
PO0         1      0/0         20    0/10         50          0
Fa0         0      0/0         0     0/10         0           0
```

Example 10-10 show ip eigrp neighbors privileged EXEC command output

```
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                Interface   Hold Uptime   SRTT  RTO  Q  Seq Type
                               (sec)          (ms)  Cnt Num
0   192.168.2.1              PO0        13 00:08:15   20    200  0  2
```

Example 10-11 show ip eigrp topology privileged EXEC command output

```
Router# show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(192.168.3.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 192.168.1.0/24, 1 successors, FD is 30720
   via 192.168.2.1 (30720/28160), POS0
P 192.168.2.0/24, 1 successors, FD is 10752
   via Connected, POS0
P 192.168.3.0/24, 1 successors, FD is 28160
   via Connected, FastEthernet0
```

Example 10-12 show ip eigrp traffic privileged EXEC command output

```
Router# show ip eigrp traffic
IP-EIGRP Traffic Statistics for process 1
  Hellos sent/received: 273/136
  Updates sent/received: 5/2
  Queries sent/received: 0/0
```

```

Replies sent/received: 0/0
Acks sent/received: 1/2
Input queue high water mark 1, 0 drops
SIA-Queries sent/received: 0/0
SIA-Replies sent/received: 0/0

```

Border Gateway Protocol and Classless Interdomain Routing

Border Gateway Protocol (BGP) is an Exterior Gateway Protocol (EGP) that allows you to set up an interdomain routing system to automatically guarantee the loop-free exchange of routing information between autonomous systems. In BGP, each route consists of a network number, a list of autonomous systems that information has passed through (called the autonomous system path), and a list of other path attributes.

Layer 3 switching supports BGP version 4, including CIDR. CIDR lets you reduce the size of your routing tables by creating aggregate routes resulting in supernets. CIDR eliminates the concept of network classes within BGP and supports the advertising of IP prefixes. CIDR routes can be carried by OSPF, EIGRP, and RIP.

Configuring BGP

To configure BGP routing, perform the following steps, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router(config)# ip routing | Enables IP routing (default.) |
| Step 2 | Router(config)# router bgp <i>autonomous-system</i> | Defines BGP as the routing protocol and starts the BGP routing process. |
| Step 3 | Router(config-router)# network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>] | Flags a network as local to this autonomous system and enters it in the BGP table. |
| Step 4 | Router(config-router)# end | Returns to privileged EXEC mode. |

Example 10-13 Configuring BGP Routing

```

Router(config)# ip routing
Router(config)# router bgp 30
Router(config-router)# network 192.168.1.1
Router(config-router)# neighbor 192.168.2.1
Router(config-router)# end

```

For more information about configuring BGP routing, refer to the “Configuring BGP” chapter in the *Cisco IOS IP and IP Routing Configuration Guide*.

Verifying the BGP Configuration

Table 10-6 lists some common EXEC commands used to view the BGP configuration.

Table 10-6 BGP Show Commands

| Command | Purpose |
|--|--|
| Router# <code>show ip protocols [summary]</code> | Displays the protocol configuration. |
| Router# <code>show ip bgp neighbor</code> | Displays detailed information about the BGP and TCP connections to individual neighbors. |
| Router# <code>show ip bgp summary</code> | Displays the status of all BGP connections. |
| Router# <code>show ip bgp</code> | Displays the content of the BGP routing table. |

Example 10-14 BGP Configuration Information

```

Router# show ip protocols
Routing Protocol is "bgp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is enabled
  Automatic route summarization is enabled
  Redistributing: connected
  Neighbor(s):
    Address          FiltIn FiltOut DistIn DistOut Weight RouteMap
    192.168.2.1
  Maximum path: 1
  Routing for Networks:
  Routing Information Sources:
    Gateway          Distance    Last Update
  Distance: external 20 internal 200 local 200

Router# show ip bgp neighbor
BGP neighbor is 192.168.2.1, remote AS 1, internal link
  BGP version 4, remote router ID 192.168.2.1
  BGP state = Established, up for 00:08:46
  Last read 00:00:45, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
  Received 13 messages, 0 notifications, 0 in queue
  Sent 13 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Default minimum time between advertisement runs is 5 seconds

For address family: IPv4 Unicast
  BGP table version 3, neighbor version 3
  Index 1, Offset 0, Mask 0x2
  2 accepted prefixes consume 72 bytes
  Prefix advertised 2, suppressed 0, withdrawn 0
  Number of NLRI in the update sent: max 2, min 0

  Connections established 1; dropped 0
  Last reset never
  Connection state is ESTAB, I/O status: 1, unread input bytes: 0
  Local host: 192.168.2.2, Local port: 179
  Foreign host: 192.168.2.1, Foreign port: 11001

  Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x45B7B4):
Timer          Starts    Wakeups    Next
Retrans        13         0          0x0
TimeWait       0          0          0x0

```

```

AckHold          13          9          0x0
SendWnd          0          0          0x0
KeepAlive        0          0          0x0
GiveUp           0          0          0x0
PmtuAger         0          0          0x0
DeadWait         0          0          0x0

```

```

iss: 3654396253  snduna: 3654396567  sndnxt: 3654396567    sndwnd: 16071
irs: 3037331955  rcvnxt: 3037332269  rcvwnd:    16071  delrcvwnd: 313

```

```

SRTT: 247 ms, RTTO: 663 ms, RTV: 416 ms, KRTT: 0 ms
minRTT: 4 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs

```

```

Datagrams (max data segment is 1460 bytes):
Rcvd: 15 (out of order: 0), with data: 13, total data bytes: 313
Sent: 22 (retransmit: 0), with data: 12, total data bytes: 313

```

```
Router# show ip bgp summary
```

```

BGP router identifier 192.168.3.1, local AS number 1
BGP table version is 3, main routing table version 3
3 network entries and 4 paths using 435 bytes of memory
2 BGP path attribute entries using 120 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 3/6 prefixes, 4/0 paths, scan interval 60 secs

```

```

Neighbor      V    AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.2.1   4    1     14     14      3     0   0 00:09:45      2

```

```
Router# show ip bgp
```

```

BGP table version is 3, local router ID is 192.168.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

```

```

      Network          Next Hop           Metric LocPrf Weight Path
* i192.168.1.0        192.168.2.1         0      100      0 ?
* i192.168.2.0        192.168.2.1         0      100      0 ?
*>                   0.0.0.0             0                   32768 ?
*> 192.168.3.0        0.0.0.0             0                   32768 ?

```

Configuring IS-IS

To configure IS-IS routing, perform the following steps, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router(config)# router isis [<i>tag</i>] | Defines IS-IS as the IP routing protocol. |
| Step 2 | Router(config-router)# net <i>network-entity-title</i> | Configures network entity titles (NETs) for the routing process; you can specify a name for a NET as well as an address. |
| Step 3 | Router(config-router)# interface <i>interface-type interface-id</i> | Enters interface configuration mode. |
| Step 4 | Router(config-if)# ip address <i>ip-address</i> <i>mask</i> | Assigns an IP address to the interface. |
| Step 5 | Router(config-if)# ip router isis [<i>tag</i>] | Specifies that this interface should run IS-IS. |
| Step 6 | Router(config-if)# end | Returns to privileged EXEC mode. |

Example 10-15 Configuring IS-IS Routing

```

Router(config)# router isis
Router(config-router)# net 49.0001.0000.0000.000a.00
Router(config-router)# interface gigabitethernet 0
Router(config-if)# ip router isis
Router(config-if)# end

```

For more information about configuring IS-IS routing, refer to the “Configuring Integrated IS-IS” chapter in the *Cisco IOS IP and IP Routing Configuration Guide*.

Verifying the IS-IS Configuration

To verify the IS-IS configuration, use the EXEC commands listed in [Table 10-7](#).

Table 10-7 IS-IS Show Commands

| Command | Purpose |
|-------------------------------------|---|
| Router# show ip protocols [summary] | Displays the protocol configuration. |
| Router# show isis database | Displays the IS-IS link-state database. |
| Router# show clns neighbor | Displays the ES and IS neighbors. |



Note

The ML Series does not support Connectionless Network Service Protocol (CLNS) routing.

Example 10-16 IS-IS Configuration

```

Router# show ip protocols
Routing Protocol is "isis"
  Invalid after 0 seconds, hold down 0, flushed after 0
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: isis
  Address Summarization:
    None
  Maximum path: 4
  Routing for Networks:
    FastEthernet0
    POS0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.1     115          00:06:48
  Distance: (default is 115)

Router# show isis database

IS-IS Level-1 Link State Database:
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
Router_A.00-00  0x00000003   0xA72F        581            0/0/0
Router_A.02-00  0x00000001   0xA293        581            0/0/0
Router.00-00    * 0x00000004  0x79F9        582            0/0/0
IS-IS Level-2 Link State Database:
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
Router_A.00-00  0x00000004   0xF0D6        589            0/0/0
Router_A.02-00  0x00000001   0x328C        581            0/0/0

```

```

Router.00-00          * 0x00000004  0x6A09          586          0/0/0

Router# show clns neighbors

System Id      Interface  SNPA              State Holdtime  Type Protocol
Router_A      PO0       0005.9a39.6790   Up       7          L1L2 IS-IS

```

Configuring Static Routes

Static routes are user-defined routes that cause packets moving between a source and a destination to take a specified path. Static routes can be important if the router cannot build a route to a particular destination. They are also useful for specifying a gateway of last resort to which all unroutable packets are sent.

Beginning in privileged EXEC mode, follow these steps to configure a static route:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router# configure terminal | Enters global configuration mode. |
| Step 2 | Router(config)# ip route <i>prefix mask { address interface } [distance]</i> | Establishes a static route. |
| Step 3 | Router(config)# end | Returns to privileged EXEC mode. |
| Step 4 | Router# copy running-config startup-config | (Optional) Saves your entries in the configuration file. |

Example 10-17 Static Route

```
Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1
```

Use the **no ip route** *prefix mask {address | interface}* global configuration command to remove a static route.

Example 10-18 show ip route privileged EXEC command output (with a static route configured)

```

Router# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is 192.168.2.1 to network 0.0.0.0

C    192.168.2.0/24 is directly connected, POS0
C    192.168.3.0/24 is directly connected, FastEthernet0
S*   0.0.0.0/0 [1/0] via 192.168.2.1

```

The output from the **show ip route** privileged EXEC command lists codes for the routing protocols. [Table 10-8](#) shows the default administrative distances for these routing protocols.

Table 10-8 Routing Protocol Default Administrative Distances

| Route Source | Default Distance |
|---------------------|------------------|
| Connected interface | 0 |
| Static route | 1 |
| EIRGP summary route | 5 |
| External BGP | 20 |
| Internal EIGRP | 90 |
| OSPF | 110 |
| RIP | 120 |
| External EIGRP | 170 |
| Internal BGP | 200 |
| Unknown | 225 |

Monitoring Static Routes

You can display statistics about static routes with the **show ip route** command. For more **show ip** privileged EXEC command options and for explanations of fields in the resulting display, refer to the *Cisco IOS IP and IP Routing Command Reference* publication.

| Command | Purpose |
|-----------------------|--|
| Router# show ip route | Displays detailed information about the routing table. |

Example 10-19 show ip route output (with a static route configured)

```
Router# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 192.168.2.1 to network 0.0.0.0

C    192.168.2.0/24 is directly connected, POS0
C    192.168.3.0/24 is directly connected, FastEthernet0
S*   0.0.0.0/0 [1/0] via 192.168.2.1
```

Monitoring and Maintaining the IP Network

You can remove all contents of a particular cache, table, or database. You can also display specific statistics. Use the privileged EXEC commands in [Table 10-9](#) to clear routes or display status.

Table 10-9 Commands to Clear IP Routes or Display Route Status

| Command | Purpose |
|---|---|
| Router# <code>clear ip route</code> { <i>network</i> [<i>mask</i> *]} | Clears one or more routes from the IP routing table. |
| Router# <code>show ip protocols</code> | Displays the parameters and state of the active routing protocol process. |
| Router# <code>show ip route</code> [{ <i>address</i> [<i>mask</i>] [<i>longer-prefixes</i>] [<i>protocol</i> [<i>process-id</i>]}] | Displays the current state of the routing table. |
| Router# <code>show ip interface</code> <i>interface</i> | Displays detailed information about the interface. |
| Router# <code>show ip interface brief</code> | Displays summary status information about all interfaces. |
| Router# <code>show ip route summary</code> | Displays the current state of the routing table in summary form. |
| Router# <code>show ip route supernets-only</code> | Displays supernets. |
| Router# <code>show ip cache</code> | Displays the routing table used to switch IP traffic. |
| Router# <code>show route-map</code> [<i>map-name</i>] | Displays all route maps configured or only the one specified. |

Understanding IP Multicast Routing

As networks increase in size, multicast routing becomes critically important as a means to determine which segments require multicast traffic and which do not. IP multicasting allows IP traffic to be propagated from one source to a number of destinations, or from many sources to many destinations. Rather than sending one packet to each destination, one packet is sent to the multicast group identified by a single IP destination group address.

A principal component of IP multicasting is the Internet Group Management Protocol (IGMP). Hosts identify their multicast group membership by sending IGMP messages to the ML-Series card. Traffic is sent to all members of a multicast group. A host can be a member of more than one group at a time. In addition, a host does not need to be a member of a group to send data to that group. When you enable Protocol Independent Multicast (PIM) on an interface, you will have enabled IGMP operation on that same interface.

The ML-Series cards support the protocol independent multicast (PIM) routing protocol and the Auto-RP configuration.

PIM includes three different modes of behavior for dense and sparse traffic environments. These are referred to as *dense mode*, *sparse mode*, and *sparse-dense mode*.

PIM dense mode assumes that the downstream networks want to receive the datagrams forwarded to them. The ML-Series card forwards all packets on all outgoing interfaces until pruning and truncating occur. Interfaces that have PIM dense mode enabled receive the multicast data stream until it times out. PIM dense mode is most useful under these conditions:

- When senders and receivers are in close proximity to each other
- When the internetwork has fewer senders than receivers
- When the stream of multicast traffic is constant

PIM sparse mode assumes that the downstream networks do not want to forward multicast packets for a group unless there is an explicit request for the traffic. PIM sparse mode defines a rendezvous point, which is used as a registration point to facilitate the proper routing of packets.

When a sender wants to send data, it first sends the data to the rendezvous point. When a ML-Series card is ready to receive data, it registers with the rendezvous point. After the data stream begins to flow from the sender to the rendezvous point and then to the receiver, ML-Series cards in the data path optimize the path by automatically removing any unnecessary hops, including the rendezvous point.

PIM sparse mode is optimized for environments in which there are many multipoint data streams and each multicast stream goes to a relatively small number of LANs in the internetwork. PIM sparse mode is most useful under these conditions:

- When there are few receivers in the group
- When senders and receivers are separated by WAN links
- When the stream of multicast traffic is intermittent


Note

The ML-Series card support Reverse Path Forwarding (RPF) multicast, but not RPF unicast.

Configuring IP Multicast Routing

To configure IP multicast routing, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|---|
| Step 1 | Router(config)# ip multicast-routing | Enables IP multicasting on the ML-Series card. |
| Step 2 | Router(config)# interface <i>type number</i> | Enters interface configuration mode to configure any interface. |
| Step 3 | Router(config-if)# ip pim {dense-mode sparse mode sparse-dense-mode} | Runs IP multicast routing on each interface on which you enter this command. You must indicate dense mode, sparse mode, or sparse-dense mode. |
| Step 4 | Router(config)# ip pim rp-address <i>rendezvous-point ip address</i> | Configures a rendezvous point for the multicast group. |
| Step 5 | Router(config-if)# end | Returns to privileged EXEC mode. |
| Step 6 | Router# copy running-config startup-config | (Optional) Saves your configuration changes to NVRAM. |

Monitoring and Verifying IP Multicast Operation

After IP multicast routing is configured, you can monitor and verify its operation by performing the commands listed in [Table 10-10](#), from privileged EXEC mode.

Table 10-10 IP Multicast Routing Show Commands

| Command | Purpose |
|---|--|
| <code>Router#show ip mroute</code> | Shows the complete multicast routing table and the combined statistics of packets processed. |
| <code>Router#show ip pim neighbor</code> | When used in EXEC mode, lists the PIM neighbors discovered by the Cisco IOS software. |
| <code>Router#show ip pim interface</code> | Displays information about interfaces configured for PIM. |
| <code>Router#show ip pim rp</code> | When used in EXEC mode, displays the active rendezvous points (RPs) that are cached with associated multicast routing entries. |



Configuring IRB

This chapter describes how to configure integrated routing and bridging (IRB) for the ML-Series card. For more information about the Cisco IOS commands used in this chapter, refer to the *Cisco IOS Command Reference* publication.

This chapter includes the following major sections:

- [Understanding Integrated Routing and Bridging, page 11-1](#)
- [Configuring IRB, page 11-2](#)
- [Monitoring and Verifying IRB, page 11-4](#)



Caution

Cisco Inter-Switch Link (ISL) and Cisco Dynamic Trunking Protocol (DTP) are not supported by the ML-Series, but the ML-Series broadcast forwards these formats. Using ISL or DTP on connecting devices is not recommended. Some Cisco devices attempt to use ISL or DTP by default.

Understanding Integrated Routing and Bridging

Your network might require you to bridge local traffic within several segments and have hosts on the bridged segments reach the hosts or ML-Series card on routed networks. For example, if you are migrating bridged topologies into routed topologies, you might want to start by connecting some of the bridged segments to the routed networks.

Using the integrated routing and bridging (IRB) feature, you can route a given protocol between routed interfaces and bridge groups within a single ML-Series card. Specifically, local or unroutable traffic is bridged among the bridged interfaces in the same bridge group, while routable traffic is routed to other routed interfaces or bridge groups.

Because bridging is in the data link layer and routing is in the network layer, they have different protocol configuration models. With IP, for example, bridge group interfaces belong to the same network and have a collective IP network address. In contrast, each routed interface represents a distinct network and has its own IP network address. Integrated routing and bridging uses the concept of a Bridge Group Virtual Interface (BVI) to enable these interfaces to exchange packets for a given protocol.

A BVI is a virtual interface within the ML-Series card that acts like a normal *routed* interface. A BVI does not support bridging but actually represents the corresponding bridge group to routed interfaces within the ML-Series card. The interface number is the link between the BVI and the bridge group.

Before configuring IRB, consider the following:

- The default routing/bridging behavior in a bridge group (when IRB is enabled) is to bridge all packets. Make sure that you explicitly configure routing on the BVI for IP traffic.

- Packets of unroutable protocols such as local-area transport (LAT) are always bridged. You cannot disable bridging for the unroutable traffic.
- Protocol attributes should not be configured on the bridged interfaces when you are using IRB to bridge and route a given protocol. You can configure protocol attributes on the BVI, but you cannot configure bridging attributes on the BVI.
- A bridge links several network segments into one large, flat network. To bridge a packet coming from a routed interface among bridged interfaces, the bridge group should be represented by one interface.
- All ports in a BVI group must have matching MTU settings.

Configuring IRB

The process of configuring integrated routing and bridging consists of the following tasks:

1. Configure bridge groups and routed interfaces.
 - a. Enable bridging.
 - b. Assign interfaces to the bridge groups.
 - c. Configure the routing.
2. Enable IRB.
3. Configure the BVI.
 - a. Enable the BVI to accept routed packets.
 - b. Enable routing on the BVI.
4. Configure IP addresses on the routed interfaces.
5. Verify the IRB configuration.

When you configure the BVI and enable routing on it, packets that come in on a routed interface destined for a host on a segment that is in a bridge group are routed to the BVI and forwarded to the bridging engine. From the bridging engine, the packet exits through a bridged interface. Similarly, packets that come in on a bridged interface but are destined for a host on a routed interface go first to the BVI. The BVI forwards the packets to the routing engine that sends them out on the routed interface.

To configure a bridge group and an interface in the bridge group, perform the following procedure, beginning in global configuration mode:

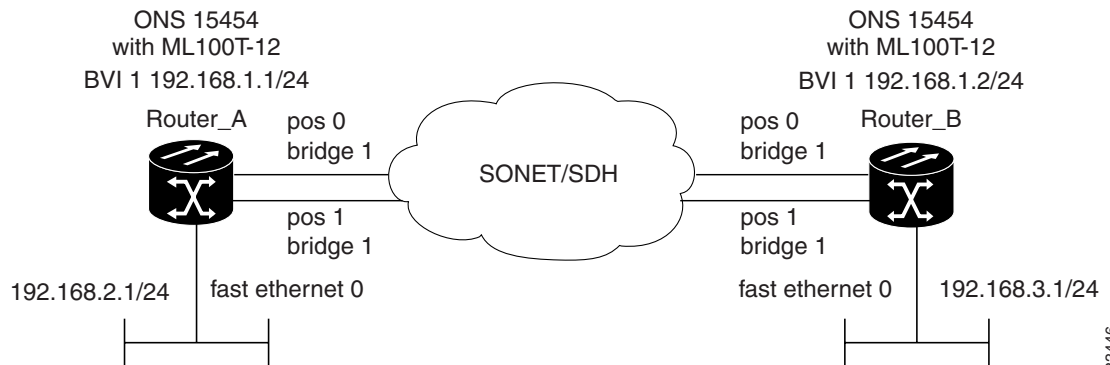
| | Command | Purpose |
|--------|--|--|
| Step 1 | Router(config)# bridge <i>bridge-group</i> protocol {ieee rstp} | Defines one or more bridge groups. |
| Step 2 | Router(config)# interface <i>type number</i> | Enters interface configuration mode. |
| Step 3 | Router(config-if)# bridge-group <i>bridge-group</i> | Assigns the interface to the specified bridge group. |
| Step 4 | Router(config-if)# end | Returns to privileged EXEC mode. |

To enable and configure IRB and BVI, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|---|
| Step 1 | Router(config)# bridge irb | Enables IRB. Allows bridging of traffic. |
| Step 2 | Router(config)# interface bvi <i>bridge-group</i> | Configures the BVI by assigning the number of the corresponding bridge group to the BVI. Each bridge group can have only one corresponding BVI. |
| Step 3 | Router(config-if)# ip address <i>ip-address</i> <i>ip-address-subnet-mask</i> | Configures IP addresses on routed interfaces. |
| Step 4 | Router(config-if)# exit | Exits the interface configuration mode. |
| Step 5 | Router(config)# bridge <i>bridge-group</i> route <i>protocol</i> | Enables a BVI to accept and route routable packets received from its corresponding bridge group. Enter this command for each protocol that you want the BVI to route from its corresponding bridge group to other routed interfaces. |
| Step 6 | Router(config)# end | Returns to the privileged EXEC mode. |
| Step 7 | Router# copy running-config startup-config | (Optional) Saves your configuration changes to NVRAM. |

Configuring IRB

Example 11-1 Configuring IRB



83446

Example 11-2 Configuring Router A

```
bridge irb
bridge 1 protocol ieee
bridge 1 route ip
!
!
interface FastEthernet0
ip address 192.168.2.1 255.255.255.0
!
interface POS0
no ip address
```

```

    crc 32
  bridge-group 1
    pos flag c2 1
  !
  interface POS1
    no ip address
    crc 32
  bridge-group 1
    pos flag c2 1
  !
  interface BVI1
    ip address 192.168.1.1 255.255.255.0
  !
  router ospf 1
    log-adjacency-changes
    network 192.168.1.0 0.0.0.255 area 0
    network 192.168.2.0 0.0.0.255 area 0

```

Example 11-3 Configuring Router B

```

bridge irb
bridge 1 protocol ieee
bridge 1 route ip
!
!
interface FastEthernet0
ip address 192.168.3.1 255.255.255.0
!
interface POS0
no ip address
crc 32
bridge-group 1
pos flag c2 1
!
interface POS1
no ip address
crc 32
bridge-group 1
pos flag c2 1
!
interface BVI1
ip address 192.168.1.2 255.255.255.0
!
router ospf 1
log-adjacency-changes
network 192.168.1.0 0.0.0.255 area 0
network 192.168.3.0 0.0.0.255 area 0

```

Monitoring and Verifying IRB

Table 11-1 shows the privileged EXEC commands for monitoring and verifying IRB.

Table 11-1 Commands for Monitoring and Verifying IRB

| Command | Purpose |
|--|---|
| Router# show interfaces bvi <i>bridge-group</i> | Shows BVI information, such as the BVI MAC address and processing statistics. |
| Router# show interfaces [<i>type-number</i>] irb | Shows BVI information for the following: <ul style="list-style-type: none"> • Protocols that this bridged interface can route to the other routed interface (if this packet is routable). • Protocols that this bridged interface bridges |

The following is sample output from the **show interfaces bvi** and **show interfaces irb** commands:

Example 11-4 Monitoring and Verifying IRB

```

Router# show interfaces bvi1
BVI1 is up, line protocol is up
  Hardware is BVI, address is 0011.2130.b340 (bia 0000.0000.0000)
  Internet address is 100.100.100.1/24
  MTU 1500 bytes, BW 145152 Kbit, DLY 5000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 03:35:28, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts (0 IP multicast)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    1353 packets output, 127539 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out

Router# show interfaces irb
BVI1
Software MAC address filter on BVI1
  Hash Len  Address  Matches  Act  Type
  0x00:  0 ffff.ffff.ffff      0 RCV Physical broadcast
GigabitEthernet0
Bridged protocols on GigabitEthernet0:
  clns      ip
Software MAC address filter on GigabitEthernet0
  Hash Len  Address  Matches  Act  Type
  0x00:  0 ffff.ffff.ffff      0 RCV Physical broadcast
  0x58:  0 0100.5e00.0006      0 RCV IP multicast
  0x5B:  0 0100.5e00.0005      0 RCV IP multicast
  0x65:  0 0011.2130.b344      0 RCV Interface MAC address
  0xC0:  0 0100.0ccc.cccc      0 RCV CDP
  0xC2:  0 0180.c200.0000      0 RCV IEEE spanning tree
POS0
Routed protocols on POS0:
  ip

```

```

Bridged protocols on POS0:
  clns      ip
Software MAC address filter on POS0
  Hash Len   Address      Matches  Act    Type
0x00:  0  ffff.ffff.ffff      9 RCV Physical broadcast
0x58:  0  0100.5e00.0006      0 RCV IP multicast
0x5B:  0  0100.5e00.0005     1313 RCV IP multicast
0x61:  0  0011.2130.b340      38 RCV Interface MAC address
0x61:  1  0011.2130.b340      0 RCV Bridge-group Virtual Interface
0x65:  0  0011.2130.b344      0 RCV Interface MAC address
0xC0:  0  0100.0ccc.cccc     224 RCV CDP
0xC2:  0  0180.c200.0000      0 RCV IEEE spanning tree
POS1
SPR1
Bridged protocols on SPR1:
  clns      ip
Software MAC address filter on SPR1
  Hash Len   Address      Matches  Act    Type
0x00:  0  ffff.ffff.ffff      0 RCV Physical broadcast
0x60:  0  0011.2130.b341      0 RCV Interface MAC address
0x65:  0  0011.2130.b344      0 RCV Interface MAC address
0xC0:  0  0100.0ccc.cccc      0 RCV CDP
0xC2:  0  0180.c200.0000      0 RCV IEEE spanning tree

```

Table 11-1 describes significant fields shown in the display.

Table 11-2 show interfaces irb Field Descriptions

| Field | Description |
|-----------------------------------|---|
| Routed protocols on... | List of the routed protocols configured for the specified interface. |
| Bridged protocols on... | List of the bridged protocols configured for the specified interface. |
| Software MAC address filter on... | Table of software MAC address filter information for the specified interface. |
| Hash | Hash key/relative position in the keyed list for this MAC-address entry. |
| Len | Length of this entry to the beginning element of this hash chain. |
| Address | Canonical (Ethernet ordered) MAC address. |
| Matches | Number of received packets matched to this MAC address. |
| Routed protocols on... | List of the routed protocols configured for the specified interface. |
| Bridged protocols on... | List of the bridged protocols configured for the specified interface. |



Configuring VRF Lite

This chapter describes how to configure VPN Routing and Forwarding Lite (VRF Lite) for the ML-Series cards. For additional information about the Cisco IOS commands used in this chapter, refer to the *Cisco IOS Command Reference* publication. This chapter contains the following major sections:

- [Understanding VRF Lite, page 12-1](#)
- [Configuring VRF Lite, page 12-2](#)
- [VRF Lite Configuration Example, page 12-2](#)
- [Monitoring and Verifying VRF Lite, page 12-7](#)



Note

If you have already configured bridging, you may now proceed with configuring VRF Lite as an optional step.

Understanding VRF Lite

VRF is an extension of IP routing that provides multiple routing instances. It provides a separate IP routing and forwarding table to each VPN and is used in concert with MP-iBGP (Multi-Protocol internal BGP) between provider equipment (PE) routers to provide Layer 3 MPLS-VPN. However, ML-Series VRF implementation is without MP-iBGP. With VRF Lite, the ML Series is considered a PE-extension or a customer equipment (CE)-extension. VRF Lite is considered a PE-extension since it has VRF (but without MP-iBGP), and it is considered a CE-extension since this CE can have multiple VRFs and serves many customer with one CE box.

Under VRF Lite, an ML-Series CE can have multiple interfaces/subinterfaces with PE for different customers (while a normal CE is only for one customer). It holds VRFs (routing information) locally and it does not distribute the VRFs to its connected PE(s). It uses VRF information to direct traffic to the correct interfaces/subinterfaces when it receives traffic from customers' routers or from Internet service provider (ISP) PE router(s).

Configuring VRF Lite

Perform the following procedure to configure VRF Lite:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router(config)# ip vrf <i>vrf-name</i> | Enters VRF configuration mode and assigns a VRF name. |
| Step 2 | Router(config-vrf)# rd <i>route-distinguisher</i> | Creates a VPN route distinguisher. |
| Step 3 | Router(config-vrf)# route-target { import export both } <i>route-distinguisher</i> | Creates a list of import and/or export route target communities for the specified VRF. |
| Step 4 | Router(config-vrf)# import map <i>route-map</i> | (Optional) Associates the specified route map with the VRF. |
| Step 5 | Router(config-vrf)# exit | Exits the current configuration mode and enters global configuration mode. |
| Step 6 | Router(config)# interface type number | Specifies an interface and enters interface configuration mode. |
| Step 7 | Router(config-vrf)# ip vrf forwarding <i>vrf-name</i> | Associates a VRF with an interface or subinterface. |
| Step 8 | Router(config-if)# end | Exits to privileged EXEC mode. |
| Step 9 | Router# copy running-config startup-config | (Optional) Saves configuration changes to NVRAM. |

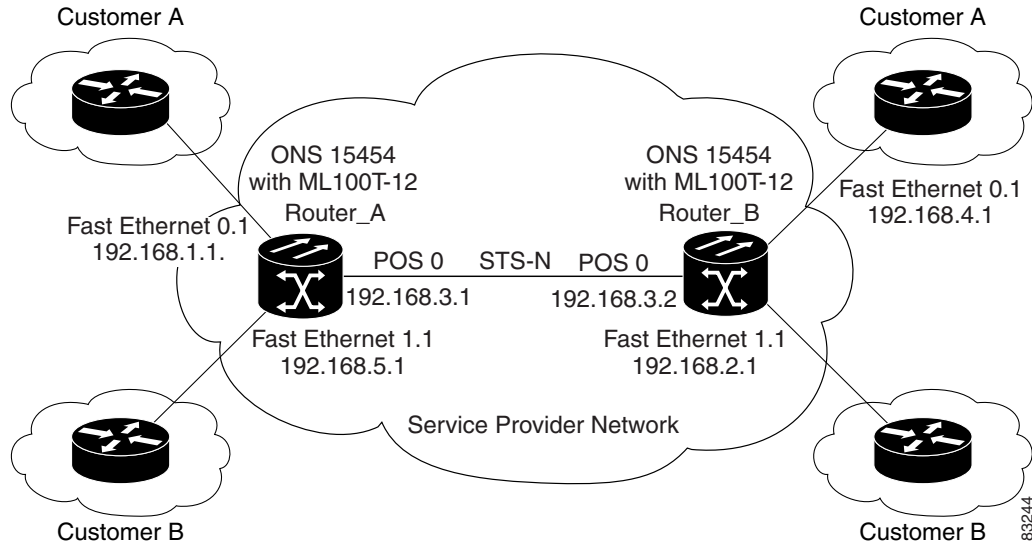
Example 12-1 Configuring a VRF (*customer_a*) with a route-distinguisher of 1:1 to Fast Ethernet Interface 0.1

```
Router(config)# ip vrf customer_a
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target both 1:1
Router(config)# interface fastEthernet 0.1
Router(config-subif)# ip vrf forwarding customer_a
```

VRF Lite Configuration Example

Figure 12-1 shows an example of a VRF Lite configuration. The configurations for Router A and Router B are provided in Example 12-2 and Example 12-3 on page 12-4, respectively. The associated routing tables are shown in Example 12-4 on page 12-5 through Example 12-9 on page 12-7.

Figure 12-1 VRF Lite—Sample Network Scenario

**Example 12-2 Router A Configuration**

```

hostname Router_A
!
ip vrf customer_a
  rd 1:1
  route-target export 1:1
  route-target import 1:1
!
ip vrf customer_b
  rd 2:2
  route-target export 2:2
  route-target import 2:2
!
bridge 1 protocol ieee
bridge 2 protocol ieee
bridge 3 protocol ieee
!
!
interface FastEthernet0
  no ip address
!
interface FastEthernet0.1
  encapsulation dot1Q 2
  ip vrf forwarding customer_a
  ip address 192.168.1.1 255.255.255.0
  bridge-group 2
!
interface FastEthernet1
  no ip address
!
interface FastEthernet1.1
  encapsulation dot1Q 3
  ip vrf forwarding customer_b
  ip address 192.168.2.1 255.255.255.0
  bridge-group 3
!
interface POS0

```

```

no ip address
crc 32
no cdp enable
pos flag c2 1
!
interface POS0.1
encapsulation dot1Q 1 native
ip address 192.168.50.1 255.255.255.0
bridge-group 1
!
interface POS0.2
encapsulation dot1Q 2
ip vrf forwarding customer_a
ip address 192.168.100.1 255.255.255.0
bridge-group 2
!
interface POS0.3
encapsulation dot1Q 3
ip vrf forwarding customer_b
ip address 192.168.200.1 255.255.255.0
bridge-group 3
!
router ospf 1
log-adjacency-changes
network 192.168.50.0 0.0.0.255 area 0
!
router ospf 2 vrf customer_a
log-adjacency-changes
network 192.168.1.0 0.0.0.255 area 0
network 192.168.100.0 0.0.0.255 area 0
!
router ospf 3 vrf customer_b
log-adjacency-changes
network 192.168.2.0 0.0.0.255 area 0
network 192.168.200.0 0.0.0.255 area 0
!

```

Example 12-3 Router_B Configuration

```

hostname Router_B
!
ip vrf customer_a
rd 1:1
route-target export 1:1
route-target import 1:1
!
ip vrf customer_b
rd 2:2
route-target export 2:2
route-target import 2:2
!
bridge 1 protocol ieee
bridge 2 protocol ieee
bridge 3 protocol ieee
!
!
interface FastEthernet0
no ip address
!
interface FastEthernet0.1
encapsulation dot1Q 2
ip vrf forwarding customer_a
ip address 192.168.4.1 255.255.255.0

```

```

    bridge-group 2
    !
interface FastEthernet1
    no ip address
    !
interface FastEthernet1.1
    encapsulation dot1Q 3
    ip vrf forwarding customer_b
    ip address 192.168.5.1 255.255.255.0
    bridge-group 3
    !
interface POS0
    no ip address
    crc 32
    no cdp enable
    pos flag c2 1
    !
interface POS0.1
    encapsulation dot1Q 1 native
    ip address 192.168.50.2 255.255.255.0
    bridge-group 1
    !
interface POS0.2
    encapsulation dot1Q 2
    ip vrf forwarding customer_a
    ip address 192.168.100.2 255.255.255.0
    bridge-group 2
    !
interface POS0.3
    encapsulation dot1Q 3
    ip vrf forwarding customer_b
    ip address 192.168.200.2 255.255.255.0
    bridge-group 3
    !
router ospf 1
    log-adjacency-changes
    network 192.168.50.0 0.0.0.255 area 0
    !
router ospf 2 vrf customer_a
    log-adjacency-changes
    network 192.168.4.0 0.0.0.255 area 0
    network 192.168.100.0 0.0.0.255 area 0
    !
router ospf 3 vrf customer_b
    log-adjacency-changes
    network 192.168.5.0 0.0.0.255 area 0
    network 192.168.200.0 0.0.0.255 area 0
    !

```

Example 12-4 Router_A Global Routing Table

```

Router_A# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.50.0/24 is directly connected, POS0.1

```

Example 12-5 Router_A customer_a VRF Routing Table

```
Router_A# show ip route vrf customer_a
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

O    192.168.4.0/24 [110/2] via 192.168.100.2, 00:15:35, POS0.2
C    192.168.1.0/24 is directly connected, FastEthernet0.1
C    192.168.100.0/24 is directly connected, POS0.2
```

Example 12-6 Router_A customer_b VRF Routing Table

```
Router_A# show ip route vrf customer_b
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.200.0/24 is directly connected, POS0.3
O    192.168.5.0/24 [110/2] via 192.168.200.2, 00:10:32, POS0.3
C    192.168.2.0/24 is directly connected, FastEthernet1.1
```

Example 12-7 Router_B Global Routing Table

```
Router_B# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.50.0/24 is directly connected, POS0.1
```

Example 12-8 Router_B customer_a VRF Routing Table

```
Router_B# sh ip route vrf customer_a
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set
```

```

C    192.168.4.0/24 is directly connected, FastEthernet0.1
O    192.168.1.0/24 [110/2] via 192.168.100.1, 00:56:24, POS0.2
C    192.168.100.0/24 is directly connected, POS0.2

```

Example 12-9 Router_B customer_b VRF Routing Table

```

Router_B# show ip route vrf customer_b
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.200.0/24 is directly connected, POS0.3
C    192.168.5.0/24 is directly connected, FastEthernet1.1
O    192.168.2.0/24 [110/2] via 192.168.200.1, 00:10:51, POS0.3

```

Monitoring and Verifying VRF Lite

Table 12-1 shows the privileged EXEC commands for monitoring and verifying VRF Lite.

Table 12-1 Commands for Monitoring and Verifying VRF Lite

| Command | Purpose |
|---|---|
| Router# show ip vrf | Displays the set of VRFs and interfaces |
| Router# show ip route vrf vrf-name | Displays the IP routing table for a VRF |
| Router# show ip protocols vrf vrf-name | Displays the routing protocol information for a VRF |
| Router# ping vrf vrf-name ip-address | Pings an IP address that has a specific VRF |



Configuring Quality of Service

This chapter describes the QoS features built into your ML-Series card and how to map QoS scheduling at both the system and interface levels.

This chapter contains the following major sections:

- [Understanding QoS, page 13-1](#)
- [ML-Series QoS, page 13-3](#)
- [Configuring QoS, page 13-10](#)
- [Monitoring and Verifying QoS, page 13-16](#)
- [Configuration Examples, page 13-17](#)

The ML-Series card employs the Cisco IOS Modular QoS CLI (MQC). For more information about general MQC configuration, refer to the following IOS documents:

- *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.1* at this URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm
- *Cisco IOS Quality of Service Solutions Command Reference, Release 12.1* at this URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm

Understanding QoS

The ML-Series card multiplexes multiple IP/Ethernet services onto the SONET/SDH circuit and dynamically allocate transmission bandwidth to data services based on data service requirements, which allow the network to operate at a significantly higher level of utilization. To support service-level agreements (SLAs), this dynamic allocation must accommodate the service elements of bandwidth, including loss and delay. The characteristics of these service elements make up QoS.

Priority Mechanism in IP and Ethernet

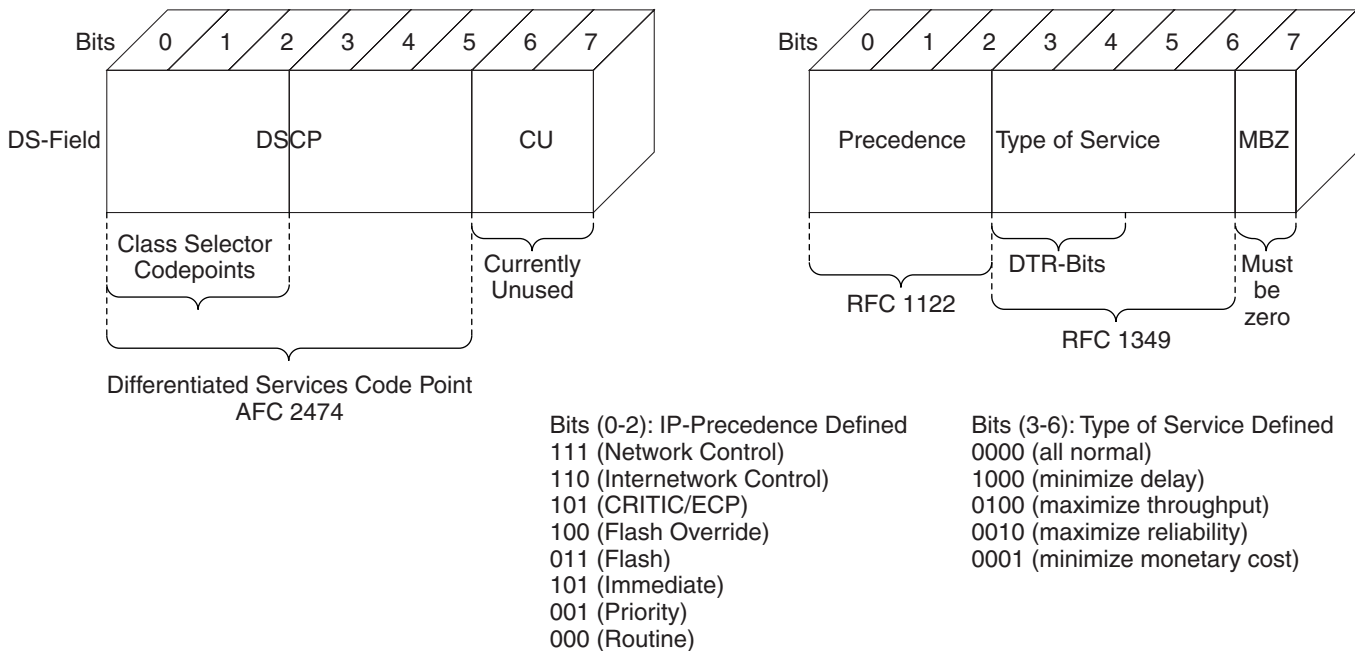
For any QoS service to be applied to data, there must be a way to mark or identify an IP packet or an Ethernet frame. When identified, a specific priority may be assigned to each individual IP packet or Ethernet frame. The IP Precedence or the IP Differentiated Services Code Point (DSCP) field prioritizes the IP packets, and the Ethernet class of service (IEEE 802.1p defined CoS) is used for the Ethernet frames. IP precedence and Ethernet CoS are further described below.

IP Precedence and Differentiated Services Code Point

IP precedence uses the three precedence bits in the IPv4 header's ToS (Type of Service) field to specify class of service for each IP packet (RFC 1122). The most significant three bits on the IPv4 ToS field provides up to eight distinct classes, of which six are used for classifying services and the remaining two are reserved. On the edge of the network, the IP Precedence is assigned by the client device or the router, so that each subsequent network element can provide services based on the determined policy or the service level agreement (SLA).

IP Differentiated Services Code Point (DSCP) uses the six bits in the IPv4 header to specify class of service for each IP packet (RFC 2474). [Figure 13-1](#) illustrates IP precedence and DSCP. The DSCP field classifies packets into any of the 64 possible classes. On the network edge the IP DSCP is assigned by the client device or the router, so that each subsequent network element can provide services based on the determined policy or the service level agreement.

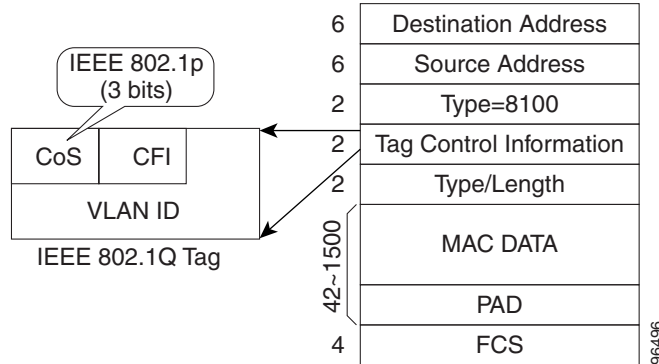
Figure 13-1 IP Precedence and Differentiated Service Code Point (DSCP)



Ethernet CoS

Ethernet CoS refers to three bits within a four byte IEEE 802.1Q (VLAN) header used to indicate the priority of the Ethernet frame as it passes through a switched network. The CoS bits in the 802.1Q header are commonly referred to as the 802.1p bits. There are three CoS bits which provide eight classes, matching the number delivered by IP Precedence. In many real-world networks, a packet may traverse both Layer 2 and Layer 3 domains. To maintain QoS across the network, the IP ToS can be mapped to the Ethernet CoS and vice versa, for example in linear or one-to-one mapping, because each mechanism supports eight classes. Similarly, a set of DSCP values (64 classes) may be mapped into each of the eight individual Ethernet CoS values. [Figure 13-2](#) is an IEEE 802.1Q Ethernet frame, which consists of a 2 byte Ethertype and a 2-byte tag (IEEE 802.1Q Tag) on the Ethernet protocol header.

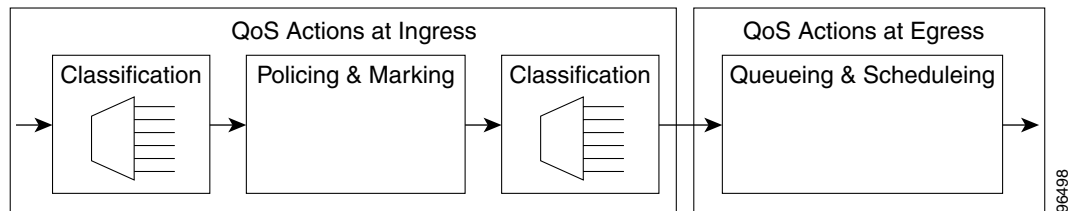
Figure 13-2 Ethernet frame and the CoS bit (802.1p)



ML-Series QoS

The ML-Series QoS classifies each packet in the network based on its input interface, Bridge Group (VLAN), Ethernet CoS, IP Precedence, IP DSCP, or RPR-CoS. Once classified into class flows, further QoS functions can be applied to each packet as it traverses the card. Figure 13-3 illustrates the ML-Series QoS flow.

Figure 13-3 ML-Series QoS Flow



Policing provided by the ML-Series card ensures that attached equipment does not submit more than a pre-defined amount of bandwidth (Rate Limiting) into the network. The policing feature can be used to enforce the committed information rate (CIR) and the peak information rate (PIR) available to a customer at an interface. Policing also helps characterize the statistical nature of the information allowed into the network so that traffic engineering can more effectively ensure that the amount of committed bandwidth is available on the network, and the peak bandwidth is over-subscribed with an appropriate ratio. The policing action is applied per classification.

Priority marking can set the Ethernet 802.1p CoS bits or RPR-CoS bits as they exit the ML-Series card. The marking feature operates on the outer 802.1p tag, and provides a mechanism for tagging packets at the ingress of a QinQ packet. The subsequent network elements can provide QoS based only on this service-provider-created QoS indicator.

Per class flow queuing enables fair access to excess network bandwidth, allows allocation of bandwidth to support SLAs, and ensures that applications with high network resource requirements are adequately served. Buffers are allocated to queues dynamically from a shared resource pool. The allocation process incorporates the instantaneous system load as well as the allocated bandwidth to each queue to optimize buffer allocation. Congestion management on the ML-Series is performed through a tail drop mechanism along with discard eligibility on the egress scheduler.

The ML-Series uses a Weighted Deficit Round Robin (WDRR) scheduling process to provide fair access to excess bandwidth as well as guaranteed throughput to each class flow.

Admission control is a process that is invoked each time that service is configured on the ML-Series card to ensure that QoS resources are not over committed. In particular, admission control ensures that no configurations are accepted, where a sum of the committed bandwidths on an interface, exceeds total bandwidth on the interface.

Classification

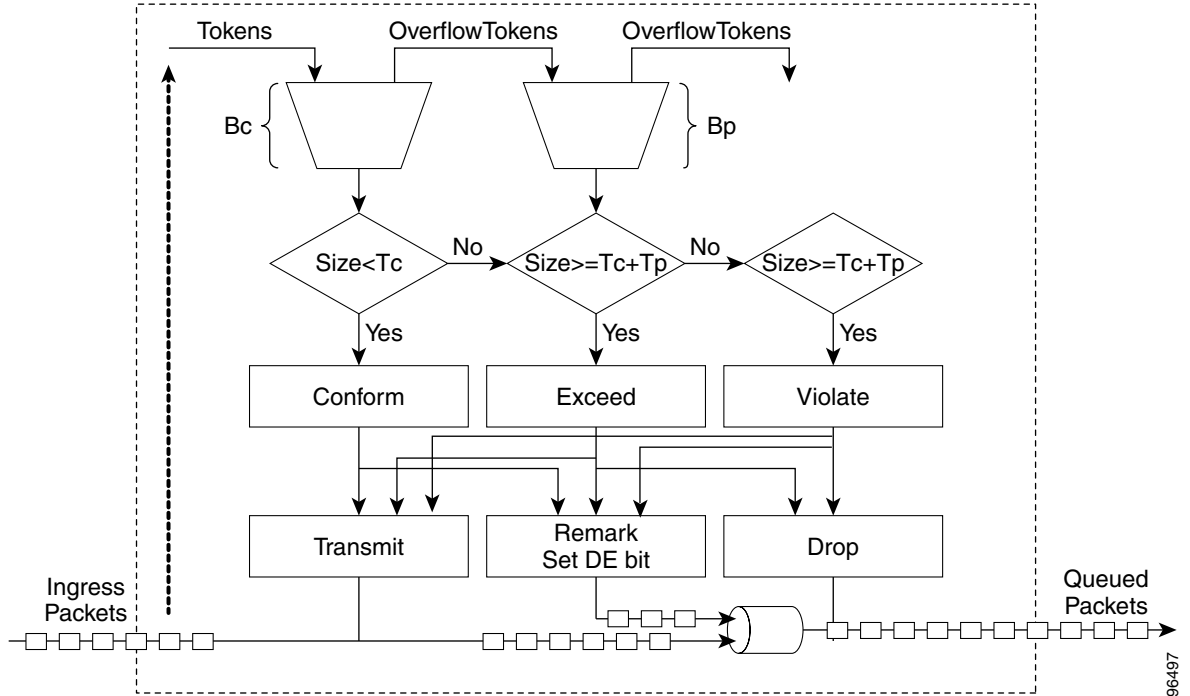
Classification can be based on any single packet classification criteria or a combination (logical AND and OR). A total of 254 classes, not including the class default, can be defined on the card. Classification of packets is configured using the Modular CLI **class-map** command. For traffic transiting the RPR, only the Input Interface and/or the RPR-CoS can be used as classification criteria.

Policing

Dual leaky bucket policer is a process where the first bucket (CIR bucket) is filled with tokens at a known rate (CIR), which is a parameter that can be configured by the operator. [Figure 13-4](#) illustrates the dual leaky bucket policer model. The tokens fill the bucket up to a maximum level, which is the amount of burstable committed (BC) traffic on the policer. The non-conforming packets of the first bucket are the overflow packets, which are passed to the second leaky bucket (the PIR bucket). The second leaky bucket is filled with these tokens at a known rate (PIR), which is a parameter that can be configured by the operator. The tokens fill the PIR bucket up to a maximum level (BP), which is the amount of peak burstable traffic on the policer. The non-conform packets of the second bucket are the overflow packets, which can be dropped or marked according to the policer definition.

On the dual leaky bucket policer, the packets conforming to the CIR are conform packets, the packets not conforming to CIR but conforming to PIR are exceed packets, and the packets not conforming to either the PIR or CIR are violate packets.

Figure 13-4 Dual Leaky Bucket Policer Model



Marking and Discarding

On the ML-Series card's policer, the conform packets can be transmitted or marked and transmitted. The exceed packets can be transmitted, marked and transmitted, or dropped. The violating packets can be transmitted, marked and transmitted, or dropped. The primary application of the dual-rate or three-color policer is to mark the conform packets with CoS bit 21, mark the exceed packet with CoS bit 1, and discard the violated packets so all the subsequent network devices can implement the proper QoS treatment per frame/packet basis based on these priority marking without knowledge of each SLA.

If a marked packet has a provider-supplied Q-Tag inserted before transmission, the marking only affects the provider Q-Tag. If a Q-Tag was received, it will be re-marked. If a marked packet is transported over the RPR ring, the marking also affects the RPR-CoS bit.

If a Q-Tag is inserted (QinQ), the marking affects the added Q-Tag. If the ingress packet contains a Q-tag and is transparently switched, the existing Q-Tag is marked. In case of a packet without any Q-Tag, the marking does not have any significance.

The local scheduler treats all non-conforming packets as discard eligible regardless of their CoS setting or the global cos commit definition. For RPR implementation, the discard eligible (DE) packets are marked using the DE bit on the RPR header. The discard eligibility based on the CoS Commit or the policing action is local to the ML-Series card scheduler, but it is global for the RPR ring.

Queuing

ML-Series card queuing uses a shared buffer pool to allocate memory dynamically to different traffic queues. The ML-Series card uses a total of 12 MB memory for the buffer pool. Ethernet ports share 6 MB of the memory, and POS ports share the remaining 6 MBs of memory. Memory space is allocated in 1500-byte increments.

Each queue has an upper limit on the allocated number of buffers based on the class bandwidth assignment of the queue and the number of queues configured. This upper limit is typically 30% to 50% of the shared buffer capacity. Dynamic buffer allocation to each queue may be reduced based on the number of queues needing extra buffering. The dynamic allocation mechanism provides fairness in proportion to service commitments as well as optimization of system throughput over a range of system traffic loads.

The Low Latency Queue (LLQ) is defined by setting the weight to infinity or committing 100% bandwidth. When a LLQ is defined, a policer should also be defined on the ingress for that specific class to limit the maximum bandwidth consumed by the LLQ; otherwise there may be a potential risk of LLQ occupying the whole bandwidth and starving the other unicast queues.

The ML-Series includes support for 400 user-definable queues, which are assigned per the classification and bandwidth allocation definition. The classification used for scheduling classifies the frames/packet after the policing action, so if the policer is used to mark or change the CoS bits of the ingress frames/packet, the new values are applicable for the classification of traffic for queuing and scheduling. The ML-Series provides buffering for 4000 packets.

Scheduling

Scheduling is provided by a series of schedulers that perform a weighted deficit round robin (WDRR) as well as priority scheduling mechanisms from the queued traffic associated with each egress port.

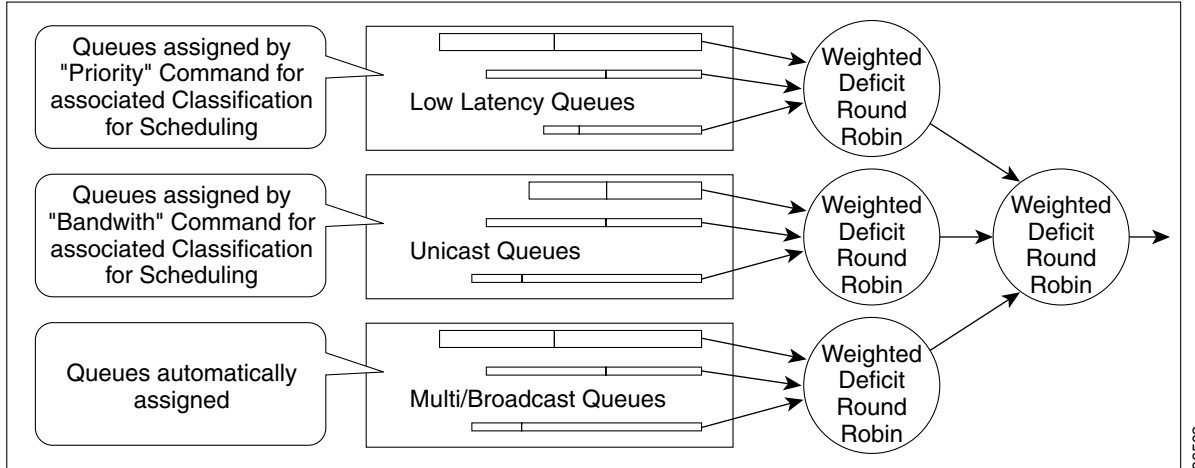
Though ordinary round robin servicing of queues can be done in constant time, the unfairness that occurs when different queues use different packet sizes. Deficit Round Robin (DRR) scheduling solves this problem. If a queue was not able to send a packet in its previous round because its packet size was too large, the remainder from the previous amount of credits a queue gets in each round (quantum) is added to the quantum for the next round.

WDRR extends the quantum idea from the DRR to provide weighted throughput for each queue. Different queues have different weights, and the quantum assigned to each queue in its round is proportional to the relative weight of the queue among all the queues serviced by that scheduler.

Weights are assigned to each queue as a result of the service provisioning process. When coupled with policing and policy mapping provisioning, these weights and the WDRR scheduling process ensure that QoS commitments are provided to each service flow.

Figure 13-5 illustrates the ML-Series card's queuing and scheduling.

Figure 13-5 Queuing and Scheduling Model



The weighting structure allows traffic to be scheduled at 1/2048 of the port rate. This equates to approximately 488 kbps for traffic exiting a GigabitEthernet port, approximately 293 kbps for traffic exiting an OC-12c port, and approximately 49 kbps for traffic exiting a FastEthernet port.

The multicast/broadcast queue is automatically created on every egress port of the ML-Series card with a committed bandwidth of 10%. This queue is used for multicast/broadcast data traffic, control traffic, L2 protocol tunnelling, and flooding traffic of the unknown MAC during MAC learning. If the aggregate of multicast/broadcast traffic at any egress port exceeds 10% of the bandwidth those frames beyond 10% of the bandwidth, are treated as best effort by the scheduler.

The unicast queues are created as the output service policy implementation on the egress ports. Each unicast queue is assigned with a committed bandwidth and the weight of the queue is determined by the normalization of committed bandwidth of all defined unicast queues for that port. The traffic beyond the committed bandwidth on any queue is treated by the scheduler according to the relative weight of the queue.

The LLQ is created as the output service policy implementation on the egress ports. Each LLQ queue is assigned with a committed bandwidth of 100% and is served with lower latency. To limit the bandwidth usage by the LLQ, a strict policer needs to be implemented on the ingress for the LLQ traffic classes.

The discard eligibility (DE) allows some packets to be treated as committed and some as discard-eligible on the scheduler. For the Ethernet frames the CoS (802.1p) bits are used to identify committed and discard eligible packets, where the RPR-CoS and the DE bits are used for RPR traffic. When congestion occurs and a queue begins to fill, the discard-eligible packets hit a lower tail-drop threshold than the committed packets. Committed packets are not dropped until the total committed load exceeds the interface output. The tail-drop thresholds adjust dynamically in the card to maximize use of the shared buffer pool while guaranteeing fairness under all conditions.

Multicast QoS

On the ML-Series cards, multicast (including IP-multicast) and broadcast traffic forwarding is supported at line-rate; however the QoS implementation on multicast traffic varies from the unicast QoS. The difference is in the priority handling for the multicast traffic on the scheduler.

For unicast packets, the priority is defined by the **bandwidth** command, which creates a CIR for the unicast packets in a particular class.

The priority handling of multicast packets is not based on the **bandwidth** command. Instead, multicast frames are assigned to a queue that has a committed bandwidth of 10% of the port bandwidth. If the multicast and broadcast traffic exceeds 10% of the port bandwidth, frames exceeding 10% are given low priority (best effort). The 10% committed bandwidth for multicast is applied to the aggregate traffic and does not allow the multicast traffic of one customer to be given higher priority than another customer, unlike the QoS model for unicast traffic.

The scheduler allocates 10% of the bandwidth for multicast and broadcast traffic. Any other QoS implementation is not applicable for multicast and broadcast traffic except the allocation of 10% bandwidth for all multicast/broadcast traffics. Buffers are allocated to queues dynamically from a shared resource pool.

Control Packets and L2 Tunneled Protocols

The control packets originated by the ML-Series card have a higher priority than data packets. The external Layer 2 and Layer 3 control packets are handled as data packets and assigned to broadcast queues. Bridge protocol data unit (BPDU) prioritization in the ML-Series card gives Layer 2-tunneled BPDU sent out the multicast/broadcast queue a higher discard value and therefore a higher priority than other packets in the multicast/broadcast queue. The Ethernet CoS (802.1p) for Layer 2-tunneled protocols can be assigned by the ML-Series card.

Priority Marking

Priority marking allows the operator to assign the 802.1p CoS bits of packets that exit the card. This marking allows the operator to use the CoS bits as a mechanism for signaling to downstream nodes the QoS treatment the packet should be given. This feature operates on the outer-most 802.1p CoS field. When used with the QinQ feature, priority marking allows the user traffic (inner Q-Tag) to traverse the network transparently, while providing a means for the network to internally signal QoS treatment at Layer 2.

Priority marking follows the classification process, and therefore any of the classification criteria identified earlier can be used as the basis to set the outgoing 802.1p CoS field. For example, a specific CoS value can be mapped to a specific Bridge Group.

Priority marking is configured using the MQC **set-cos** command. If packets would otherwise leave the card without an 802.1q tag, then the set cos will have no effect on that packet. If an 802.1q tag is inserted in the packet (either a normal tag or a Q-in-Q tag), the inserted tag will have the set cos priority. If an 802.1q tag is present on packet ingress and retained on packet egress, the priority of that tag will be modified. If the ingress interface is an QinQ Access port, and the **set cos** policy-map classifies based on ingress tag priority, this will classify based on the user priority. This is a way to allow the user-tag priority to determine the Service Provider tag priority. When a packet does not match any **set cos** policy-map, the priority of any preserved tag is unchanged and the priority of any inserted 802.1q tag is set to 0.

The **set-cos** command on the output service policy is only applied to unicast traffic. Priority marking for multicast/broadcast traffic can only be achieved by the set cos action of the policing process on the input service policy.

QinQ Implementation

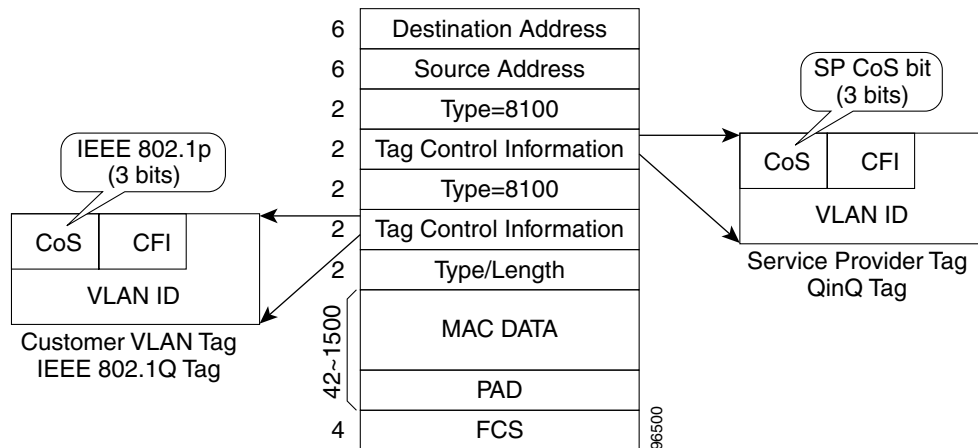
The hierarchical VLAN or 802.1Q tunneling feature enables the service provider to transparently carry the customer VLANs coming from any specific port (UNI) and transport them over the service provider network. This feature is also known as QinQ, which is performed by adding an additional 802.1Q tag on every customer frame.

Using the QinQ feature, service providers can use a single VLAN to support customers with multiple VLANs. QinQ preserves customer VLAN IDs and segregates traffic from different customers within the service-provider infrastructure, even when traffic from different customers originally shared the same VLAN ID. The QinQ also expands VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets. When the service provider tag is added, the QinQ network typically loses any visibility to the IP header or the customer Ethernet 802.1Q tag on the QinQ encapsulated frames.

On the ML-Series cards, the QinQ access ports (802.1Q tunnel ports or QinQ UNI ports) have visibility to the customer CoS and the IP Precedence or IP DSCP values; therefore, the SP tag can be assigned with proper CoS bit which would reflect the customer IP Precedence, IP DSCP, or CoS bits. In the QinQ network, the QoS is then implemented based on the 802.1p bit of the service provider tag. The ML-Series cards do not have visibility into the customer CoS or IP Precedence or DSCP values after the packet is double-tagged (that is beyond the entry point of the QinQ service).

Figure 13-6 illustrates the QinQ implementation on the ML-Series card.

Figure 13-6 QinQ



The ML-Series cards can be used as the 802.1Q tunneling device for the QinQ network and also provide the option to copy the customer frame's CoS bit into the CoS bit of the added QinQ tag. This way the service provider QinQ network can be fully aware of the necessary QoS treatment for each individual customer frames.

Flow Control Pause and QoS

If flow control and port-based policing are both enabled for an interface, flow control handles the bandwidth. If the policer gets non-compliant flow, then the policer drops or demarks the packets using the policer definition of the interface.

**Note**

QoS and policing are not supported on the ML-Series card interface when link aggregation is used.

**Note**

Egress shaping is not supported on the ML-Series cards.

QoS on RPR

For VLAN bridging over RPR, all ML-Series cards on the ring must be configured with the base RPR and RPR QoS configuration. SLA and bridging configurations are only needed at customer RPR access points, where 802.1q VLAN CoS is copied to the RPR CoS. This 802.1q VLAN CoS copying can be overwritten with a **set cos <action>** command. The cos commit rule applies at RPR ring ingress. Transit RPR ring traffic is classified on CoS only.

If the packet does not have a VLAN header, the RPR CoS for non-VLAN traffic is set using the following rules:

1. The default CoS is 0.
2. If the packet comes in with an assigned CoS. The assigned CoS replaces the default, or if an IP packet originates locally, the IP Precedence setting replaces the CoS setting.
3. The input policy map has a **set cos** action.
4. The output policy map has a **set cos** action (except for broadcast or multicast packets).

The RPR header contains a CoS value and DE indicator. The RPR DE is set for non-committed traffic.

Configuring QoS

This section describes the tasks for configuring the ML-Series card QoS functions using the Modular Quality of Service Command-Line Interface (MQC). The ML-Series card does not support the full set of MQC functionality.

To configure and enable class-based QoS features, perform the procedures described in the following sections:

- [Creating a Traffic Class, page 13-11](#)
- [Creating a Traffic Policy, page 13-12](#)
- [Attaching a Traffic Policy to an Interface, page 13-15](#)
- [Configuring CoS-based QoS, page 13-16](#)
- [Monitoring and Verifying QoS, page 13-16 \(Optional\)](#)

For QoS configuration examples, see the “[Configuration Examples](#)” section on page 13-17.

Creating a Traffic Class

The **class-map** global configuration command is used to create a traffic class. The syntax of the **class-map** command is as follows:

```
class-map [match-any | match-all] class-name
no class-map [match-any | match-all] class-name
```

The match-all and match-any options need to be specified only if more than one match criterion is configured in the traffic class. The **class-map match-all** command is used when all of the match criteria in the traffic class must be met for a packet to match the specified traffic class. The **class-map match-any** command is used when only one of the match criterion in the traffic class must be met for a packet to match the specified traffic class. If neither the **match-all** nor **match-any** keyword is specified, the traffic class will behave in a manner consistent with **class-map match-all** command.

To create a traffic class containing match criteria, use the **class-map** global configuration command to specify the traffic class name, and then use the following **match** commands in class-map configuration mode, as needed:

| Command | Purpose |
|---|--|
| Router(config)# class-map <i>class-map-name</i> | Specifies the user-defined name of the traffic class. Names can be a maximum of 40 alphanumeric characters. If match-all or match-any are not specified, traffic must match all the match criteria to be classified as part of the traffic class. There is no default-match criteria. Multiple match criteria are supported. The command matches either ALL or ANY of the criteria, as controlled by the match-all and match-any subcommands of the class-map command. |
| Router(config)# class-map match-all <i>class-map-name</i> | Specifies that all match criteria must be met for traffic entering the traffic class to be classified as part of the traffic class. |
| Router(config)# class-map match-any <i>class-map-name</i> | Specifies that one of the match criteria must be met for traffic entering the traffic class to be classified as part of the traffic class. |
| Router(config-cmap)# match any | Specifies that all packets will be matched. |
| Router(config-cmap)# match bridge-group <i>bridge-group-number</i> | Specifies the bridge-group-number against whose contents packets are checked to determine if they belong to the class. |
| Router(config-cmap)# match cos <i>cos-number</i> | Specifies the CoS value against whose contents packets are checked to determine if they belong to the class. |

| Command | Purpose |
|---|---|
| Router(config-cmap)# match input-interface <i>interface-name</i> | Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class. The shared packet ring (SPR) interface, spr1 , used in resilient packet ring (RPR) is a valid interface-name for the ML-Series card. For more information on the SPR interface, see Chapter 16, “Configuring Resilient Packet Ring.” The input-interface choice is not valid when applied to the INPUT of an interface (redundant). |
| Router(config-cmap)# match ip dscp <i>ip-dscp-value</i> | Specifies up to eight differentiated services code point (DSCP) values used as match criteria. The value of each service code point is from 0 to 63. |
| Router (config-cmap)# match ip precedence <i>ip-precedence-value</i> | Specifies up to eight IP Precedence values used as match criteria. |

Creating a Traffic Policy

To configure a traffic policy, use the **policy-map** global configuration command to specify the traffic policy name, and use the following configuration commands to associate a traffic class, which was configured with the **class-map** command and one or more QoS features. The traffic class is associated with the traffic policy when the **class** command is used. The **class** command must be issued after entering policy-map configuration mode. After entering the **class** command, you are automatically in policy-map class configuration mode, which is where the QoS policies for the traffic policy are defined.

When the bandwidth or priority action is used on any class in a policy map, then there must be a class defined by the **match-any** command, which has a bandwidth or priority action in that policy map. This is to ensure that all traffic can be classified into a default class which has some assigned bandwidth. A minimum bandwidth can be assigned if the class is not expected to be used or no reserved bandwidth is desired for default traffic.

The QoS policies that can be applied in the traffic policy in policy-map class configuration mode are detailed in the following example:

The syntax of the **policy-map** command is:

```
policy-map policy-name
no policy-map policy-name
```

The syntax of the **class** command is:

```
class class-name
no class class-name
```

All traffic that fails to meet the matching criteria belongs to the default traffic class. The default traffic class can be configured by the user, but cannot be deleted.

To create a traffic policy, use the following commands as needed, beginning in global configuration mode:

| Command | Purpose |
|--|---|
| Router (config)# policy-map <i>policy-name</i> | Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters. |
| Router (config-pmap)# class <i>class-name</i> | Specifies the name of a predefined traffic class, which was configured with the class-map command, used to classify traffic to the traffic policy. |
| Router (config-pmap)# class class-default | Specifies the default class to be created as part of the traffic policy. |
| Router (config-pmap-c)# bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> } | <p>Specifies a minimum bandwidth guarantee to a traffic class in periods of congestion. A minimum bandwidth guarantee can be specified in kbps or by a percentage of the overall available bandwidth.</p> <p>Valid choices for the ML-Series card are:</p> <ul style="list-style-type: none"> • Rate in kilobits per second (8 to 2000000) • Percent of total available bandwidth (1 to 100) <p>If multiple classes and bandwidth actions are specified in a single policy map, they must use the same choice in specifying bandwidth (kilobits or percent).</p> <p>Note When using the bandwidth command, excess traffic (beyond the configured commit) will be allocated any available bandwidth in proportion to the relative bandwidth commitment of its traffic class compared to other traffic classes. Excess traffic from two classes with equal commits will have equal access to available bandwidth. Excess traffic from a class with a minimum commit might receive only a minimum share of available bandwidth compared to excess bandwidth from a class with a high commit.</p> |

| Command | Purpose |
|---|--|
| <pre>Router (config-pmap-c)# police <i>cir-rate-bps</i> <i>normal-burst-byte</i> [<i>max-burst-byte</i>] [pir <i>pir-rate-bps</i>] [conform-action {set-cos-transmit transmit}] [exceed-action {set-cos-transmit drop}] [violate-action {set-cos-transmit transmit}]</pre> | <p>Defines a policer for the currently selected class when the policy map is applied to input. Policing is supported only on ingress, not on egress.</p> <ul style="list-style-type: none"> • For <i>cir-rate-bps</i>, specify the average committed information rate (cir) in bits per second (bps). The range is 96000 to 800000000. • For <i>normal-burst-byte</i>, specify the cir burst size in bytes. The range is 8000 to 64000. • (Optional) For <i>maximum-burst-byte</i>, specify the peak information rate (pir) burst in bytes. The range is 8000 to 64000. • (Optional) For <i>pir-rate-bps</i>, specify the average pir traffic rate in bps where the range is 96000 to 800000000. • (Optional) Conform action options are: <ul style="list-style-type: none"> – Set a CoS priority value and transmit – Transmit packet (default) • (Optional) Exceed action options are: <ul style="list-style-type: none"> – Set a CoS value and transmit – Drop packet (default) • (Optional) The violate action is only valid if pir is configured. Violate action options are: <ul style="list-style-type: none"> – Set a CoS value and transmit – Drop packet (default) |
| <pre>Router (config-pmap-c)# priority <i>kbps</i></pre> | <p>Specifies low latency queuing for the currently selected class. This command can only be applied to an output. When the policy-map is applied to an output, an output queue with strict priority is created for this class. The only valid rate choice is in kilobits per second (8 to 2000000).</p> <p>Note When using the priority action, the traffic in that class is given a 100% commit (CIR), regardless of the rate entered as the priority rate. To ensure that CIR commitments are met for the interface, a policer must be configured on the input of all interfaces that might deliver traffic to this output class, limiting the peak rate to the priority rate entered.</p> |

| Command | Purpose |
|--|--|
| <pre>Router (config-pmap-c)# set-cos <i>cos-value</i></pre> | <p>Specifies a class of service (CoS) value or values to associate with the packet. The number is in the range from 0 to 7.</p> <p>This command may only be used in a policy-map applied to an output. It specifies the VLAN CoS priority to set for the outbound packets in the currently selected class. If QinQ is used, the top-level VLAN tag is marked. If outbound packets have no VLAN tag, the action has no effect. This action is applied to the packet after any set cos action done by a policer, and will therefore override the CoS set by a policer action.</p> <p>If a packet is marked by the policer and forwarded out an interface that also has a set cos action assigned for the traffic class, the value specified by the police action will take precedence in setting the 802.1p CoS field.</p> <p>This command also sets the CoS value in the RPR header for packets exiting the ML-Series on the RPR interface.</p> |
| <pre>Router (config-pmap-c)# set-cos-tranmit <i>cos-value</i></pre> | <p>Sets (marks) a VLAN CoS priority associated with the packets under/over the policed rate.</p> <p>The classification of any output policy map will be based on the new CoS value set (even if the packet were to enter and/or exit without a VLAN tag). An output policy map may override the VLAN CoS priority set for the packet. When the packet is transmitted out an interface, the associated VLAN CoS priority will be written into the VLAN tag. If QinQ is used, the top-level VLAN tag at the time of transmit will be marked. If the packet is transmitted without a VLAN tag, no marking will occur.</p> <p>This command also sets the CoS value in the RPR header for packets exiting the ML-Series on the RPR interface.</p> |

Attaching a Traffic Policy to an Interface

Use the **service-policy** interface configuration command to attach a traffic policy to an interface and to specify the direction in which the policy should be applied (either on packets coming into the interface or packets leaving the interface). Only one traffic policy can be applied to an interface in a given direction.

Use the **no** form of the command to detach a traffic policy from an interface. The **service-policy** command syntax is as follows:

```
service-policy {input | output} policy-map-name
no service-policy {input | output} policy-map-name
```

To attach a traffic policy to an interface, use the following commands in interface configuration mode, as needed:

| Command | Purpose |
|--|--|
| Router(config)# interface <i>interface-id</i> | Enters interface configuration mode, and specifies the interface to apply the policy map. Valid interfaces are limited to physical Ethernet and POS interfaces. Note Policy maps cannot be applied to SPR interfaces , subinterfaces, port channel interfaces, or BVIs. |
| Router(config-if)# service-policy output <i>policy-map-name</i> | Specifies the name of the traffic policy to be attached to the output direction of an interface. The traffic policy evaluates all traffic leaving that interface. |
| Router(config-if)# service-policy input <i>policy-map-name</i> | Specifies the name of the traffic policy to be attached to the input direction of an interface. The traffic policy evaluates all traffic entering that interface. |

Configuring CoS-based QoS

Cisco ONS 15454/ONS 15454 SDH Software Release 4.1 introduces the global **cos commit** *<cos value>* command. This command allows the ML-Series card to base the QoS treatment for a packet coming in on a network interface on the attached CoS value, rather than on a per-customer-queue policer.

CoS-based QoS is applied with a single global **cos commit** *<cos value>* command:

| Command | Purpose |
|--|--|
| Router(config)# cos-commit <i><cos value></i> | Labels packets that come in with a CoS equal to or higher than the <i>cos value</i> as CIR and packets with a lower CoS as DE. |

Monitoring and Verifying QoS

To display the information relating to a traffic class or traffic policy, use one of the following commands in EXEC mode, as needed.

Table 13-1 Commands for QoS Status

| Command | Purpose |
|--|--|
| Router# show class-map <i>name</i> | Displays the traffic class information of the user-specified traffic class. |
| Router# show policy-map | Displays all configured traffic policies. |
| Router# show policy-map <i>name</i> | Displays the user-specified policy map. |
| Router# show policy-map <i>interface</i> <i>interface</i> | Displays configurations of all input and output policies attached to an interface. Statistics displayed with this command are unsupported and show zero. |

Examples of these commands are shown here:

```
Router# show class-map
Class Map match-any class-default (id 0)
  Match any
Class Map match-all policer (id 2)
  Match ip precedence 0

Router# show policy-map
Policy Map police_f0
  class policer
    police 1000000 10000 conform-action transmit exceed-action drop

Router# show policy-map interface

FastEthernet0

  service-policy input: police_f0

  class-map: policer (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    match: ip precedence 0

  class-map: class-default (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    match: any
      0 packets, 0 bytes
      5 minute rate 0 bps
```

Configuration Examples

This section provides the specific command and network configuration examples:

- [Traffic Classes Defined Example](#)
- [Traffic Policy Created Example](#)
- [class-map match-any and class-map match-all Commands Example](#)
- [class-map match-any and class-map match-all Commands Example](#)
- [match spr1 Interface Example](#)
- [ML-Series VoiP Example](#)
- [ML-Series Policing Example](#)
- [ML-Series CoS-based QoS Example](#)

Traffic Classes Defined Example

This example shows how to create a class map called class1 that matches incoming traffic entering interface fastethernet0:

```
Router(config)# class-map class1
Router(config-cmap)# match input-interface fastethernet0
```

This example shows how to create a class map called class2 that matches incoming traffic with IP-precedence values of 5, 6, and 7:

```
Router(config)# class-map match-any class2
Router(config-cmap)# match ip precedence 5 6 7
```

**Note**

If a class-map contains a match rule which specifies multiple values, such as 5 6 7 in this example, then the class-map must be match-any, not the default match-all. Without the match-any an error message is printed and the class is ignored. The supported commands which allow multiple values are **match cos**, **match ip precedence** and **match ip dscp**.

This example shows how to create a class map called class3 that matches incoming traffic based on bridge group 1:

```
Router(config)# class-map class3
Router(config-cmap)# match bridge-group 1
```

Traffic Policy Created Example

In the following example, a traffic policy called policy1 is defined to contain policy specifications, including a bandwidth allocation request, for the default class and two additional classes—class1 and class2. The match criteria for these classes were defined in the traffic classes, see the [“Creating a Traffic Class” section on page 13-11](#).

```
Router(config)# policy-map policy1
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth 1000
Router(config-pmap)# exit
```

```
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap)# exit
```

```
Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap)# exit
```

class-map match-any and class-map match-all Commands Example

This section illustrates the difference between the **class-map match-any** command and the **class-map match-all** command. The **match-any** and **match-all** options determine how packets are evaluated when multiple match criteria exist. packets must either meet all of the match criteria (**match-all**) or one of the match criteria (**match-any**) in order to be considered a member of the traffic class.

The following example shows a traffic class configured with the **class-map match-all** command:

```
Router(config)# class-map match-all cisco1
Router(config-cmap)# match cos 1
Router(config-cmap)# match bridge-group 10
```

If a packet arrives with a traffic class called cisco1 configured on the interface, the packet is evaluated to determine if it matches the cos 1 and bridge group 10. If both of these match criteria are met, the packet matches traffic class cisco1.

The following example shows a traffic class configured with the **class-map match-any** command:

```
Router(config)# class-map match-any cisco2
Router(config-cmap)# match cos 1
Router(config-cmap)# match bridge-group 10
Router(config-cmap)# match ip dscp 5
```

In traffic class called cisco2, the match criteria are evaluated consecutively until a successful match criterion is located. The packet is first evaluated to determine whether cos 1 can be used as a match criterion. If cos 1 can be used as a match criterion, the packet is matched to traffic class cisco2. If cos 1 is not a successful match criterion, then bridge-group 10 is evaluated as a match criterion. Each matching criterion is evaluated to see if the packet matches that criterion. Once a successful match occurs, the packet is classified as a member of traffic class cisco2. If the packet matches none of the specified criteria, the packet is classified as a member of the traffic class.

Note that the **class-map match-all** command requires that all of the match criteria must be met in order for the packet to be considered a member of the specified traffic class (a logical AND operator). In the example, cos 1 AND bridge group 10 have to be successful match criteria. However, only one match criterion must be met for the packet in the **class-map match-any** command to be classified as a member of the traffic class (a logical OR operator). In the example, cos 1 OR bridge group 10 OR ip dscp 5 have to be successful match criteria.

match spr1 Interface Example

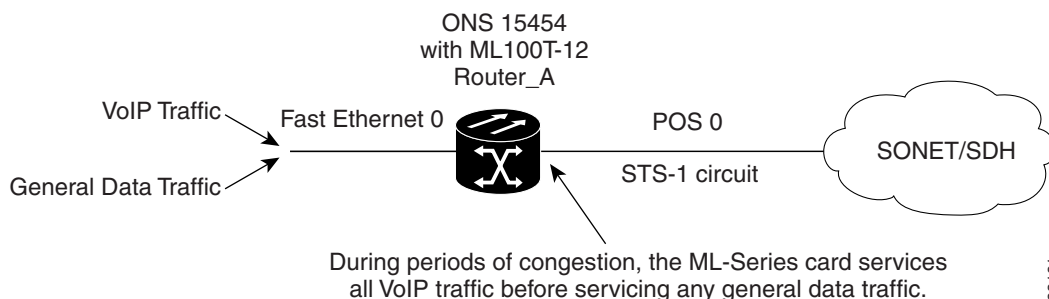
In the following example, the SPR interface is specified as a parameter to the <match input-interface> CLI when defining a class-map.

```
Router(config)#class-map spr1-cos1
Router(config-cmap)#match input-interface spr1
Router(config-cmap)#match cos 1
Router(config-cmap)#end
Router#sh class-map spr1-cos1
Class Map match-all spr1-cos1 (id 3)
Match input-interface SPR1
Match cos 1
```

ML-Series VoIP Example

Figure 13-7 shows an example of ML-Series QoS. The associated commands are provided in the sections that follow the figure.

Figure 13-7 ML-Series VoIP Example



```
class-map match-all voip
```

```

    match ip precedence 5
class-map match-any default
  match any
!
!
policy-map pos0
  class default
    bandwidth 1000

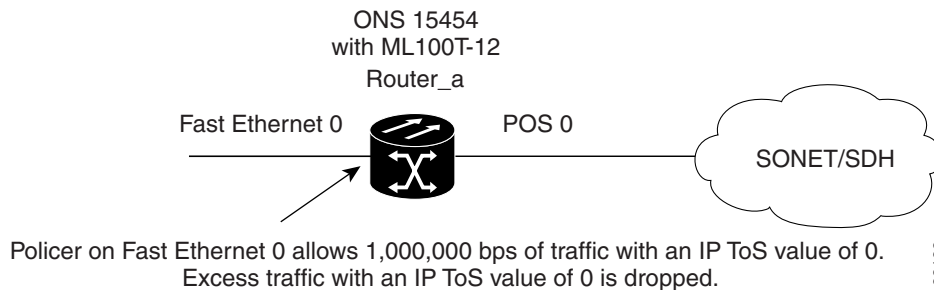
  class voip
    priority 1000
!
interface FastEthernet0
  ip address 1.1.1.1 255.255.255.0
!
interface POS0
  ip address 2.1.1.1 255.255.255.0
  service-policy output pos0
  crc 32
  no cdp enable
  pos flag c2 1

```

ML-Series Policing Example

Figure 13-8 shows an example of ML-Series policing. The example shows how to configure a policer that will restrict traffic with an IP precedence of 0 to 1,000,000 bps. The associated code is provided in the sections that follow the figure.

Figure 13-8 ML-Series Policing Example



```

!
class-map match-all policer
  match ip precedence 0
!
policy-map police_f0
  class policer
    police 1000000 10000 conform-action transmit exceed-action drop
!
interface FastEthernet0
  service-policy input police_f0
!

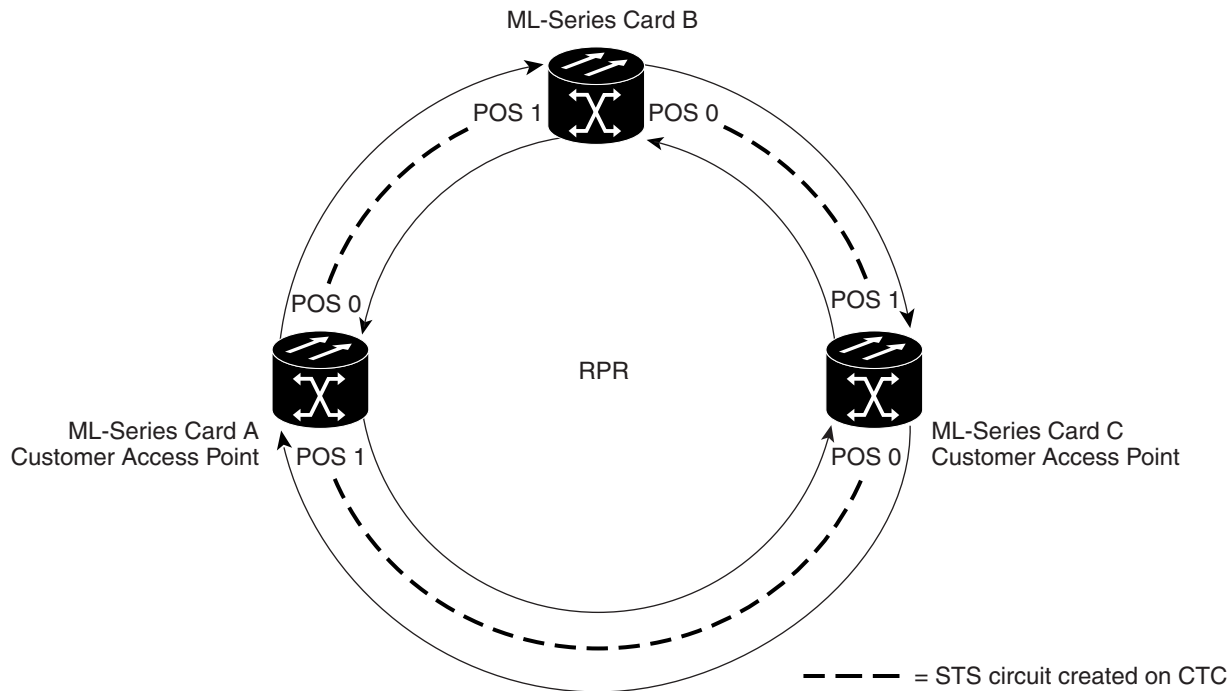
```

83182

ML-Series CoS-based QoS Example

Figure 13-9 shows an example of ML-Series CoS-based QoS. The associated code is provided in the sections following the figure. The CoS example assumes that the ML-Series cards are configured into an RPR and the ML-Series card POS ports are linked by point-to-point SONET circuits. For more information on configuring RPR, see Chapter 16, “Configuring Resilient Packet Ring.”

Figure 13-9 ML-Series CoS Example



96501

ML-Series A Configuration (Customer Access Point)

```
hostname ML-Series A
Cos commit 2

Policy-map Fast5_in
  class class-default
  police 5000 8000 8000 pir 10000 conform-action
  set-cos-transmit 2 exceed-action set-cos-transmit
  1 violate-action drop]
```

ML-Series B Configuration

```
hostname ML-Series B
Cos commit 2
```

ML-Series C Configuration (Customer Access Point)

```
hostname ML-Series C
```

```
Cos commit 2

Policy-map Fast5_in
  class class-default
    police 5000 8000 8000 pir 10000 conform-action
    set-cos-transmit 2 exceed-action set-cos-transmit
    1 violate-action drop
```



Configuring the Switching Database Manager

This chapter describes the switching database manager (SDM) features built into the ML100T-12 and ML1000-2 cards.

This chapter contains the following major sections:

- [Understanding the SDM, page 14-1](#)
- [Configuring SDM, page 14-2](#)

Understanding the SDM

ML-Series cards use the forwarding engine and ternary content-addressable memory (TCAM) to implement high-speed forwarding. The high-speed forwarding information is maintained in TCAM. The SDM is the software subsystem that manages the switching information maintained in TCAM.

SDM organizes the switching information in TCAM into application-specific regions and configures the size of these application regions. SDM enables exact-match and longest-match address searches, which result in high-speed forwarding. SDM manages TCAM space by partitioning application-specific switching information into multiple regions.

TCAM identifies a location index associated with each packet forwarded and conveys it to the forwarding engine. The forwarding engine uses this location index to derive information associated with each forwarded packet.

The key benefits of SDM in switching are its ability to organize the switching information in TCAM into application-specific regions and its ability to configure the size of these application regions. SDM enables exact-match and longest-match address searches, which result in high-speed forwarding.

SDM Regions

SDM partitions TCAM space into multiple application-specific regions and interacts with the individual application control layers to store switching information. SDM consists of the following types of regions:

- **Exact-match region**—The exact-match region consists of entries for multiple application regions such as IP adjacencies.

- Longest-match region—Each longest-match region consists of multiple buckets or groups of Layer 3 address entries organized in decreasing order by mask length. All entries within a bucket share the same mask value and key size. The buckets can change their size dynamically by borrowing address entries from neighboring buckets. Although the size of the whole application region is fixed, you can reconfigure it.
- Weighted-exact-match region—The weighted-exact-match region consists of exact-match-entries with an assigned weight or priority. For example, with QoS, multiple exact match entries may exist, but some have priority over others. The weight is used to select one entry when multiple entries match.

TCAM space consists of 65,536 entries, each entry being 64 bits wide. Because SDM is responsible for managing TCAM space, SDM partitions the entire TCAM space for each application region based on user configuration. Although the maximum size of all application regions is fixed, you can reconfigure the maximum size of each application region.

Table 14-1 lists default partitioning for each application region in TCAM.

Table 14-1 Default Partitioning by Application Region in TCAM

| Application Region | Lookup Type | Key Size | Default Size | No. of TCAM Entries |
|--------------------|----------------------|----------|----------------|---------------------|
| IP Adjacency | Exact-match | 64 bits | 65536 (shared) | 65536 (shared) |
| IP Prefix | Longest-match | 64 bits | 65536 (shared) | 65536 (shared) |
| QoS Classifiers | Weighted exact-match | 64 bits | 65536 (shared) | 65536 (shared) |
| IP VRF Prefix | Longest prefix match | 64 bits | 65536 (shared) | 65536 (shared) |
| IP Multicast | Longest prefix match | 64 bits | 65536 (shared) | 65536 (shared) |
| MAC Addr | Longest prefix match | 64 bits | 65536 (shared) | 65536 (shared) |
| Access List | Weighted exact match | 64 bits | 65536 (shared) | 65536 (shared) |

Configuring SDM

This section describes the commands necessary to configure the SDM. It includes commands to configure the size of the SDM regions. The commands described in this section are unique to the switching software.

Configuring SDM Regions

TCAM space consists of 65,536 entries, each entry being 64 bits wide. Since SDM is responsible for managing TCAM space, SDM partitions the entire TCAM space for each application region based on user configuration. A change in the partition configuration takes effect the next time you reboot the system.

The application region size in SDM is represented by the number of 64-bit entries. The combined size of all the application regions should be calculated in terms of 64-bit TCAM entries and should not exceed 65,536 bytes, which is the total TCAM size.

To configure SDM maximum size for each application region, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|--|---|
| Step 1 | Router(config)# sdm size <i>region-name</i> [k-entries] <i>number-of-entries</i> | Sets the name of the application region whose size you want to configure. You can enter the size as multiples of 1K (that is, 1024) entries or in absolute number of entries. |
| Step 2 | Router(config)# end | Exits to privileged EXEC mode. |

Example 14-1 Limiting the IP-Prefix Region to 2K Entries

```
Router # configure terminal
Router(config)# sdm size ip-prefix k-entries 2
Router(config)# end
```

To display the number of available TCAM entries, enter the **show sdm size** command from global configuration mode:

```
Router # show sdm size
Active Switching Database Region Maximum Sizes :
  IP Adjacency       : 65536 64-bit entries
  IP Prefix           : 204864-bit entries
  QoS Classifiers    : 65536 64-bit entries
  IP VRF Prefix      : 65536 64-bit entries
  IP Multicast       : 65536 64-bit entries
  MAC Addr           : 65536 64-bit entries
  Access List        : 6553664-bit entries
```

Configuring Access Control List Size in TCAM

The default maximum size of the Access Control List (ACL) is 65,536 64-bit entries. You can enter the **sdm access-list** command to limit the TCAM space for ACLs, as shown in [Table 14-2](#).

Table 14-2 Partitioning the TCAM Size for ACLs

| Task | Command |
|--|--|
| sdm access-list <i>number-of-entries</i> | Sets the name of the application region for which you want to configure the size. You can enter the size as an absolute number of entries. |

Example 14-2 Configuring 8,192 Entries for the ACL Region in TCAM

```
Router# configure terminal
Router(config)# sdm access-list 8192
Router(config)# end
```




Configuring Access Control Lists

This chapter describes the access control list (ACL) features built into the ML-Series card.

This chapter contains the following major sections:

- [Understanding ACLs, page 15-1](#)
- [ML-Series ACL Support, page 15-1](#)
- [Modifying ACL TCAM Size, page 15-5](#)

Understanding ACLs

ACLs provide network control and security, allowing you to filter packet flow into or out of ML-Series interfaces. ACLs, which are sometimes called filters, allow you to restrict network use by certain users or devices. ACLs are created for each protocol and are applied on the interface for either inbound or outbound traffic. ACLs do not apply to outbound Control Plane traffic. Only one ACL filter can be applied per direction per subinterface.

When creating ACLs, you define criteria to apply to each packet processed by the ML-Series card; the ML-Series card decides whether to forward or block the packet based on whether or not the packet matches the criteria in your list. Packets that do not match any criteria in your list are automatically blocked by the implicit “deny all traffic” criteria statement at the end of every ACL.

ML-Series ACL Support

Both control-plane and data-plane ACLs are supported on the ML-Series card:

- **Control-plane ACLs:** ACLs used to filter control data that is processed by the CPU of the ML-Series card (for example, distribution of routing information, IGMP joins, and so on).
- **Data-plane ACLs:** ACLs used to filter user data being routed or bridged through the ML Series in hardware (for example, denying access to a host, and so on). These ACLs are applied to an interface in the input or output direction using the **ip access-group** command.

The following apply when using data-plane ACLs on the ML-Series card:

- ACLs are supported on all interface types, including bridged interfaces.
- Reflexive and dynamic ACLs are not supported on the ML-Series card.
- Access violations accounting is not supported on the ML-Series card.
- ACL logging is supported only for packets going to the CPU, not for switched packets.

- IP Standard ACLs applied to bridged egress interfaces are not supported in the data-plane. When bridging, ACLs are only supported on ingress.

IP ACLs

The following ACL styles for IP are supported:

- Standard IP ACLs: These use source addresses for matching operations.
- Extended IP ACLs (control plane only): These use source and destination addresses for matching operations and optional protocol type and port numbers for finer granularity of control.
- Named ACLs: These use source addresses for matching operations.



Note

By default, the end of the ACL contains an implicit deny statement for everything if it did not find a match before reaching the end. With standard ACLs, if you omit the mask from an associated IP host address ACL specification, 0.0.0.0 is assumed to be the mask.

After creating an ACL, you must apply it to an interface, as shown in the [“Applying the ACL to an Interface”](#) section on page 15-4.

Named IP ACLs

You can identify IP ACLs with a name, but it must be an alphanumeric string. Named IP ACLs allow you to configure more IP ACLs in a router than if you used numbered ACLs. If you identify your ACL with an alphabetic rather than a numeric string, the mode and command syntax are slightly different.

Consider the following before configuring named ACLs:

- A standard ACL and an extended ACL cannot have the same name.
- Numbered ACLs are also available, as described in the [“Creating Numbered Standard and Extended IP ACLs”](#) section on page 15-3.

User Guidelines

Keep the following in mind when you configure IP network access control:

- You can program ACL entries into ternary content addressable memory (TCAM).
- You do not have to enter a deny everything statement at the end of your ACL; it is implicit.
- You can enter ACL entries in any order without any performance impact.
- For every eight TCAM entries, the ML-Series card uses one entry for TCAM management purposes.
- Do not set up conditions that result in packets getting lost. This situation can happen when a device or interface is configured to advertise services on a network that has ACLs that deny these packets.
- IP ACLs are not supported for double-tagged (QinQ) packets. They will however be applied to IP packets entering on a QinQ access port.

Creating IP ACLs

The following sections describe how to create numbered standard, extended, and named standard IP ACLs:

- [Creating Numbered Standard and Extended IP ACLs, page 15-3](#)
- [Creating Named Standard IP ACLs, page 15-4](#)
- [Creating Named Extended IP ACLs \(Control Plane Only\), page 15-4](#)
- [Applying the ACL to an Interface, page 15-4](#)

Creating Numbered Standard and Extended IP ACLs

Table 15-1 lists the global configuration commands used to create numbered standard and extended IP ACLs.

Table 15-1 Commands for Numbered Standard and Extended IP ACLs

| Command | Purpose |
|--|--|
| Router(config)# access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>] | Defines a standard IP ACL using a source address and wildcard. |
| Router(config)# access-list <i>access-list-number</i> {deny permit} any | Defines a standard IP ACL using an abbreviation for the source and source mask of 0.0.0.0 255.255.255.255. |
| Router(config)# access-list <i>access-list-number</i> {deny permit} <i>protocol</i> <i>source</i> <i>source-wildcard</i> <i>destination</i> <i>destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] | Defines an extended IP ACL number and the access conditions. |
| Router(config)# access-list <i>access-list-number</i> {deny permit} <i>protocol</i> any any | Defines an extended IP ACL using an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255, and an abbreviation for a destination and destination wildcard of 0.0.0.0 255.255.255.255. |
| Router(config)# access-list <i>access-list-number</i> {deny permit} <i>protocol</i> host <i>source</i> host <i>destination</i> | Defines an extended IP ACL using an abbreviation for a source and source wildcard of source 0.0.0.0, and an abbreviation for a destination and destination wildcard of destination 0.0.0.0. |

Creating Named Standard IP ACLs

To create a named standard IP ACL, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router(config)# ip access-list standard <i>name</i> | Defines a standard IP ACL using an alphabetic name. |
| Step 2 | Router(config-std-nacl)# deny { <i>source</i> [<i>source-wildcard</i>] any } or permit { <i>source</i> [<i>source-wildcard</i>] any } | In access-list configuration mode, specifies one or more conditions as permitted or denied. This determines whether the packet is passed or dropped. |
| Step 3 | Router(config)# exit | Exits access-list configuration mode. |

Creating Named Extended IP ACLs (Control Plane Only)

To create a named extended IP ACL, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|---|
| Step 1 | Router(config)# ip access-list extended <i>name</i> | Defines an extended IP ACL using an alphabetic name. |
| Step 2 | Router(config-ext-nacl)# { deny permit } <i>protocol source source-wildcard destination destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] { deny permit } <i>protocol any any</i> { deny permit } <i>protocol host source host destination</i> | In access-list configuration mode, specifies the conditions allowed or denied. Or: Defines an extended IP ACL using an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255, and an abbreviation for a destination and destination wildcard of 0.0.0.0 255.255.255.255. Or: Defines an extended IP ACL using an abbreviation for a source and source wildcard of <i>source</i> 0.0.0.0, and an abbreviation for a destination and destination wildcard of <i>destination</i> 0.0.0.0. |

Applying the ACL to an Interface

After you create an ACL, you can apply it to one or more interfaces. ACLs can be applied on either the inbound or the outbound direction of an interface. When controlling access to an interface, you can use a name or number. If a standard ACL is applied, the ML-Series card compares the source IP address with the ACL. To apply an ACL to one or more interfaces, use the command in [Table 15-2](#).



Note

IP Standard ACLs applied to the ingress of a bridged virtual interface (BVI) will be applied to all bridged IP traffic in the associated bridge-group, in addition to the BVI Ingress traffic.

Table 15-2 Applying ACL to Interface

| Command | Purpose |
|---|----------------------------------|
| <code>ip access-group {access-list-number name} {in out}</code> | Controls access to an interface. |

Modifying ACL TCAM Size

You can change the TCAM size by entering the `sdm access-list` command. For more information on ACL TCAM sizes, see the [“Configuring Access Control List Size in TCAM”](#) section on page 14-3.



Note

To increase the ACL TCAM size, you must decrease another region’s TCAM size, such as IP, IP multicast, or L2 switching.



Caution

You will need to increase the TCAM size if you see the following error message:

```
Warning:Programming TCAM entries failed
Please remove last ACL command to re-activate ACL operation.
!<ACL number or name> <IP or IPX> <INPUT_ACL or OUTPUT_ACL> from TCAM group for !<interface>
Please see the documentation to see if TCAM space can be
increased on this platform to alleviate the problem.
```

Example 15-1 Monitor and Verify ACLs

```
Router# show ip access-lists 1
Standard IP access list 1
  permit 192.168.1.1
  permit 192.168.1.2
```




Configuring Resilient Packet Ring



Note

The terms "Unidirectional Path Switched Ring" and "UPSR" may appear in Cisco literature. These terms do not refer to using Cisco ONS 15xxx products in a unidirectional path switched ring configuration. Rather, these terms, as well as "Path Protected Mesh Network" and "PPMN," refer generally to Cisco's path protection feature, which may be used in any topological network configuration. Cisco does not recommend using its path protection feature in any particular topological network configuration.

This chapter describes how to configure Resilient Packet Ring (RPR) and Dual RPR Interconnect (DRPRI) for the ML-Series card.

This chapter contains the following major sections:

- [Understanding RPR, page 16-1](#)
- [Configuring Point-to-Point Circuits on CTC for RPR, page 16-4](#)
- [Configuring RPR on IOS, page 16-5](#)
- [Monitoring and Verifying RPR, page 16-10](#)
- [Understanding Dual RPR Interconnect, page 16-10](#)
- [Configuring DRPRI, page 16-12](#)
- [Monitoring and Verifying DRPRI, page 16-16](#)

Understanding RPR

RPR is an emerging network architecture designed for metro fiber ring networks. This new MAC protocol is designed to overcome the limitations of IEEE 802.1D Spanning Tree Protocol (STP), IEEE 802.1W Rapid Spanning Tree Protocol (RSTP) and SONET/SDH in packet-based networks. RPR operates at the Layer 2 level and is compatible with Ethernet and SONET/SDH.

The ML-Series card's RPR relies on the QoS features of the ML-Series card for efficient bandwidth utilization with service level agreement (SLA) support. ML-Series card QoS mechanisms apply to all SONET/SDH traffic on the ML-Series card, whether passed-through, bridged, or stripped.

When an ML-Series card is configured with RPR and made part of a shared packet ring (SPR), the ML-Series card assumes it is part of a ring. If a packet is not destined for devices attached to the specific ML-Series, the ML-Series card simply continues to forward this transit traffic along the SONET/SDH circuit relying on the circular path of the ring architecture to guarantee the packet will eventually arrive

at the destination. This eliminates the need to queue and forward the packet flowing through the non-destination ML-Series card. From a Layer 2 or Layer 3 perspective, the entire RPR looks like one shared network segment.

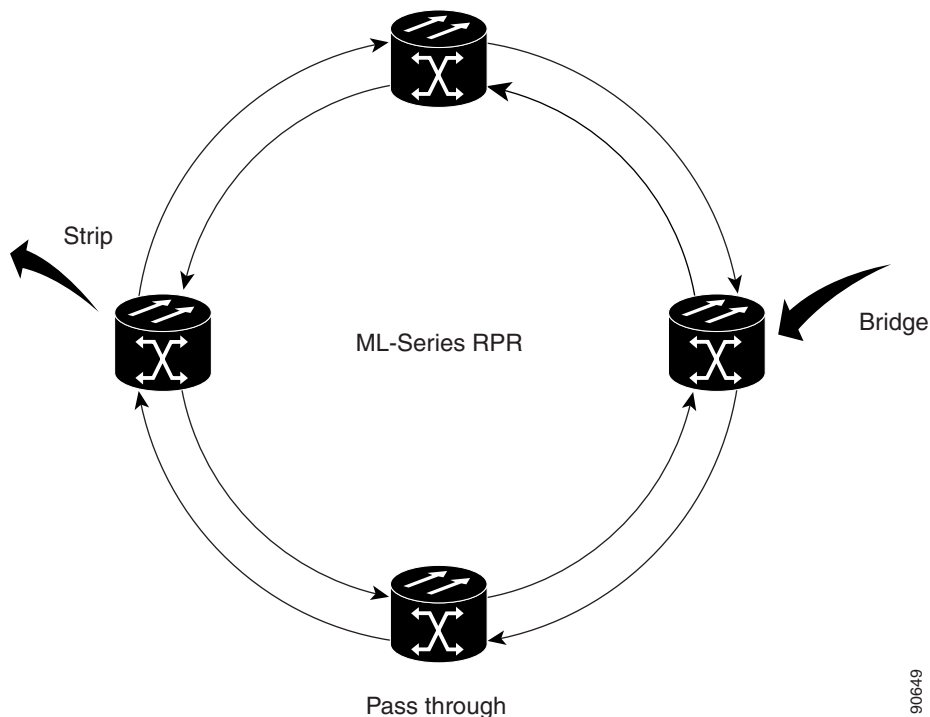
RPR supports operation over protected and unprotected SONET/SDH circuits. On unprotected SONET/SDH circuits RPR provides SONET/SDH-like protection without the redundant SONET/SDH protection path. Eliminating the need for a redundant SONET/SDH path frees bandwidth for additional traffic. RPR also incorporates spatial reuse of bandwidth through a hash algorithm for east/west packet transmission. RPR utilizes the entire ring bandwidth and does not need to block ring segments like STP or RSTP.

Packet Handling Operations

The RPR protocol, using the transmitted packet's header information, allows the interfaces to quickly determine the operation that needs to be applied to the packet. An ML-Series card configured with RPR is part of the ring and has three basic packet-handling operations: bridge, pass-through, or strip.

Figure 16-1 illustrates these operations. Bridging connects and passes packets between the Ethernet ports on the ML-Series and the Packet over SONET/SDH (POS) circuit circling the ring. Pass-through lets the packets continue through the ML-Series card and along the ring, and stripping takes the packet off the ring and discards it. Because STP or RSTP is not in effect between nodes when RPR is configured, the transmitting RPR port strips its own packets after they return from circling the ring. A hash algorithm is used to determine the direction of the packet around the RPR.

Figure 16-1 RPR Packet Handling Operations



90649

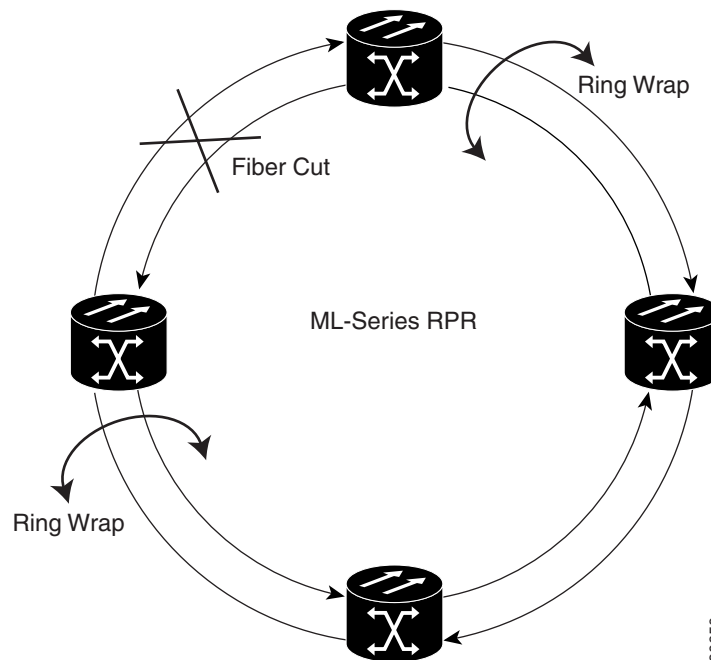
Ring Wrapping

RPR initiates ring wraps in the event of a fiber cut, node failure, node restoration, new node insertion, or other traffic problems. This protection mechanism redirects traffic to the original destination by sending it in the opposite direction around the ring after a link state change or after receiving SONET/SDH path level alarms. Ring wrapping on the ML-Series card allows sub 50-msec convergence times. RPR convergence times are comparable to SONET/SDH and much faster than STP or RSTP.

RPR on the ML-Series card survives both unidirectional and bidirectional transmission failures within the ring. Unlike STP or RSTP, RPR restoration is scalable, increasing the number of ML-Series cards in a ring does not increase the convergence time.

RPR will initiate ring wraps immediately (default) or delay the wrap with a configured carrier delay time. When configured to wrap traffic after the carrier delay, a POS trigger delay time should be added to the carrier delay time to estimate approximate convergence times. The default and minimum POS trigger delay time for the ML-Series Card is 200 ms. A carrier delay time of 200 ms (default) and a POS trigger delay time of 200 ms (default and minimum) combine for a total convergence time of approximately 400 ms. If the carrier delay is set to 0, then the convergence time would be approximately 200 ms. [Figure 16-2](#) illustrates ring wrapping.

Figure 16-2 RPR Ring Wrapping



Note

If the carrier delay time is changed from the default, the new carrier delay time must be configured on all the ML-Series card interfaces, including the SPR, POS, and Gigabit Ethernet or FastEthernet interfaces.

**Note**

ML-Series card POS interfaces normally send PDI-P to the far-end when the POS link goes down or RPR wraps. ML-Series card POS interfaces do not send PDI-P to the far-end when PDI-P is detected, when RDI-P is being sent to the far-end or when the only defects detected are GFP LFD, GFP CSF, VCAT LOM or VCAT SQM.

MAC Address and VLAN Support

RPR improves MAC address support, because an ML-Series card does not need to learn the MAC address of pass-through packets. The ML-Series card's MAC address table only holds the MAC IDs of packets that have been bridged or stripped by that card. This allows the collective tables of the ML-Series cards in the ring to hold a greater number of MAC addresses.

RPR also enhances VLAN support relative to STP and RSTP. In an STP and RSTP, a new VLAN must be configured on all POS interfaces on the ring. In RPR, the VLAN must only be added to the configuration of those interfaces that bridge or strip packets for that VLAN. The ML-Series card still has a 255 architectural maximum limit of VLAN/bridge-group per ML-Series card. But because the ML-Series card only needs to hold the VLANs incorporating that card, the collective number of VLANs held by all the ML-Series cards in the ring can be much greater.

Configuring Point-to-Point Circuits on CTC for RPR

RPR on the Cisco ONS 15454 enables two or more ML-Series cards to become one functional network segment or SPR. The bridged ML-Series cards are connected to each other through point-to-point STS circuits, which use one of the first ML-Series card's POS ports as a source and one of the second ML-Series card's POS ports as a destination. All ML-Series cards in an SPR must be connected directly or indirectly by point-to-point circuits.

The point-to-point circuits use the ONS 15454 SONET/SDH network. Provision the point-to-point circuits using CTC or TL1 in the same manner as an ONS 15454 OC-N card STS circuits. The *ONS 15454 Procedure Guide* or *Cisco ONS 15454 SDH Procedure Guide* provides specific instructions on how to create an automatically routed optical circuit.

When configuring a point-to-point circuit on the ML-Series:

- Leave all CTC Circuit Creation Wizard options at default, except **Fully Protected Path** on the Circuit Routing Preferences dialog, which provides SONET/SDH protection and should be unchecked. RPR provides Layer 2 protection for SPR circuits.
- Check the option **Using Required Nodes and Spans** to route automatically on the Circuit Routing Preferences dialog. If the source and destination nodes are adjacent on the ring, exclude all nodes except the source and destination on the Circuit Routing Preferences dialog. This forces the circuit to be routed directly between source and destination and preserves STS circuits, which would be consumed if the circuit routed through other nodes in the ring. If there is a node or nodes that do not contain an ML-Series card between the two nodes containing ML-Series card, include this node or nodes in the included nodes in the Circuit Routing Preference dialog, along with the source and destination nodes.
- Keep in mind that ML-Series card STS circuits do not support unrelated circuit creation options, such as unidirectional traffic, creating cross-connects only (TL1-like), interdomain (UCP), protected drops, or path protection path selectors.

Once the CTC circuit process is complete, begin an IOS session to configure RPR/SPR on the ML-Series card and interfaces.

**Note**

A best practice is to configure SONET/SDH circuits from port 0 (east) to port 1 (west) or port 1 (east) to port 0 (west) around the SONET/SDH ring. This setup is also required for Cisco CTM network management software to recognize the ML-Series configuration as an SPR.

Configuring RPR on IOS

You configure RPR on the ML-Series cards by creating a shared packet ring (SPR) interface from the Cisco IOS CLI. The SPR is a virtual interface, similar to an EtherChannel interface. The POS interfaces are the physical interfaces associated with the RPR SPR interface. An ML-Series card supports a single SPR interface. The SPR interface has a single MAC address and provides all the normal attributes of a Cisco IOS interface, such as support for default routes. An SPR interface is considered a trunk port, and like all trunk ports, subinterfaces must be configured for the SPR interface to become part of a bridge group.

An SPR interface is configured similarly to a EtherChannel (port-channel) interface. The members of the SPR interface must be POS interfaces. Instead of using the **channel-group** command to define the members, you use the **spr-intf-ID** command. And like port-channel, you configure the SPR interfaces instead of the POS interface.

**Caution**

In configuring an SPR, if one ML-Series card is not configured with an SPR interface, but valid STS/STM circuits connect this ML-Series card to the other ML-Series cards in the SPR, no traffic will flood between the properly configured ML-Series cards in the SPR, and no alarms will indicate this condition. Cisco recommends configuring all of the ML-Series cards in an SPR before sending traffic.

**Note**

RPR is only supported with LEX encapsulation. LEX is the default encapsulation for the ML-Series.

To provision RPR, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|---------------|---|--|
| Step 1 | <code>Router(config)#bridge irb</code> | Enables the Cisco IOS software to both route and bridge a given protocol on separate interfaces within a single ML-Series card. |
| Step 2 | <code>Router(config)#interface spr 1</code> | Creates the SPR interface on the ML-Series card or enters the SPR interface configuration mode. The only valid SPR number is 1. |
| Step 3 | <code>Router(config-if)#spr station-id <i>station-ID-number</i></code> | Configures a station ID. The user must configure a different number for each SPR interface that attaches to the RPR. Valid station ID numbers range from 1 to 254. |

| Step | Command | Purpose |
|---------|---|--|
| Step 4 | Router(config-if)# spr wrap {immediate delayed} | (Optional) Sets the RPR wrap mode to either wrap traffic the instant it detects a link state change or to wrap traffic after the carrier delay, which gives the SONET/SDH protection time to register the defect and declare the link down. Immediate should be used if RPR is running over unprotected SONET/SDH circuits. Delayed should be run for BLSR or path protection protected circuits. The default setting is immediate. |
| Step 5 | Router(config-if)# pos trigger defect ber_sd_b3 | Triggers an RPR wrap when bit errors in the SONET/SDH signal cause a signal degrade condition. This command also increases awareness of the signal degrade by raising a TPTFAIL alarm against the POS interface. |
| Step 6 | Router(config-if)# bridge-group <i>bridge-group-number</i> | (Optional) Assigns the SPR interface to a bridge-group. This bridge group bridges the SPR and desired FastEthernet or Gigabit Ethernet interfaces. |
| Step 7 | Router(config-if)# carrier delay msec <i>milliseconds</i> | (Optional) Sets the carrier delay time. The default setting is 200 milliseconds, which is optimum for SONET/SDH protected circuits. The default unit of time for setting the carrier delay is seconds. The msec command resets the time unit to milliseconds. Note If the carrier delay time is changed from the default, the new carrier delay time must be configured on all the ML-Series card interfaces, including the SPR, POS, and Gigabit Ethernet or FastEthernet interfaces. |
| Step 8 | Router(config) interface pos 0 | (Optional) Enters interface configuration mode for POS port 0 to set carrier delay time. |
| Step 9 | Router(config-if)# carrier delay msec <i>milliseconds</i> | (Optional) Sets the carrier delay time. The default setting is 200-msec, which is optimum for SONET/SDH protected circuits. The default unit of time for setting the carrier delay is seconds. The msec command resets the time unit to milliseconds. |
| Step 10 | Router(config) interface pos 1 | (Optional) Enters interface configuration mode for POS port 1 to set optional commands. |
| Step 11 | Router(config-if)# carrier delay msec <i>milliseconds</i> | (Optional) Sets the carrier delay time. The default setting is 200 milliseconds, which is optimum for SONET/SDH protected circuits. The default unit of time for setting the carrier delay is seconds. The msec command resets the time unit to milliseconds. |

| | Command | Purpose |
|---------|---|--|
| Step 12 | Router(config-if)# end | Exits to privileged EXEC mode. |
| Step 13 | Router# copy running-config startup-config | (Optional) Saves configuration changes to NVRAM. |

Each of the ML-Series card's two POS ports must be assigned to the SPR interface. The ML-Series STS circuit must have first been provisioned through CTC for the virtual POS ports to appear in Cisco IOS.

**Caution**

The SPR interface is the routed interface. Do not enable Layer 3 addresses or assign bridge groups on the POS interfaces assigned to the SPR interface.

**Caution**

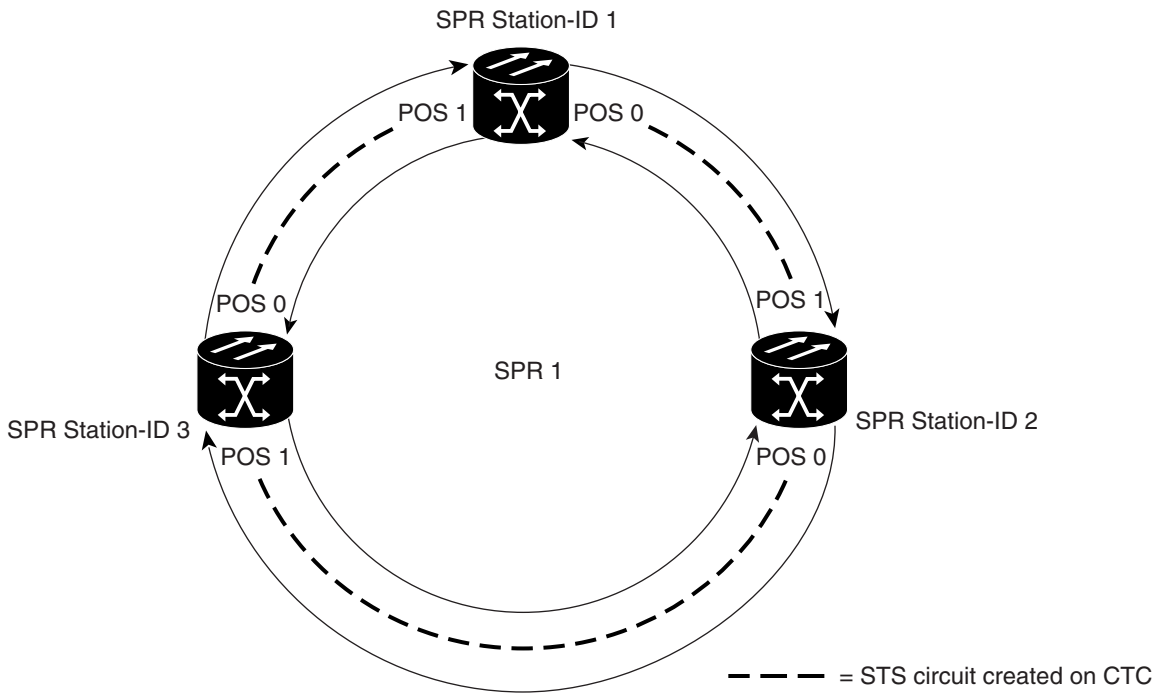
When traffic coming in on an SPR interface needs to be policed, the same input service policy needs to be applied to both the POS ports that are part of the SPR interface.

To assign a POS interface on the ML-Series to the SPR, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|---|--|
| Step 1 | Router(config)# interface pos <i>number</i> | Enters the interface configuration mode to configure the first POS interface that you want to assign to the SPR. |
| Step 2 | Router(config-if)# spr-intf-ID <i>shared-packet-ring-number</i> | Assigns the POS interface to the SPR interface. The shared packet ring number must be the same shared packet ring number that you assigned to the SPR interface. |
| Step 3 | Router(config-if)# interface pos <i>number</i> | Enters the interface configuration mode to configure the second POS interface that you want to assign to the SPR. |
| Step 4 | Router(config-if)# spr-intf-ID <i>shared-packet-ring-number</i> | Assigns the POS interface to the SPR interface. The shared packet ring number must be the same shared packet ring number that you assigned to the SPR interface. |
| Step 5 | Router(config-if)# end | Exits to privileged EXEC mode. |
| Step 6 | Router# copy running-config startup-config | (Optional) Saves the configuration changes to NVRAM. |

RPR IOS Configuration

The associated code is provided in the sections that follow the figure. The configuration assumes that ML-Series card POS ports are already linked by point-to-point SONET/SDH circuits configured through CTC.

Example 16-1 RPR Configuration Example

98606

Example 16-2 SPR Station-ID 1 Configuration

```

bridge irb
!
interface SPR1
no ip address
no keepalive
spr station-ID 1
hold-queue 150 in
bridge-group 1
!
interface POS0
no ip address
spr-intf-ID 1
!
interface POS1
no ip address
spr-intf-ID 1

interface GigabitEthernet0
no ip address
no ip route-cache
bridge-group 1

interface GigabitEthernet1
no ip address
no ip route-cache
bridge-group 1

```

Example 16-3 SPR Station-ID 2 Configuration

```
bridge irb
!
interface SPR1
no ip address
no keepalive
spr station-ID 2
hold-queue 150 in
bridge-group 1
!
interface POS0
no ip address
spr-intf-ID 1
!
interface POS1
no ip address
spr-intf-ID 1

interface GigabitEthernet0
no ip address
no ip route-cache
bridge-group 1

interface GigabitEthernet1
no ip address
no ip route-cache
bridge-group 1
```

Example 16-4 SPR Station-ID 3 Configuration

```
bridge irb
!
interface SPR1
no ip address
no keepalive
spr station-ID 3
hold-queue 150 in
bridge-group 1
!
interface POS0
no ip address
spr-intf-ID 1
!
interface POS1
no ip address
spr-intf-ID 1

interface GigabitEthernet0
no ip address
no ip route-cache
bridge-group 1

interface GigabitEthernet1
no ip address
no ip route-cache
bridge-group 1
```

Monitoring and Verifying RPR

After RPR is configured, you can monitor its status using the **show interface spr** or **show run interface spr** command.

Example 16-5 Monitor and Verify RPR

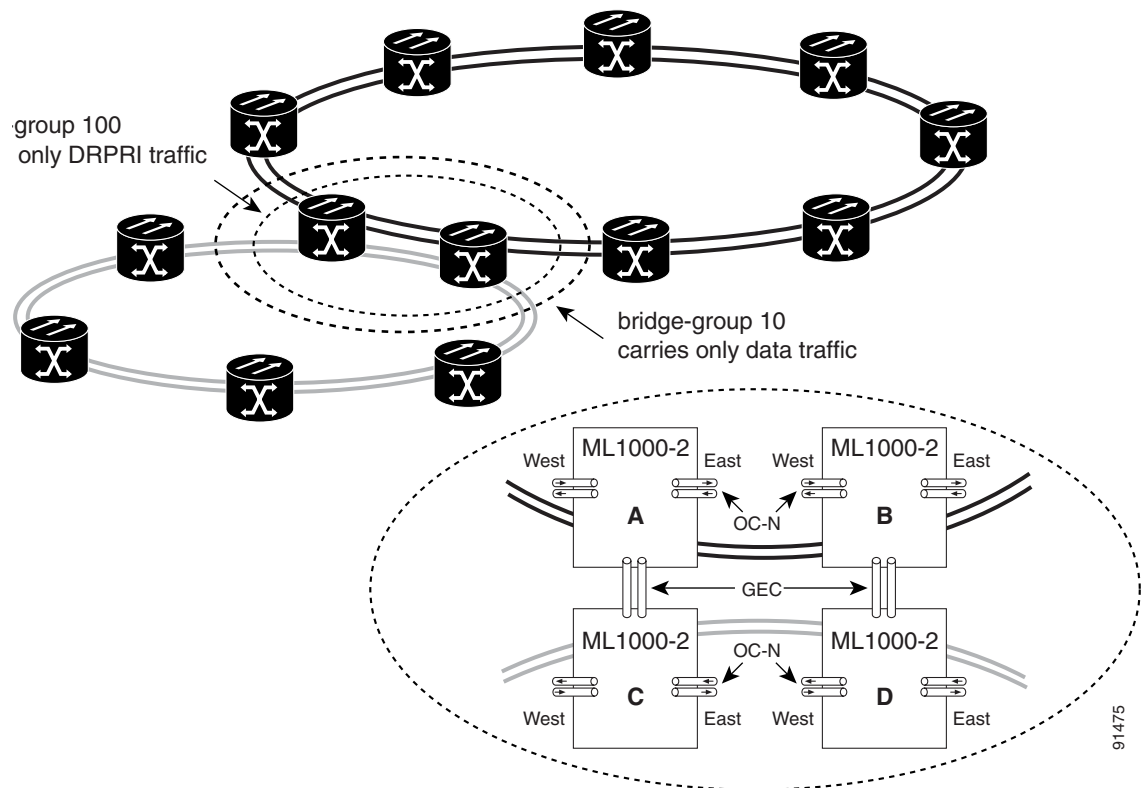
```
Router# show interfaces spr 1
SPR1 is up, line protocol is up
Hardware is POS-SPR, address is 0005.9a39.714a (bia 0000.0000.0000)
MTU 1500 bytes, BW 1244160 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ONS15454-G1000, loopback not set
Keepalive not set
DTR is pulsed for 33391 seconds on reset
ARP type: ARPA, ARP Timeout 04:00:00
    No. of active members in this SPR interface: 2
        Member 0 : POS0
        Member 1 : POS1
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/150/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/80 (size/max)
5 minute input rate 1000 bits/sec, 2 packets/sec
5 minute output rate 2000 bits/sec, 4 packets/sec
    1014 packets input, 96950 bytes
    Received 0 broadcasts (0 IP multicast)
    0 runts, 0 giants, 0 throttles
    0 parity
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 input packets with dribble condition detected
1640 packets output, 158832 bytes, 0 underruns
0 output errors, 0 applique, 9 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
```

Understanding Dual RPR Interconnect

Cisco ML-Series RPR includes a mechanism to interconnect rings for protection from node failure. The bridge-group protocol, DRPRI, provides two parallel connections of the rings linked by a special instance of RSTP. One connection is the active node and the other is the standby node. During a failure of the active node, link, or card, a proprietary algorithm detects the failure and causes a switchover to the standby node.

The paired ML1000-2 cards share the same station ID and are viewed by other members of the RPR as a single card. In [Figure 16-3](#), paired cards A and B have the same SPR station ID, and paired cards C and D have the same station ID. The interconnected nodes do not need to be adjacent on the RPR. Bridging, IP routing, IP Multicast, policing and bandwidth allocations can still be provisioned on DRPRI ML1000-2 cards.

Figure 16-3 Dual RPR Interconnect Network and Paired Cards



DRPRI has these characteristics:

- Four ML1000-2 cards are required
- All four ML1000-2 cards must be part of the same bridge-group (VLAN.)
- Each paired set of ML1000-2 cards must have the same SPR station ID.
- The bridge-group must be configured on SPR subinterfaces.
- The DRPRI bridge-group is limited to one protocol, so a bridge-group with DRPRI implemented cannot also implement RSTP or STP.
- On each of the four ML1000-2 cards, both Gigabit Ethernet ports must be joined in Gigabit EtherChannel (GEC) and the GEC interface included in the DRPRI bridge-group, or one Gigabit Ethernet port must be shut down and the other one included in the DRPRI bridge-group. We recommend the GEC method.
- The DRPRI bridge-group cannot also be used to carry data traffic.
- A DRPRI node can only be used for interconnecting two RPRs. The front ports of the cards should not be used to carry other traffic.
- Non-DRPRI bridge-groups carrying traffic between rings should not have STP or RSTP configured.
- Non-DRPRI bridge-groups carrying traffic between rings must be configured on each of the four ML-Series cards.
- Q-in-Q and protocol tunnels cannot be started on DRPRI nodes, but DRPRI nodes can bridge Q-in-Q and protocol tunnels across the connected rings.

Configuring DRPRI

DRPRI requires two pairs of ML-Series cards with one pair configured as RPR and belonging to the first of two adjacent RPRs, and the second pair configured as RPR and belonging to the second RPR (Figure 16-3). DRPRI is configured on each of the four ML1000-2 cards that connect the two adjacent RPRs. The process of configuring DRPRI consists of the following tasks:

1. Configure a bridge-group with the DRPRI protocol.
2. Configure the SPR interface.
 - a. Assign a station ID number.
 - b. Assign a DRPRI ID of 0 or 1.
3. Create an SPR subinterface and assign the bridge-group to the subinterface.
4. Create a GEC interface.
5. Create a GEC subinterface and assign the bridge-group to the subinterface.

To enable and configure DRPRI, perform the following procedure, beginning in global configuration mode:

| | Command | Purpose |
|--------|--|--|
| Step 1 | <code>Router(config)#bridge irb</code> | Enables the Cisco IOS software to both route and bridge a given protocol on separate interfaces within a single ML-Series card. |
| Step 2 | <code>Router(config)#bridge <i>bridge-group-number</i> protocol drpri-rstp</code> | Creates the bridge-group number shared by the four ML1000-2 cards and assigns the protocol for DRPRI to the bridge-group. The same command using the same bridge group number must be given on each of the four cards. |
| Step 3 | <code>Router(config)#interface spr 1</code> | Creates the SPR interface for RPR or enters the SPR interface configuration mode on a previously created SPR interface. The only valid SPR number is 1. |
| Step 4 | <code>Router(config-if)#spr station-ID <i>station-ID-number</i></code> | Configures a station identification number. The user must configure the same station ID on both the paired cards. Valid station ID numbers range from 1 to 254. |
| Step 5 | <code>Router(config-if)#spr drpri-ID {0 1}</code> | Creates a DRPRI identification number of 0 or 1 to differentiate between the ML1000-2 cards paired for DRPRI. |
| Step 6 | <code>Router(config-if)#interface spr <i>shared-packet-ring-sub-interface-number</i></code> | Creates the SPR subinterface. |
| Step 7 | <code>Router(config-subif)#encapsulation dot1q <i>vlan-ID</i></code> | Sets the SPR subinterface encapsulation to IEEE 802.1Q. |
| Step 8 | <code>Router(config-subif)#bridge-group <i>bridge-group-number</i></code> | Assigns the SPR subinterface to a bridge-group. |
| Step 9 | <code>Router(config)# interface port-channel <i>channel-number</i></code> | Creates the Gigabit EtherChannel (GEC) interface or channel-group. |

| | Command | Purpose |
|---------|---|--|
| Step 10 | Router(config-if)# interface gigabitethernet <i>number</i> | Enters interface configuration mode for the first Gigabit Ethernet interface that you want to assign to the GEC subinterface. |
| Step 11 | Router(config-if)# channel-group <i>channel-number</i> | Assigns the Gigabit Ethernet interfaces to the GEC. The channel number must be the same channel number you assigned to the EtherChannel interface. |
| Step 12 | Router(config-if)# interface gigabitethernet <i>number</i> | Enters interface configuration mode for the second Gigabit Ethernet interface that you want to assign to the GEC subinterface. |
| Step 13 | Router(config-if)# channel-group <i>channel-number</i> | Assigns the Gigabit Ethernet interfaces to the GEC. The channel number must be the same channel number you assigned to the EtherChannel interface. |
| Step 14 | Router(config-subif)# interface port-channel <i>channel-sub-interface-number</i> | Creates the Gigabit EtherChannel (GEC) subinterface. |
| Step 15 | Router(config-subif)# encapsulation dot1q <i>vlan-ID</i> | Sets subinterface encapsulation to IEEE 802.1Q. The VLAN ID used should be the same VLAN ID used in step 7. |
| Step 16 | Router(config-subif)# bridge-group <i>bridge-group-number</i> | Assigns the GEC subinterface to the bridge-group. |
| Step 17 | Router(config-if)# end | Exits to privileged EXEC mode. |
| Step 18 | Router# copy running-config startup-config | (Optional) Saves configuration changes to NVRAM. |

DRPRI IOS Configuration

Figure 16-3 on page 16-11 shows an example of RPR configuration. The associated code is provided in the following section.

Example 16-6 ML-Series A Configuration

```
hostname ML-Series A
bridge irb
bridge 100 protocol drpri-rstp

interface Port-channell
no ip address
no ip route-cache
hold-queue 300 in

interface Port-channell.1
encapsulation dot1Q 10
no ip route-cache
bridge-group 100

interface SPR1
no ip address
no keepalive
spr station-ID 1
```

```

hold-queue 150 in

interface SPR1.1
  encapsulation dot1Q 10
  bridge-group 100

interface GigabitEthernet0
  no ip address
  no ip route-cache
  channel-group 1

interface GigabitEthernet1
  no ip address
  no ip route-cache
  channel-group 1

interface POS0
  no ip address
  spr-intf-ID 1
  crc 32

interface POS1
  no ip address
  spr-intf-ID 1
  crc 32

ip classless
no ip http server

```

Example 16-7 ML-Series B Configuration

```

hostname ML-Series B
bridge irb
bridge 100 protocol drpri-rstp

interface Port-channel1
  no ip address
  no ip route-cache
  hold-queue 300 in

interface Port-channel1.1
  encapsulation dot1Q 10
  no ip route-cache
  bridge-group 100

interface SPR1
  no ip address
  no keepalive
  spr station-ID 1
  spr drpr-ID 1
  hold-queue 150 in

interface SPR1.1
  encapsulation dot1Q 10
  bridge-group 100

interface GigabitEthernet0
  no ip address
  no ip route-cache
  channel-group 1

interface GigabitEthernet1
  no ip address

```

```
no ip route-cache
channel-group 1

interface POS0
no ip address
spr-intf-ID 1
crc 32

interface POS1
no ip address
spr-intf-ID 1
crc 32

ip classless
no ip http server
```

Example 16-8 ML-Series C Configuration

```
hostname ML-Series C
bridge irb
bridge 100 protocol drpri-rstp

interface Port-channel1
no ip address
no ip route-cache
hold-queue 300 in

interface Port-channel1.1
encapsulation dot1Q 10
no ip route-cache
bridge-group 100

interface SPR1
no ip address
no keepalive
spr station-ID 2
hold-queue 150 in

interface SPR1.1
encapsulation dot1Q 10
bridge-group 100

interface GigabitEthernet0
no ip address
no ip route-cache
channel-group 1

interface GigabitEthernet1
no ip address
no ip route-cache
channel-group 1

interface POS0
no ip address
spr-intf-ID 1
crc 32

interface POS1
no ip address
spr-intf-ID 1
crc 32
```

```
ip classless
no ip http server
```

Example 16-9 ML-Series D Configuration

```
hostname ML-Series D
bridge irb
bridge 100 protocol drpri-rstp

interface Port-channel1
no ip address
no ip route-cache
hold-queue 300 in

interface Port-channel1.1
encapsulation dot1Q 10
no ip route-cache
bridge-group 100

interface SPR1
no ip address
no keepalive
spr station-ID 2
spr drpr-ID 1
hold-queue 150 in

interface SPR1.1
encapsulation dot1Q 10
bridge-group 100

interface GigabitEthernet0
no ip address
no ip route-cache
channel-group 1

interface GigabitEthernet1
no ip address
no ip route-cache
channel-group 1

interface POS0
no ip address
spr-intf-ID 1
crc 32

interface POS1
no ip address
spr-intf-ID 1
crc 32

ip classless
no ip http server
```

Monitoring and Verifying DRPRI

After DRPRI is configured, you can monitor its status using the **show bridge verbose** command.

Example 16-10 show bridge verbose command

```
Router#show bridge bridge-group-number verbose
```



Command Reference



Note

The terms "Unidirectional Path Switched Ring" and "UPSR" may appear in Cisco literature. These terms do not refer to using Cisco ONS 15xxx products in a unidirectional path switched ring configuration. Rather, these terms, as well as "Path Protected Mesh Network" and "PPMN," refer generally to Cisco's path protection feature, which may be used in any topological network configuration. Cisco does not recommend using its path protection feature in any particular topological network configuration.

This appendix provides a command reference for those Cisco IOS commands or those aspects of Cisco IOS commands unique to ML-Series cards. For information about the standard IOS Release 12.1 commands, refer to the IOS documentation set available from the Cisco.com home page. Use the Select an Area pull-down menu to select **Products and Services > Technical Documentation**. On the Cisco Product Documentation home page, select **Release 12.1** from the Cisco IOS Software drop-down list.

■ [no] bridge < bridge-group-number > protocol {drpri-rstp | ieee | rstp}

[no] bridge < *bridge-group-number* > protocol {drpri-rstp | ieee | rstp}

To define the protocol employed by a bridge-group use the bridge protocol global configuration command. If no protocol will be employed by the bridge-group, this command is not needed. To remove a protocol from the bridge group, use the no form of this command with the appropriate keywords and arguments.

| Parameter | Description |
|----------------------------|---|
| drpri-rstp | The protocol that enables the Dual RPR Interconnect (DRPRI) feature of the ML-Series cards. |
| ieee | IEEE 802.1D Spanning Tree Protocol |
| rstp | IEEE 802.1W Rapid Spanning Tree Protocol |
| <i>bridge-group-number</i> | The identifying number of the bridge group being assigned a protocol. |

Defaults

N/A

Command Modes

Global configuration

Usage Guidelines

The protocol DRPRI-RSTP is only employed when configuring ML-Series cards as part of a DRPRI. For more information, see [Configuring DRPRI, page 16-12](#). A bridge group with DRPRI is limited to one protocol, so the bridge group cannot also implement RSTP or STP.

Examples

The following example assigns the DRPRI protocol to the bridge group with the bridge group number of 100.

```
Router(config)# bridge 100 protocol drpri-rstp
```


[no] clock auto

Use the clock auto command to determine whether the system clock parameters are configured automatically from the TCC+/TCC2. When enabled both summertime and timezone are automatically configured, and the system clock is periodically synchronized to the TCC+/TCC2. Use the no form of the command to disable this feature.

Syntax Description This command has no arguments or keywords.

Defaults The default setting is clock auto.

Command Modes Global configuration

Usage Guidelines The no form of the command is required before any manual configuration of summertime, timezone, or clock. The no form of the command is required if Network Time Protocol (NTP) is configured in Cisco IOS. The ONS 15454 SONET/SDH is also configured through CTC to use a NTP or Simple Network Time Protocol (SNTP) server to set the date and time of the node.

Examples Router(config)#no clock auto

Related Commands

- clock summertime
- clock timezone
- clock set

interface spr 1

Use this command to create a shared packet ring (spr) interface on an ML-Series card for RPR. If the interface has already been created, this command enters spr interface configuration mode. The only valid spr interface number is 1.

Defaults

N/A

Command Modes

Global configuration

Usage Guidelines

The command allows the user to create a virtual interface for the RPR/SPR. Commands, such as **spr wrap** or **spr station-id**, can then be applied to the RPR through SPR configuration command mode.

Examples

The following example creates the shared packet ring interface:

```
Router(config)# interface spr 1
```

Related Commands**spr drpri-id****spr-intf-id****spr station-id****spr wrap**

[no] pos flag c2 <value>

Use this command to specify the C2 byte value for transmitted and received frames. Use the no form of the command to return the C2 byte to its default value.

| Parameter | Description |
|--------------|---------------|
| <i>value</i> | C2 byte value |

Defaults

When changing the encapsulation on a POS port between LEX and PPP/HDLC, the scrambling and c2 settings will be automatically changed to their default values according to the table below.

| encap | scrambling | c2 |
|----------|---------------------|------------------|
| LEX | pos scramble-spe | pos flag c2 0x01 |
| PPP/HDLC | no pos scramble-spe | pos flag c2 0xCF |

In PPP/HDLC encapsulation, changing the scrambling, automatically changes the “pos flag c2” to its default according to the table below. In LEX encapsulation, changing the scrambling does not affect c2.

| encap | scrambling | c2 |
|----------|---------------------|------------------|
| PPP/HDLC | pos scramble-spe | pos flag c2 0xCF |
| PPP/HDLC | no pos scramble-spe | pos flag c2 0x16 |

Command Modes

Interface configuration mode (POS only)

Usage Guidelines

This value is normally configured to match the setting on the peer Path Terminating Equipment (PTE). Using the correct order of operations will avoid having the non-default settings overridden by the encapsulation change. The recommended order follows:

- Set encap to PPP/HDLC
- Set scrambling (if a non-default setting is required)
- Set c2 (if a non-default setting is required)

Also note that the crc setting varies among different types of PTE. The default crc on the ML series card is 32-bits, regardless of any other settings. In most circumstances, the default settings should be correct, but users need to verify this with the user documentation for the PTE.

Examples

```
Gateway(config)#int pos0
Gateway(config-if)#pos flag c2 0x16
```

Related Commands

pos trigger defects
pos report

[no] pos report *alarm*

Use this command to specify which alarms/signals are logged to the console. This command has no effect on whether alarms are reported to the TCC2/TCC2P and CTC. These conditions are soaked and cleared per Telcordia GR-253. Use the no form of the command to disable reporting of a specific alarm/signal.

| Syntax Description | Parameter | Description |
|--------------------|--------------|---|
| | <i>alarm</i> | The SONET/SDH alarm that is logged to the console. The alarms are as follows: all —All link down alarm failures ber_sd_b3 —PBIP BER in excess of SD threshold failure ber_sf_b3 —PBIP BER in excess of SD threshold failure encap —Path Signal Label Encapsulation Mismatch failure pais —Path Alarm Indication Signal failure plop —Path Loss of Pointer failure ppdi —Path Payload Defect Indication failure pplm —Payload label mismatch path prdi —Path Remote Defect Indication failure ptim —Path Trace Indicator Mismatch failure puneq —Path Label Equivalent to Zero failure |

Defaults The default is to report all alarms.

Command Modes Interface configuration mode (POS only)

Usage Guidelines This value is normally configured to match the setting on the peer PTE.

Examples

```
Gateway(config)# int pos0
Gateway(config-if)# pos report all
Gateway(config-if)# pos flag c2 1
03:16:51: %SONET-4-ALARM: POS0: PPLM
Gateway(config-if)# pos flag c2 0x16
03:17:34: %SONET-4-ALARM: POS0: PPLM cleared
```

Related Commands pos trigger defects

[non] pos trigger defects *condition*

Use this command to specify which conditions cause the associated POS link state to change. These conditions are soaked/cleared using the delay specified in the **pos trigger delay** command. Use the no form of the command to disable triggering on a specific condition.

| Syntax Description | Parameter | Description |
|--------------------|------------------|--|
| | <i>condition</i> | <p>The SONET/SDH condition that causes the link state change. The conditions are as follows:</p> <ul style="list-style-type: none"> all—All link down alarm failures ber_sd_b3—PBIP BER in excess of SD threshold failure ber_sf_b3—PBIP BER in excess of SD threshold failure (default) encap—Path Signal Label Encapsulation Mismatch failure (default) pais—Path Alarm Indication Signal failure (default) plop—Path Loss of Pointer failure (default) ppdi—Path Payload Defect Indication failure (default) pplm—Payload label mismatch path (default) prdi—Path Remote Defect Indication failure (default) ptim—Path Trace Indicator Mismatch failure (default) puneq—Path Label Equivalent to Zero failure (default) |

Defaults See list in above description.

Command Modes Interface configuration mode (POS only)

Usage Guidelines This value is normally configured to match the setting on the peer PTE.

Examples

```
Gateway(config)# int pos0
Gateway(config-if)# pos trigger defects all
```

Related Commands pos trigger delay

[no] pos trigger delay <time>

Use this command to specify which conditions cause the associated POS link state to go change. The conditions specified in the **pos trigger defects** command are soaked/cleared using this delay. Use the no form of the command to use the default value.

| Parameter | Description |
|-----------|---|
| time | delay time in milliseconds, 200 to 2000 |

Defaults

The default value is 200 milliseconds.

Command Modes

Interface configuration mode (POS only)

Usage Guidelines

This value is normally configured to match the setting on the peer Path Terminating Equipment (PTE). The time granularity for this command is 50 milliseconds.

Examples

```
Gateway(config)#int pos0
Gateway(config-if)#pos trigger delay 500
```

Related Commands

pos trigger defects

[no] pos scramble-spe

Use this command to enable scrambling.

Syntax Description This command has no arguments or keywords.

Defaults The default value depends on the encapsulation.

| encap | scrambling |
|----------|---------------------|
| LEX | pos scramble-spe |
| PPP/HDLC | no pos scramble-spe |

Command Modes Interface configuration mode (POS only)

Usage Guidelines This value is normally configured to match the setting on the peer Path Terminating Equipment (PTE). This command may change the pos flag c2 configuration.

Examples

```
Gateway(config)#int pos0
Gateway(config-if)#pos scramble-spe
```

Related Commands pos flag c2

show controllers pos <interface-number> [details]

Use this command to display the status of the POS controller. Use the details argument to obtain certain additional information as described below.

| Parameter | Description |
|------------------|-----------------------------------|
| interface-number | Number of the POS interface <0-1> |

Defaults

N/A

Command Modes

Privileged EXEC

Usage Guidelines

This command may be used to help diagnose and isolate POS or SONET problems.

Examples

```
Gateway#show controllers pos0 details
Interface POS0
Hardware is Packet/Ethernet over Sonet
PATH
  PAIS      = 0          PLOP      = 0          PRDI      = 0          PTIM      = 0
  PPLM      = 3          PUNEQ     = 0          PPDJ      = 0
  BER_SF_B3 = 0          BER_SD_B3 = 0          BIP(B3)   = 0          REI       = 15
  NEWPTR    = 1          PSE       = 0          NSE       = 0

Active Alarms : None
Demoted Alarms: None
Active Defects: None
Alarms reportable to CLI: PAIS PRDI PLOP PUNEQ PPLM PTIM PPDJ BER_SF_B3 BER_SD_B3
Link state change defects: PAIS PLOP PRDI BER_SF_B3
Link state change time   : 500 (msec)

DOS FPGA channel number: 0
Starting STS (0 based) : 0
Circuit size           : STS-3c
RDI Mode                : 1 bit
C2 (tx / rx)           : 0x16 / 0x16
Framing                 : SONET

Path Trace
Mode                    : off
Buffer                  : Unstable
Remote hostname         :
Remote interface        :
Remote IP addr          :

B3 BER thresholds:
SFBER = 1e-5,   SDBER = 1e-7
Xmt Str:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
Exp Str:
```



```

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
Rcv Str:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

B3 BER thresholds:
BER TH: SFBER=1e-5, SDBER=1e-7,
BER TH: Cur SFBER=0, Cur SDBER=0, berMap=0x00,
BER TH: BER 1e-3, BIP Sum=0, setTh=7404, clrTh=0
WIND BER TH: SetCross=0x0003, setTh=3630
Counts= 0, 0,
BER TH: BER 1e-4, BIP Sum=0, setTh=2637, clrTh=2931
WIND BER TH: SetCross=0x0003, setTh=1266
Counts= 0, 0,
BER TH: BER 1e-5, BIP Sum=0, setTh=1380, clrTh=1602
WIND BER TH: SetCross=0x001F, setTh=237
Counts= 0, 0, 0, 0, 0,
BER TH: BER 1e-6, BIP Sum=0, setTh=1245, clrTh=1458
WIND BER TH: SetCross=0x01FF, setTh=105
Counts= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
BER TH: BER 1e-7, BIP Sum=0, setTh=1248, clrTh=1458
WIND BER TH: SetCross=0x03FF, setTh=93
Counts= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
BER TH: BER 1e-8, BIP Sum=0, setTh=1248, clrTh=1458
WIND BER TH: SetCross=0x03FF, setTh=93
Counts= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
BER TH: BER 1e-9, BIP Sum=0, setTh=1248, clrTh=1458
WIND BER TH: SetCross=0x03FF, setTh=93
Counts= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
BER TH: BER 1e-10, BIP Sum=0, setTh=0, clrTh=1458
WIND BER TH: SetCross=0x03FF, setTh=0
Counts= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

```

Related Commands **show interface pos**
clear counters

show interface pos <interface-number>

Use this command to display the status of the POS.

| Parameter | Description |
|------------------|-----------------------------------|
| interface-number | Number of the POS interface <0-1> |

Defaults

N/A

Command Modes

Privileged EXEC

Usage Guidelines

This command may be used to help diagnose and isolate POS or SONET/SDH problems.

Examples

```

Gateway#show interfaces pos0
POS0 is up, line protocol is up
  Hardware is Packet/Ethernet over Sonet
  Description: foo bar
  MTU 4470 bytes, BW 155520 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 32, loopback not set
  Keepalive set (10 sec)
  Scramble enabled
  Last input 00:00:09, output never, output hang never
  Last clearing of "show interface" counters 05:17:30
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec

    2215 total input packets, 223743 post-HDLC bytes
    0 input short packets, 223951 pre-HDLC bytes
    0 input long packets , 0 input runt packets
    0 input CRCError packets , 0 input drop packets
    0 input abort packets
    0 input packets dropped by ucode

    0 packets input, 0 bytes
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 parity
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort

  2216 total output packets, 223807 output pre-HDLC bytes
  224003 output post-HDLC bytes

  0 packets output, 0 bytes, 0 underruns
  0 output errors, 0 applique, 8 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions

```

Related Commands Show controller pos
 Clear counters

show ons alarm

Use this command to display all the active alarms on the card.

Syntax Description This command has no arguments or keywords.

Defaults N/A

Command Modes Privileged EXEC

Usage Guidelines This command may be used to help diagnose and isolate card problems.

Examples

```
Gateway# show ons alarm
Equipment
Active Alarms: None

Port Alarms
  POS0 Active: TPTFAIL
  POS1 Active: TPTFAIL
  GigabitEthernet0 Active: None
  GigabitEthernet1 Active: None

POS0
Active Alarms : None
Demoted Alarms: None

POS1
Interface not provisioned
```

Related Commands

- show controller pos**
- show ons alarm defects**
- show ons alarm failures**

show ons alarm defect eqpt

This command displays the equipment layer defects.

Syntax Description This command has no arguments or keywords.

Defaults N/A

Command Modes Privileged EXEC

Usage Guidelines This command displays set of active defects for the equipment layer and the possible set of defects that can be set.

Examples

```
router#show ons alarm defect eqpt
Equipment Defects
Active: CONTBUS-IO-B
Reportable to TCC/CLI: CONTBUS-IO-A CONTBUS-IO-B CTNEQPT-PBWORK CTNEQPT-PBPROT EQPT
RUNCFG-SAVENEED ERROR-CONFIG
```

Related Commands [show ons alarm failures](#)

show ons alarm defect port

This command displays the port layer defects.

Syntax Description This command has no arguments or keywords.

Defaults N/A

Command Modes Privileged EXEC

Usage Guidelines This command displays set of active defects for the link layer and the possible set of defects that can be set. Note that the TPTFAIL defect can only occur on the POS ports and CARLOSS can only occur on the ethernet ports.

Examples

```
router#show ons alarm defect port
Port Defects
  POS0
  Active: TPTFAIL
  Reportable to TCC: CARLOSS TPTFAIL
  POS1
  Active: TPTFAIL
  Reportable to TCC: CARLOSS TPTFAIL
GigabitEthernet0
  Active: None
  Reportable to TCC: CARLOSS TPTFAIL
GigabitEthernet1
  Active: None
  Reportable to TCC: CARLOSS TPTFAIL
```

Related Commands `show interface`
`show ons alarm failures`

show ons alarm defect pos <interface-number>

This commands displays the link layer defects.

| Parameter | Description |
|------------------|-------------------------------|
| interface-number | Number of the interface <0-1> |

Defaults

N/A

Command Modes

Privileged EXEC

Usage Guidelines

This commands displays set of active defects for the POS layer and the possible set of defects that can be set.

Examples

```
POS0
Active Defects: None
Alarms reportable to TCC/CLI: PAIS PRDI PLOP PUNEQ PPLM PTIM PPDI BER_SF_B3 BER_SD_B3
```

Related Commands

show controller pos
show ons alarm failures

show ons alarm failure eqpt

This command displays the equipment layer failures.

Syntax Description This command has no arguments or keywords.

Defaults N/A

Command Modes Privileged EXEC

Usage Guidelines This command displays set of active failures for the equipment layer. If an EQPT alarm is present the Board Fail defect that was the source of the alarm will be displayed.

Examples

```
router#show ons alarm failure eqpt
Equipment
Active Alarms: None
```

Related Commands `show ons alarm defect`

show ons alarm failure port

This command displays the port layer failures.

Syntax Description This command has no arguments or keywords.

Defaults N/A

Command Modes Privileged EXEC

Usage Guidelines This command displays set of active failures for the link layer.

Examples

```
router#show ons alarm failure port
Port Alarms
  POS0 Active: TPTFAIL
  POS1 Active: TPTFAIL
  GigabitEthernet0 Active: None
  GigabitEthernet1 Active: None
```

Related Commands

- show interface**
- show ons alarm defect**

■ `show ons alarm failure pos <interface-number>`

show ons alarm failure pos *<interface-number>*

This command displays the link layer failures.

| Parameter | Description |
|------------------|-------------------------------|
| interface-number | Number of the interface <0-1> |

Defaults

N/A

Command Modes

Privileged EXEC

Usage Guidelines

This command displays set (active) failures for a specific interface at the pos layer. The display also specifies if an alarm has been demoted, as defined in Telcordia GR-253.

Examples

```
router#show ons alarm failure pos 0
POS0
Active Alarms : None
Demoted Alarms: None
```

Related Commands

`show controller pos`
`show ons alarm defect`

spr drpri-id < 0 | 1 >

Creates a DRPRI identification number of 0 or 1 to differentiate between the ML-Series cards paired for the Dual RPR Interconnect (DRPRI) protection feature.

Defaults N/A

Command Modes SPR Interface configuration

Usage Guidelines DRPRI paired sets share the same SPR station ID, so the DRPRI identification number helps identify a particular card in a DRPRI pair.

Examples The following example assigns a DRPRI identification number of zero to the SPR interface on an ML-Series card:

```
Router(config)# interface spr 1  
Router(config-if)# spr drpri-id 0
```

Related Commands

- interface spr 1**
- spr-intf-id**
- spr station-id**
- spr wrap**

spr-intf-id <*shared-packet -ring-number*>

Assigns the POS interface to the SPR interface.

| Parameter | Description |
|----------------------------------|---|
| <i>shared-packet-ring-number</i> | The only valid shared-packet-ring-number (SPR number) is 1. |

Defaults

N/A

Command Modes

POS Interface configuration

Usage Guidelines

- The SPR number must be 1, which is the same SPR number assigned to the SPR interface.
- The members of the SPR interface must be POS interfaces.
- An SPR interface is configured similarly to a EtherChannel (port-channel) interface. Instead of using the **channel-group** command to define the members, you use the **spr-intf-ID** command. And like port-channel, you then configure the SPR interfaces instead of the POS interface.

Examples

The following example assigns an ML-Series card POS interface to an SPR interface with the shared-packet-ring-number of 1:

```
Router(config)# interface pos 0
Router(config-if)# spr-intf-id 1
```

Related Commands

```
interface spr 1
spr drpri-id
spr station-id
spr wrap
```

spr station-id <station-id-number>

Configures a station ID.

| Parameter | Description |
|--------------------------|---|
| <i>station-id-number</i> | The user must configure a different number for each SPR interface that attaches to the RPR. Valid station ID numbers range from 1 to 254. |

Defaults

N/A

Command Modes

SPR Interface configuration

Usage Guidelines

The station ID differentiates among the SPR interfaces from the different ML-Series cards attached to the RPR.

Examples

The following example sets an ML-Series card SPR station ID to 100:

```
Router(config)# interface spr 1
Router(config-if)# spr station-id 100
```

Related Commands

interface spr 1
spr drpri-id
spr-intf-id
spr wrap

spr wrap <immediate | delayed>

Sets the RPR wrap mode to either wrap traffic the instant it detects a link state change or to wrap traffic after the carrier delay, which gives the SONET protection time to register the defect and declare the link down.

| Parameter | Description |
|-----------|---|
| immediate | Wraps RPR traffic the instant it detects a link state change. |
| delayed | Wraps RPR traffic after the carrier delay time expires. |

Defaults

The default setting is immediate.

Command Modes

SPR Interface configuration

Usage Guidelines

Immediate should be used if RPR is running over unprotected SONET circuits. Delayed should be run for SONET protected circuits (BLSR or path protection).

Examples

The following example sets an ML-Series card to delayed:

```
Router(config)# interface spr 1
Router(config-if)# spr wrap delayed
```

Related Commands

```
interface spr 1
spr drpri-id
spr-intf-id
spr station-id
```



Unsupported CLI Commands

This appendix lists some of the command-line interface (CLI) commands not supported in this release, either because they are not tested, or because of hardware limitations. These unsupported commands are displayed when you enter the question mark (?) at the CLI prompt. This is not a complete list. Unsupported commands are listed by command mode.

Unsupported Privileged Exec Commands

```
clear ip accounting
show ip accounting
show ip cache
show ip tcp header-compression
show ip mcache
show ip mpacket
```

Unsupported Global Configuration Commands

```
access-list aaa <1100-1199>
access-list aaa <200-299>
access-list aaa <700-799>
async-bootp
boot
bridge <num> acquire
bridge <num> address
bridge cmf
bridge <num> bitswap-layer3-addresses
bridge <num> circuit-group
bridge <num> domain
bridge <num> lat-service-filtering
bridge <num> protocol dec
```

bridge <num> protocol ibm
bridge <num> protocol vlan-bridge
chat-script
class-map match access-group
class-map match class-map
class-map match destination-address
class-map match mpls
class-map match protocol
class-map match qos-group
class-map match source-address
clns
define
dialer
dialer-list
downward-compatible-config
file
ip access-list log-update
ip access-list logging
ip address-pool
ip alias
ip bootp
ip gdp
ip local
ip reflexive-list
ip security
ip source-route
ip tcp
ipc
map-class
map-list
multilink
netbios
partition
policy-map class queue-limit
priority-list
queue-list
router iso-igrp
router mobile

service compress-config
service disable-ip-fast-frag
service exec-callback
service nagle
service old-slip-prompts
service pad
service slave-log
subscriber-policy

Unsupported POS Interface Configuration Commands

access-expression
autodetect
bridge-group x circuit-group
bridge-group x input-
bridge-group x lat-compression
bridge-group x output-
bridge-group x subscriber-loop-control
clock
clns
custom-queue-list
down-when-looped
fair-queue
flowcontrol
full-duplex
half-duplex
hold-queue
ip accounting
ip broadcast-address
ip load-sharing per-packet
ip route-cache
ip security
ip tcp
ip verify
iso-igrp
loopback
multilink-group
netbios

priority-group
pulse-time
random-detect
rate-limit
serial
service-policy history
source
timeout
transmit-interface
tx-ring-limit

Unsupported FastEthernet or GigabitEthernet Interface Configuration Commands

access-expression
cls
custom-queue-list
fair-queue
hold-queue
ip accounting
ip broadcast-address
ip load-sharing per-packet
ip route-cache
ip security
ip tcp
ip verify
iso-igrp
keepalive
loopback
max-reserved-bandwidth
multilink-group
netbios
priority-group
random-detect
rate-limit
service-policy history
timeout

transmit-interface
tx-ring-limit

Unsupported Port-Channel Interface Configuration Commands

access-expression
carrier-delay
cdp
clns
custom-queue-list
duplex
down-when-looped
encapsulation
fair-queue
flowcontrol
full-duplex
half-duplex
hold-queue
iso-igrp
keepalive
max-reserved-bandwidth
multilink-group
negotiation
netbios
ppp
priority-group
random-detect
rate-limit
timeout
tx-ring-limit

Unsupported BVI Interface Configuration Commands

access-expression
carrier-delay
cdp
clns
flowcontrol

hold-queue
iso-igrp
keepalive
l2protocol-tunnel
load-interval
max-reserved-bandwidth
mode
multilink-group
netbios
ntp
mtu
rate-limit
timeout
transmit-interface
tx-ring-limit



Using Technical Support

This appendix describes how to resolve problems with your ML-Series card.

The appendix contains the following sections:

- [Gathering Information About Your Internetwork, page C-1](#)
- [Getting the Data from Your ML-Series Card, page C-2](#)
- [Providing Data to Your Technical Support Representative, page C-3](#)

To help resolve these problems, use *Gathering Information About Your Internetwork* as a guideline for gathering relevant information about your network prior to calling.



Note

When you have a problem that you cannot resolve, contact the Cisco Technical Assistance Center (TAC).

Gathering Information About Your Internetwork

Before gathering any specific data, compile a list of all symptoms that users have reported on the internetwork (such as connections dropping or slow host response).

The next step is to gather specific information. Typical information needed to troubleshoot internetworking problems falls into two general categories: information required for any situation; and information specific to the topology, technology, or protocol.

Information that is always required by technical support engineers includes the following:

- Network topology map for the data network and the SONET/SDH topology and provisioning.
- List of hosts and servers: Include the host and server type, number on network, and a description of the host operating systems that are implemented.
- Configuration listing of all switch routers and switches involved.
- Complete specifications of all switch routers and switches involved.
- Version numbers of software (obtained with the **show version** command) and Flash code (obtained with the **show controllers** command) on all relevant switch routers and switches.
- List of network layer protocols, versions, and vendors.
- List of alarms and conditions on all nodes in the SONET/SDH topology.
- Node equipment and configuration; including type of cross-connect cards, ML-Series cards' slot numbers, OC-N cards, and TCC+ or TCC2.

To assist you in gathering this required data, the **show tech-support** EXEC command has been added in Cisco IOS Release 11.1(4) and later. This command provides general information about the switch router that you can provide to your technical support representative when you are reporting a problem.

The **show tech-support** command outputs the equivalent of the **show version**, **show running-config**, **show controllers**, **show stacks**, **show interfaces**, **show buffers**, **show process memory**, and **show process** EXEC commands.

The specific information requirements that might be needed by technical support vary depending on the situation. They include the following:

- Output from the following general **show** commands:
 - show interfaces**
 - show controllers**
 - show processes {cpu | mem}**
 - show buffer**
 - show mem summary**
- Output from the following protocol-specific **show** commands:
 - show protocol route**
 - show protocol traffic**
 - show protocol interfaces**
 - show protocol arp**
- Output from provisioning show commands
- Output from relevant **debug** privileged EXEC commands
- Output from protocol-specific **ping** and **trace** diagnostic tests, as appropriate
- Network analyzer traces, as appropriate
- Core dumps obtained using the **exception dump** command, or using the **write core** command if the system is operational, as appropriate

Getting the Data from Your ML-Series Card

When obtaining the information from your ML-Series card, you must tailor your method to the system that you are using to retrieve the information. Following are some hints for different platforms:

- PC and Macintosh—Connect a PC or Macintosh to the console port of the ML-Series card and log all output to a disk file (using a terminal emulation program). The exact procedure varies depending on the communication package used with the system.
- Terminal connected to the console port or remote terminal—The only way to get information with a terminal connected to the console port or with a remote terminal is to attach a printer to the AUX port on the terminal (if one exists) and to force all screen output to go to the printer. Using a terminal is undesirable because there is no way to capture the data to a file.
- UNIX workstation—At the UNIX prompt, enter the command **script filename**, then use Telnet to connect to the ML-Series card. The UNIX **script** command captures all screen output to the specified filename. To stop capturing output and close the file, enter the end-of-file character (typically **Ctrl-D**) for your UNIX system.

**Note**

To get your system to automatically log specific error messages or operational information to a UNIX syslog server, enter the **logging internet-address** command. For more information about using the **logging** command and setting up a syslog server, refer to the Cisco IOS configuration guides and command references.

Providing Data to Your Technical Support Representative

When submitting information to your technical support representative, electronic data is preferred. Electronic data significantly eases the transfer of information between technical support personnel and development staff. Common electronic formats include data sent through electronic mail and files sent using FTP.

If you are submitting data to your technical support representative, use the following list (in order of most to least favorable) to determine the preferred method for submission:

- The preferred method of information submission is through FTP service over the Internet. If your environment supports FTP, you can place your file in the incoming directory on the host Cisco.com.
- The next best method is to send data by e-mail. Before using this method, be sure to contact your technical support representative, especially when transferring binary core dumps or other large files.
- If you use e-mail, do not use encoding methods such as binhex or zip. Only MIME-compliant mail should be used.
- Transfer through a PC-based communications protocol, such as Kermit, to upload files to Cisco.com. Again, be sure to contact your technical support representative before attempting any transfer.
- Transfer by disk or tape.
- The least favorable method is hard-copy transfer by fax or physical mail.



Numerics

802.1D *see* STP

802.1Q

tunnel ports with other features [8-4](#)

802.1Q tunneling

compatibility with other features [8-4](#)

defaults [8-4](#)

described [8-1](#)

A

abbreviating commands [3-12](#)

ABRs [10-10](#)

access control lists *see* ACL [15-1](#)

ACL

about [15-1](#)

applying ACLs [15-4](#)

creating

extended IP ACLs [15-3](#)

IP ACLs [15-3](#)

named extended IP ACLs [15-4](#)

named IP ACLs [15-3](#)

named standard IP ACLs [15-4](#)

numbered standard IP ACLs [15-3](#)

implementation guidelines IP ACL [15-2](#)

named IP ACL [15-2](#)

adapter cable [3-4](#)

addresses

dynamic

accelerated aging [6-9](#)

default aging [6-9](#)

multicast, STP address management [6-8](#)

administrative distances

OSPF [10-18](#)

routing protocol defaults [10-33](#)

advertisements RIP [10-5](#)

aging time, accelerated for STP [6-9, 6-18](#)

alarms [4-15](#)

area border routers *see* ABRs

ASBRs [10-10](#)

autonomous system boundary routers

see ASBRs

B

bandwidth command traffic classes [13-13](#)

BGP, about [10-28](#)

Border Gateway Protocol

See BGP

BPDU RSTP format [6-12](#)

bridge-group command [4-4, 4-5, 4-12, 5-2](#)

bridge groups, routing [11-1](#)

bridge-group virtual interface *see* BVIs

bridge irb command [11-3](#)

bridge priority command [5-2](#)

bridge protocol command [5-2](#)

bridging

configuring [5-2](#)

description [5-1](#)

feature list [1-3](#)

monitoring and verifying [5-3](#)

bvi command [11-3](#)

BVIs

configuring [11-3](#)

description [11-1](#)

displaying information about [11-5](#)
 routing enabled on [11-2](#)

C

cable, RJ-11 to RJ-45 adapter [3-4](#)
 card description [1-1](#)
 CDP, Layer 2 protocol tunneling [8-6](#)
 channel-group command [9-3, 9-5](#)
 Cisco IOS
 backing out one level [3-12](#)
 bridging functionality [5-1](#)
 command modes [3-10 to 3-12](#)
 console configuration mode [3-11](#)
 global configuration mode [3-11](#)
 interface configuration mode [3-11](#)
 listing commands [3-12](#)
 privileged EXEC mode [3-11](#)
 software basics [3-10](#)
 startup configuration file [3-9](#)
 upgrading image [1-7](#)
 user EXEC mode [3-11](#)
 clear bridge command [5-3](#)
 clear vlan command [7-5](#)
 clear vlan statistics command [5-3](#)
 commands
 bridge-group [4-4, 4-5, 4-12, 5-2](#)
 bridge irb [11-3](#)
 bridge priority [5-2](#)
 bridge protocol [5-2](#)
 bridge protocol drpri-rstp [A-2](#)
 channel-group [9-3, 9-5](#)
 clear bridge [5-3](#)
 clear vlan [7-5](#)
 clear vlan statistics [5-3](#)
 debug vlan packet [7-5](#)
 hostname [3-9](#)
 interface bvi [11-3](#)
 interface spr 1 [A-4](#)
 ip multicast-routing [10-35](#)
 ip pim [10-35](#)
 line vty [3-8](#)
 listing [3-12](#)
 network area [10-3](#)
 reference chapter [A-1](#)
 router bgp [10-4](#)
 router eigrp [10-3](#)
 router ospf [10-3](#)
 sdm size [14-3](#)
 show bridge [5-3](#)
 show bridge group [5-3](#)
 show interfaces bvi [11-5](#)
 show interfaces irb [11-5](#)
 show interfaces port-channel [9-9](#)
 show ip mroute [10-36](#)
 show sdm size [14-3](#)
 show span [5-3](#)
 show tech-support [C-2](#)
 show vlan [7-5](#)
 show vlans [5-3](#)
 spr drpri-id [A-21](#)
 spr-intf-id [A-22](#)
 spr station-id [A-23](#)
 spr wrap [A-24](#)
 configuration mode
 console [3-11](#)
 global [3-11](#)
 configuring [16-12](#)
 bridging [5-1](#)
 BVs [11-3](#)
 EtherChannel encapsulation [9-7](#)
 Fast EtherChannel [9-2](#)
 host name [3-9](#)
 integrated routing and bridging
 See IRB
 interface, overview [4-1](#)
 IP [10-1](#)
 IP multicast [10-34](#)

ISL over FEC 9-7
 management port 3-8
 VLANs 7-1

connecting to console port 3-5
 connection procedures 3-5 to 3-6
 console port, connecting to 3-5
 CoS-based QoS 13-16
 cos commit command 13-16

CTC
 Ethernet port provisioning information 2-5
 Ethernet statistics 2-1
 IOS on CTC 3-2
 POS port provisioning information 2-7
 POS statistics 2-3
 SONET alarms 2-8
 SONET circuit provisioning 2-9

D

debug vlan packet command 7-5

default configuration
 EIGRP 10-22
 Layer 2 protocol tunneling 8-7
 OSPF 10-10
 RIP 10-5
 STP 6-15

dense mode, PIM 10-34

Diffusing Update Algorithm (DUAL) 10-21

double-tagged packets
 802.1Q tunneling 8-2
 Layer 2 protocol tunneling 8-7

DRPRI 16-10, 16-12
 example 16-12
 overview 1-7

DUAL finite state machine, EIGRP 10-21

dynamic addresses
See addresses

E

EIGRP
 authentication 10-25
 components 10-21
 configuring 10-23
 default configuration 10-22
 definition 10-21
 interface parameters, configuring 10-24
 monitoring 10-26

e-mail, technical support C-3

enable mode 3-11

enable passwords 3-7

enable secret passwords 3-7

encapsulation
 configuring 802.1Q VLANs 7-2
 configuring EtherChannels 9-7

Enhanced IGRP *see* EIGRP

error messages, logging C-3

EtherChannel
 configuring encapsulation 9-7
 ISL VLANs 9-2
 port channels supported 9-2

Ethernet configuration tasks 4-4

extended system ID, STP 6-4

F

Fast EtherChannel *see* FEC

Fast Ethernet
 configuring autonegotiation 4-4
 configuring interfaces 4-4
 Fast EtherChannel 9-2

feature list 1-2

FEC
 cautions 9-3, 9-5, 12-2
 configuring 9-2, 9-4, 12-2
 configuring encapsulation 9-7
 configuring ISL 9-7

port channels supported [9-2](#)

G

GEC

bandwidth scalability [9-2](#)
 configuring [9-2, 9-4, 12-2](#)
 configuring encapsulation [9-7](#)

Gigabit EtherChannel *see* GEC

Gigabit Ethernet

configuring autonegotiation [4-5, 4-12](#)
 configuring interfaces [4-5, 4-12](#)
 global configuration mode [3-11](#)

H

hostname command [3-9](#)

HSRP, EtherChannel compatibility with [9-2](#)

I

IGMP [10-34](#)

IGP [10-9](#)

integrated routing and bridging *see* IRB

interface configuration mode [3-11](#)

interface parameters, configuring

EtherChannel [9-2, 9-5, 12-2](#)
 general [4-3](#)
 overview [4-1](#)

interface port IDs [4-2](#)

Interior Gateway Protocol *see* IGP

Internet Group Membership Protocol *see* IGMP

Internet protocol multicast

See IP multicast routing

Inter-Switch Link protocol *see* ISL

IOS commands [A-1](#)

IP access control list

See ACL

IP multicast routing

description [10-34](#)

IGMP [10-34](#)

PIM [10-34](#)

ip multicast-routing command [10-35](#)

ip pim command [10-35](#)

IP routes, monitoring [10-33](#)

IP routing protocols, configuration tasks [10-1](#)

IP unicast routing

administrative distances [10-33](#)

configuring static routes [10-32](#)

IGP [10-9](#)

IRB

BVIs [11-1](#)

configuration considerations [11-1](#)

configuring [11-2](#)

description [11-1](#)

displaying information about [11-5](#)

monitoring and verifying [11-4](#)

ISL [9-2](#)

K

keepalive command [4-14](#)

Kermit protocol [C-3](#)

L

Layer 2 feature list [1-3](#)

Layer 2 protocol tunneling

configuring [8-7](#)

default configuration [8-7](#)

defined [8-6](#)

guidelines [8-7](#)

Layer 3 feature list [1-5](#)

line vty command [3-8](#)

link state advertisements (LSAs) [10-15](#)

logging command [C-3](#)

logging router output [C-2](#)

M

MAC addresses [4-2](#)

management ports
 configuring [3-8](#)

See also console ports

match any command [13-11](#)

match cos command [13-11](#)

match ip dscp command [13-12](#)

match ip precedence command [13-12](#)

Media Access Control addresses *see* MAC addresses

message logging [C-3](#)

metro tags [8-2](#)

Modular QoS Command-Line Interface

configuration (example) [13-17](#)

configuration, verifying [13-16](#)

configuring [13-10](#)

monitoring

802.1Q tunneling [8-9](#)

EIGRP [10-26](#)

IP routes [10-33](#)

Layer 2 protocol tunneling [8-9](#)

OSPF [10-20, 10-33](#)

tunneling [8-9](#)

MSTP, interoperability with 802.1D [6-14](#)

multicast, IP *see* IP multicast routing

N

neighbor discovery/recovery, EIGRP [10-21](#)

networking protocols

IP multicast routing [10-34 to 10-35](#)

IP routing [10-1](#)

not-so-stubby areas *see* NSSA

NSSA, OSPF [10-15](#)

O

OSPF

area parameters, configuring [10-15](#)

configuring [10-3, 10-12](#)

default configuration

metrics [10-17](#)

route [10-17](#)

settings [10-10](#)

described [10-9](#)

interface parameters, configuring [10-13](#)

LSA group pacing [10-19](#)

monitoring [10-20, 10-33](#)

network area command [10-3](#)

process ID [10-3](#)

router IDs [10-19](#)

route summarization [10-17](#)

virtual links [10-17](#)

P

passive interface OSPF [10-18](#)

passwords [3-7](#)

path cost for STP [6-17](#)

PC, connecting to switch [3-5](#)

per-VLAN Spanning Tree+ (PVST+) [6-8](#)

PIM

configuring [10-35](#)

modes [10-34 to 10-35](#)

rendezvous point [10-35](#)

pin mappings for RJ-11 to RJ-45 [3-4](#)

port-channel command [9-2](#)

port channels [9-2](#)

port IDs [4-2](#)

port priority, STP [6-16](#)

POS

configuring interfaces [4-12](#)

description [4-8](#)

SONET alarms [4-15, 4-17](#)

pos delay triggers command [4-17](#)
 pos report command [4-16](#)
 pos scramble-atm command [4-15](#)
 privileged EXEC mode [3-11](#)
 procedures, connection [3-5 to 3-6](#)
 protocol-dependent modules, EIGRP [10-22](#)
 Protocol Independent Multicast *see* PIM

Q

QoS policers [13-14](#)

R

reliable transport protocol, EIGRP [10-21](#)
 remote terminals, logging router output [C-2](#)
 rendezvous points [10-35](#)
 RFC
 1058, RIP [10-5](#)
 1253, OSPF [10-9](#)
 1587, NSSAs [10-10](#)
 RIP
 advertisements [10-5](#)
 authentication [10-8](#)
 configuring [10-6](#)
 default configuration [10-5](#)
 described [10-5](#)
 hop counts [10-5](#)
 split horizon [10-9](#)
 summary addresses [10-9](#)
 RJ-11 to RJ-45 console cable adapter [3-4](#)
 RJ-45 connector, console port [3-6](#)
 route calculation timers, OSPF [10-18](#)
 router bgp command [10-4](#)
 router eigrp command [10-3](#)
 router ID, OSPF [10-19](#)
 router isis command [10-30](#)
 router ospf command [10-3](#)

route summarization, OSPF [10-17](#)
 routing protocol administrative distances [10-33](#)
 RPF [10-35](#)
 RPR
 Carrier delay [16-3](#)
 configuring [16-4](#)
 CoS-based QoS [13-16](#)
 DRPRI [16-10](#)
 example [16-7](#)
 MAC address and VLAN support [16-4](#)
 overview [1-6](#)
 QoS [13-10](#)
 understanding [16-1](#)

RSTP

overview [6-9](#)
 active topology, determining [6-10](#)
 BPDU
 format [6-12](#)
 processing [6-13](#)
 designated port, defined [6-10](#)
 designated switch, defined [6-10](#)
 interoperability with 802.1D
 described [6-14](#)
 topology changes [6-13](#)
 port roles
 described [6-9](#)
 synchronized [6-12](#)
 proposal-agreement handshake process [6-11](#)
 rapid convergence
 point-to-point links [6-11](#)
 root ports [6-10](#)
 root port, defined [6-9](#)

S

script command [C-2](#)
 SDH
 alarms [1-6](#)
 bandwidth [1-6](#)

- encapsulation [1-5](#)
- SDH alarms [4-15](#)
- SDM
 - configuring
 - autolearn [14-2](#)
 - size [14-2](#)
 - regions [14-1](#)
 - see also* TCAM
- sdm access-list command [14-3](#)
- sdm size command [14-3](#)
- service-policy command, traffic policies [13-15](#)
- service-policy input command [13-16](#)
- service-policy output command [13-16](#)
- service-provider networks
 - and 802.1Q tunneling [8-1](#)
 - and customer VLANs [8-2](#)
 - Layer 2 protocols across [8-7](#)
- set qos-group command [13-15](#)
- show bridge command [5-3](#)
- show bridge group command [5-3](#)
- show interfaces bvi command [11-5](#)
- show interfaces irb command [11-5](#)
- show interfaces port-channel command [9-9](#)
- show ip mroute command [10-36](#)
- show policy-map command
 - traffic classes [13-16](#)
- show sdm size command [14-3](#)
- show span command [5-3](#)
- show tech-support command [C-2](#)
- show vlan command [7-5](#)
- show vlans command [5-3](#)
- SNMP [1-7](#)
- SONET
 - alarms [1-6](#)
 - bandwidth [1-5](#)
 - encapsulation [1-5](#)
- SONET alarms [4-15](#)
- sparse mode, PIM [10-35](#)
- startup configuration file [3-9](#)
- static routes, configuring [10-32](#)
- statistics, OSPF [10-20, 10-33](#)
- STP
 - BPDU message exchange [6-2](#)
 - configuring
 - forward-delay time [6-18](#)
 - hello time [6-18](#)
 - path cost [6-17](#)
 - port priority [6-16](#)
 - root switch [6-16](#)
 - switch priority [6-18](#)
 - default configuration [6-15](#)
 - designated port, defined [6-3](#)
 - designated switch, defined [6-3](#)
 - disabling [6-15](#)
 - displaying status [6-19](#)
 - extended system ID
 - overview [6-4](#)
 - unexpected behavior [6-16](#)
 - forward-delay time [6-6](#)
 - inferior BPDU [6-3](#)
 - interface states
 - blocking [6-6](#)
 - disabled [6-7](#)
 - forwarding [6-6, 6-7](#)
 - learning [6-7](#)
 - listening [6-7](#)
 - overview [6-5](#)
 - Layer 2 protocol tunneling [8-6](#)
 - limitations with 802.1Q trunks [6-8](#)
 - multicast addresses, affect of [6-8](#)
 - overview [6-2](#)
 - redundant connectivity [6-8](#)
 - root port, defined [6-3](#)
 - root switch
 - effects of extended system ID [6-4](#)
 - election [6-3](#)
 - unexpected behavior [6-16](#)
 - superior BPDU [6-3](#)

supported number of spanning-tree instances [6-2, 6-9](#)
 timers, described [6-4](#)
 stub areas, OSPF [10-15](#)
 support, technical
 See technical support
 syslog server [C-3](#)
 system MTU
 802.1Q tunneling [8-4](#)
 maximums [8-4](#)

T

tagged packets, Layer 2 protocol [8-6](#)
 TCAM
 entries [14-2](#)
 Layer 3 switching information [14-1](#)
 protocol regions [14-1](#)
 space [14-1](#)
 See also SDM
 technical support
 FTP service [C-3](#)
 gathering data [C-1](#)
 logging router output [C-2](#)
 providing data [C-3](#)
 show tech-support command [C-2](#)
 terminals
 connecting to switch [3-5](#)
 terminal-emulation software [3-5](#)
 terminals, logging router output [C-2](#)
 ternary content addressable memory
 See TCAM
 traffic classes [13-11](#)
 traffic policies
 creating [13-12](#)
 interfaces, attaching [13-15](#)
 trunk ports [7-1](#)
 tunneling
 802.1Q [8-1](#)
 defined [8-1](#)

Layer 2 protocol [8-6](#)
 tunnel ports
 802.1Q, configuring [8-4, 8-8](#)
 described [8-1](#)
 incompatibilities with other features [8-4](#)

U

understanding [16-10](#)
 user EXEC mode [3-11](#)

V

verifying
 10/100BASE-T configuration [4-6](#)
 IP multicast operation [10-35](#)
 VLAN operation [7-5](#)
 virtual LANs
 See VLANs
 VLANs
 aging dynamic addresses [6-9](#)
 configuring 802.1Q [7-2](#)
 customer numbering in service-provider networks [8-3](#)
 number per system [7-1](#)
 STP and 802.1Q trunks [6-8](#)
 trunk ports [7-1](#)
 VRF Lite
 configuring [12-2](#)
 example [12-2](#)
 monitoring and verifying [12-7](#)
 understanding [12-1](#)
 VTP Layer 2 protocol tunneling [8-6](#)
 vty [3-4](#)