Cisco ONS 15454 and Cisco ONS 15454 SDH Ethernet Card Software Feature and Configuration Guide

Cisco IOS Release 12.2 (29a) SV
CTC and Documentation Release 8.5.4
June 2010

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

Text Part Number: 78-18113-01

CONTENTS

FIGURES

TABLES

Preface

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
- Obtaining Optical Networking Information
- Obtaining Documentation and Submitting a Service Request

Revision History

Date	Notes
March 2008	Added section "Terminal and Facility Loopback on LCAS Circuits In Split Fibre Routing" in the chapter CE-Series Ethernet Cards.
April 2008	• Added a new section "CE-MR-10 CCAT//VCCAT/SW-LCAS/ HW-LCAS Behavior" in section, "CE-MR-10 Ethernet Card", chapter "CE-Series Ethernet Cards". • Added notes on limitations in creating some STS and VC4 circuits in section "Available Circuit Sizes and Combinations" chapter, CE-Series Ethernet Cards.

<table>
<tr><th>Date</th><th>Notes</th></tr>
<tr><td>July 2008</td><td>Updated section “Examples” in chapter “Configuring Ethernet Virtual Circuits”.
Updated section “Flow Control Pause and QoS” in chapter “Configuring Quality of Service”.</td></tr>
<tr><td>August 2008</td><td>Reference to Auto-MDIX is removed from the following sections:
“Configuring the Fast Ethernet Interfaces for the ML100X-8” in chapter “Configuring Interfaces”.
“ML-Series Feature List” in chapter “ML-Series Card Overview”.
Added a new section “VCAT Circuit Provisioning Time Slot Limitations” in chapter “CE-Series Ethernet Cards”.</td></tr>
<tr><td>September 2008</td><td>Corrected the footnote of Table 5-1 in chapter “Configuring POS”.
Updated the sections “CE-100T-8 VCAT Characteristics” and “CE-MR-10 CCAT, VCAT, SW-LCAS, HW-LCAS Behavior” in chapter “CE-Series Ethernet Cards”.</td></tr>
<tr><td>November 2008</td><td>Added note in section “Terminal and Facility Loopback on LCAS Circuits In Split Fibre Routing” in chapter “CE-Series Ethernet Cards”.</td></tr>
<tr><td>December 2008</td><td>Added the following sections in chapter “Configuring Link Aggregation”:
Load Balancing on the ML-Series Cards
Load Balancing on the ML-MR-10 Card
VLAN Load Balancing
Configuration Commands for Load Balancing.
Added a caution in section “CE-MR-10 CCAT, VCAT, SW-LCAS, HW-LCAS Behavior”, chapter “CE-Series Ethernet Cards”.
Added a caution in section “VCAT” in chapter “Configuring POS”.</td></tr>
<tr><td>January 2009</td><td>Added the following sections in chapter “Configuring Quality of Service”:
QoS not Configured on Egress
ML-Series Egress Bandwidth Example
Updated section “IP SLA Restrictions on the ML-Series”.
Added Tables 10-1 and 10-3 and updated Table 10-4 in section “Load Balancing on the ML-Series Cards” section in chapter “Configuring Link Aggregation”.</td></tr>
<tr><td>May 2009</td><td>Added a note in section, “EoMPLS Configuration on PE-CLE SPR Interface” in chapter, “Configuring Ethernet over MPLS”.</td></tr>
<tr><td>June 2009</td><td>Added a note in section, “CE-MR-10 CCAT, VCAT, SW-LCAS, HW-LCAS Behavior” in chapter, “CE-Series Ethernet Cards”.</td></tr>
<tr><td>August 2009</td><td>Updated Ethernet wire speed values in Table 21-18 and Table 21-19 in chapter, “CE-Series Ethernet Cards”.</td></tr>
</table>

Date	Notes
October 2009	Deleted caution "Do not use the abbreviations g0 or g1 for Gigabit Ethernet user-defined abbreviations. This creates an unsupported group asynchronous interface" from chapter, "Configuring Interfaces".
February 2010	Updated image in the chapter, "Configuring VRF Lite".

Document Objectives

This guide covers the software features and operations of Ethernet cards for the Cisco ONS 15454 and Cisco ONS 15454 SDH. It 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 or Cisco ONS 15454 SDH system. It also explains software feature and configuration for CTC on the E-Series, G-Series and CE-Series cards. The E-Series cards and G-Series cards are modules in the Cisco ONS 15454 and Cisco ONS 15454 SDH. The CE-Series cards are modules in the Cisco ONS 15454. The CE-100T-8 is also available as module for the Cisco ONS 15310-CL. The Cisco ONS 15310-CL version of the card is covered in the *Cisco ONS 15310-CL and Cisco ONS 15310-MA Ethernet Card Software Feature and Configuration Guide*. Use this guide in conjunction with the appropriate publications listed in the Related Documentation section.

Audience

To use the ML-Series card chapters of this publication, you should be familiar with Cisco IOS and preferably have technical networking background and experience. To use the E-Series, G-Series and CE-Series card chapters of this publication, you should be familiar with CTC and preferably have technical networking background and experience.

Document Organization

The *Cisco ONS 15454 and Cisco ONS 15454 SDH Ethernet Card Software Feature and Configuration Guide R8.5.x* is organized into the following chapters:

- Chapter 1, "ML-Series Card 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 basic procedures for the interfaces.
- Chapter 5, "Configuring POS," provides information on the ML-Series card POS interfaces and advanced procedures for the POS interfaces.
- Chapter 6, "Configuring Bridges," provides bridging examples and procedures for the ML-Series card.
- Chapter 7, "Configuring STP and RSTP," provides spanning tree and rapid spanning tree examples and procedures for the ML-Series card.

- Chapter 8, "Configuring VLANs," provides VLAN examples and procedures for the ML-Series card.
- Chapter 9, "Configuring IEEE 802.1Q Tunneling and Layer 2 Protocol Tunneling," provides tunneling examples and procedures for the ML-Series card.
- Chapter 10, "Configuring Link Aggregation," provides Etherchannel and packet-over-SONET/SDH (POS) channel examples and procedures for the ML-Series card.
- Chapter 11, "Configuring Networking Protocols," provides network protocol examples and procedures for the ML-Series card.
- Chapter 12, "Configuring IRB," provides integrated routing and bridging (IRB) examples and procedures for the ML-Series card.
- Chapter 13, "Configuring VRF Lite," provides VPN Routing and Forwarding Lite (VRF Lite) examples and procedures for the ML-Series card.
- Chapter 14, "Configuring Quality of Service," provides quality of service (QoS) examples and procedures for the ML-Series card.
- Chapter 15, "Configuring the Switching Database Manager," provides switching database manager examples and procedures for the ML-Series card.
- Chapter 16, "Configuring Access Control Lists," provides access control list (ACL) examples and procedures for the ML-Series card.
- Chapter 17, "Configuring Cisco Proprietary Resilient Packet Ring," provides resilient packet ring (RPR) examples and procedures for the ML-Series card.
- Chapter 18, "Configuring IEEE 802.17b Resilient Packet Ring," provides IEEE 802.17b-based resilient packet ring (RPR-IEEE) examples and how to configure it on the ML-Series cards.
- Chapter 19, "Configuring Ethernet over MPLS," provides Ethernet over Multiprotocol Label Switching (EoMPLS) examples and procedures for the ML-Series card.
- Chapter 20, "Configuring Security for the ML-Series Card," describes the security features of the ML-Series card.
- Chapter 21, "CE-100T-8 Ethernet Card," describes the operation of the CE-1000-4 card.
- Chapter 22, "POS on ONS Ethernet Cards," details and explains POS on Ethernet cards. It also details Ethernet card interoperability.
- Chapter 23, "Configuring RMON," describes how to configure remote network monitoring (RMON) on the ML-Series card.
- Chapter 24, "Configuring SNMP," describes how to configure the ML-Series card for operating with Simple Network Management Protocol (SNMP).
- Chapter 25, "E-Series and G-Series Ethernet Operation," details and explains the features and operation of E-Series and G-Series Ethernet cards for the ONS 15454, ONS 15454 SDH and ONS 15327 platform.
- Chapter 26, "Configuring CDP," describes how to configure Cisco Discovery Protocol (CDP) on the ML-Series card or the ML-MR-10 card.
- Chapter 27, "Configuring CPP," describes card and port protection (CPP) for ML-MR-10 card and how to configure CPP using Cisco IOS command line interface (CLI). For information on ML-MR-10 card features.
- Chapter 28, "Configuring Ethernet Vitrual Circuits," provides information about configuring Ethernet Virtual Circuits (EVC) for the ONS 15454, ML-MR-10 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 (Cisco TAC) for ML-Series card problems.

Related Documentation

Use the *Cisco ONS 15454 and Cisco ONS 15454 SDH Ethernet Card Software Feature and Configuration Guide R8.5.x* in conjunction with the following general ONS 15454 or ONS 15454 SDH system publications:

- *Cisco ONS 15454 Procedure Guide*
  Provides procedures to install, turn up, provision, and maintain a Cisco ONS 15454 node and network.
- *Cisco ONS 15454 SDH Procedure Guide*
  Provides procedures to install, turn up, provision, and maintain a Cisco ONS 15454 SDH node and network.
- *Cisco ONS 15454 Reference Manual*
  Provides detailed card specifications, hardware and software feature descriptions, network topology information, and network element defaults.
- *Cisco ONS 15454 SDH Reference Manual*
  Provides detailed card specifications, hardware and software feature descriptions, network topology information, and network element defaults.
- *Cisco ONS 15454 Troubleshooting Guide*
  Provides alarm descriptions, alarm and general troubleshooting procedures, error messages, and performance monitoring and SNMP parameters.
- *Cisco ONS 15454 SDH Troubleshooting Guide*
  Provides general troubleshooting procedures, alarm descriptions and troubleshooting procedures, error messages, and performance monitoring and SNMP parameters.
- *Cisco ONS SONET TL1 Command Guide*
  Provides a full TL1 command and autonomous message set including parameters, AIDs, conditions, and modifiers for the Cisco ONS 15454, ONS 15327, ONS 15600, ONS 15310-CL, and ONS 15310-MA systems.
- *Cisco ONS 15454 SDH TL1 Command Guide*
  Provides a full TL1 command and autonomous message set including parameters, AIDs, conditions and modifiers for the Cisco ONS 15454 SDH.
- *Cisco ONS SONET TL1 Reference Guide*
  Provides general information and procedures for TL1 in the Cisco ONS 15454, ONS 15327, ONS 15600, ONS 15310-CL, and Cisco ONS 15310-MA systems.
- *Cisco ONS 15454 SDH TL1 Reference Guide*
  Provides general information and procedures for TL1 in the Cisco ONS 15454 SDH.
- *Cisco ONS 15454 SDH TL1 Reference Guide*
  Provides general information, procedures, and errors for TL1 in the Cisco ONS 15454 SDH.

- *Release Notes for the Cisco ONS 15454 Release 7.0*
  Provides caveats, closed issues, and new feature and functionality information.
- *Release Notes for the Cisco ONS 15454 SDH Release 7.0*
  Provides caveats, closed issues, and new feature and functionality information.
- *Release Notes for the Cisco ONS 15327 Release 7.0*
  Provides caveats, closed issues, and new feature and functionality information.

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

- Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2
- Cisco IOS Quality of Service Solutions Command Reference, Release 12.2

The ML-Series card employs Cisco IOS 12.2. For more general information on Cisco IOS 12.2, refer to the extensive Cisco IOS documentation at:

- http://www.cisco.com/

Document Conventions

This publication uses the following conventions:

Convention	Application
boldface	Commands and keywords in body text.
italic	Command input that is supplied by the user.
[ ]	Keywords or arguments that appear within square brackets are optional.
{ x I x I 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. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device. Statement 1071

SAVE THESE INSTRUCTIONS

Waarschuwing **BELANGRIJKE VEILIGHEIDSINSTRUCTIES**

Dit waarschuwingssymbool betekent gevaar. U verkeert in een situatie die lichamelijk letsel kan veroorzaken. Voordat u aan enige apparatuur gaat werken, dient u zich bewust te zijn van de bij elektrische schakelingen betrokken risico's en dient u op de hoogte te zijn van de standaard praktijken om ongelukken te voorkomen. Gebruik het nummer van de verklaring onderaan de waarschuwing als u een vertaling van de waarschuwing die bij het apparaat wordt geleverd, wilt raadplegen.

BEWAAR DEZE INSTRUCTIES

Varoitus **TÄRKEITÄ TURVALLISUUSOHJEITA**

Tämä varoitusmerkki merkitsee vaaraa. Tilanne voi aiheuttaa ruumiillisia vammoja. Ennen kuin käsittelet laitteistoa, huomioi sähköpiirien käsittelemiseen liittyvät riskit ja tutustu onnettomuuksien yleisiin ehkäisytapoihin. Turvallisuusvaroitusten käännökset löytyvät laitteen mukana toimitettujen käännettyjen turvallisuusvaroitusten joukosta varoitusten lopussa näkyvien lausuntonumeroiden avulla.

SÄILYTÄ NÄMÄ OHJEET

Attention **IMPORTANTES INFORMATIONS DE SÉCURITÉ**

Ce symbole d'avertissement indique un danger. Vous vous trouvez dans une situation pouvant entraîner des blessures ou des dommages corporels. Avant de travailler sur un équipement, soyez conscient des dangers liés aux circuits électriques et familiarisez-vous avec les procédures couramment utilisées pour éviter les accidents. Pour prendre connaissance des traductions des avertissements figurant dans les consignes de sécurité traduites qui accompagnent cet appareil, référez-vous au numéro de l'instruction situé à la fin de chaque avertissement.

CONSERVEZ CES INFORMATIONS

Warnung **WICHTIGE SICHERHEITSHINWEISE**

Dieses Warnsymbol bedeutet Gefahr. Sie befinden sich in einer Situation, die zu Verletzungen führen kann. Machen Sie sich vor der Arbeit mit Geräten mit den Gefahren elektrischer Schaltungen und den üblichen Verfahren zur Vorbeugung vor Unfällen vertraut. Suchen Sie mit der am Ende jeder Warnung angegebenen Anweisungsnummer nach der jeweiligen Übersetzung in den übersetzten Sicherheitshinweisen, die zusammen mit diesem Gerät ausgeliefert wurden.

BEWAHREN SIE DIESE HINWEISE GUT AUF.

Avvertenza **IMPORTANTI ISTRUZIONI SULLA SICUREZZA**

Questo simbolo di avvertenza indica un pericolo. La situazione potrebbe causare infortuni alle persone. Prima di intervenire su qualsiasi apparecchiatura, occorre essere al corrente dei pericoli relativi ai circuiti elettrici e conoscere le procedure standard per la prevenzione di incidenti. Utilizzare il numero di istruzione presente alla fine di ciascuna avvertenza per individuare le traduzioni delle avvertenze riportate in questo documento.

CONSERVARE QUESTE ISTRUZIONI

Advarsel **VIKTIGE SIKKERHETSINSTRUKSJONER**

Dette advarselssymbolet betyr fare. Du er i en situasjon som kan føre til skade på person. Før du begynner å arbeide med noe av utstyret, må du være oppmerksom på farene forbundet med elektriske kretser, og kjenne til standardprosedyrer for å forhindre ulykker. Bruk nummeret i slutten av hver advarsel for å finne oversettelsen i de oversatte sikkerhetsadvarslene som fulgte med denne enheten.

TA VARE PÅ DISSE INSTRUKSJONENE

Aviso **INSTRUÇÕES IMPORTANTES DE SEGURANÇA**

Este símbolo de aviso significa perigo. Você está em uma situação que poderá ser causadora de lesões corporais. Antes de iniciar a utilização de qualquer equipamento, tenha conhecimento dos perigos envolvidos no manuseio de circuitos elétricos e familiarize-se com as práticas habituais de prevenção de acidentes. Utilize o número da instrução fornecido ao final de cada aviso para localizar sua tradução nos avisos de segurança traduzidos que acompanham este dispositivo.

GUARDE ESTAS INSTRUÇÕES

¡Advertencia! **INSTRUCCIONES IMPORTANTES DE SEGURIDAD**

Este símbolo de aviso indica peligro. Existe riesgo para su integridad física. Antes de manipular cualquier equipo, considere los riesgos de la corriente eléctrica y familiarícese con los procedimientos estándar de prevención de accidentes. Al final de cada advertencia encontrará el número que le ayudará a encontrar el texto traducido en el apartado de traducciones que acompaña a este dispositivo.

GUARDE ESTAS INSTRUCCIONES

Varning! **VIKTIGA SÄKERHETSANVISNINGAR**

Denna varningssignal signalerar fara. Du befinner dig i en situation som kan leda till personskada. Innan du utför arbete på någon utrustning måste du vara medveten om farorna med elkretsar och känna till vanliga förfaranden för att förebygga olyckor. Använd det nummer som finns i slutet av varje varning för att hitta dess översättning i de översatta säkerhetsvarningar som medföljer denna anordning.

SPARA DESSA ANVISNINGAR

FONTOS BIZTONSÁGI ELOÍRÁSOK

Ez a figyelmezeto jel veszélyre utal. Sérülésveszélyt rejto helyzetben van. Mielott bármely berendezésen munkát végezte, legyen figyelemmel az elektromos áramkörök okozta kockázatokra, és ismerkedjen meg a szokásos balesetvédelmi eljárásokkal. A kiadványban szereplo figyelmeztetések fordítása a készülékhez mellékelt biztonsági figyelmeztetések között található; a fordítás az egyes figyelmeztetések végén látható szám alapján keresheto meg.

ORIZZE MEG EZEKET AZ UTASÍTÁSOKAT!

Предупреждение **ВАЖНЫЕ ИНСТРУКЦИИ ПО СОБЛЮДЕНИЮ ТЕХНИКИ БЕЗОПАСНОСТИ**

Этот символ предупреждения обозначает опасность. То есть имеет место ситуация, в которой следует опасаться телесных повреждений. Перед эксплуатацией оборудования выясните, каким опасностям может подвергаться пользователь при использовании электрических цепей, и ознакомьтесь с правилами техники безопасности для предотвращения возможных несчастных случаев. Воспользуйтесь номером заявления, приведенным в конце каждого предупреждения, чтобы найти его переведенный вариант в переводе предупреждений по безопасности, прилагаемом к данному устройству.

СОХРАНИТЕ ЭТИ ИНСТРУКЦИИ

警告 重要的安全性说明

此警告符号代表危险。您正处于可能受到严重伤害的工作环境中。在您使用设备开始工作之前，必须充分意识到触电的危险，并熟练掌握防止事故发生的标准工作程序。请根据每项警告结尾提供的声明号码来找到此设备的安全性警告说明的翻译文本。

请保存这些安全性说明

警告 安全上の重要な注意事項

「危険」の意味です。人身事故を予防するための注意事項が記述されています。装置の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止策に留意してください。警告の各国語版は、各注意事項の番号を基に、装置に付属の「Translated Safety Warnings」を参照してください。

これらの注意事項を保管しておいてください。

주의 중요 안전 지침

이 경고 기호는 위험을 나타냅니다. 작업자가 신체 부상을 일으킬 수 있는 위험한 환경에 있습니다. 장비에 작업을 수행하기 전에 전기 회로와 관련된 위험을 숙지하고 표준 작업 관례를 숙지하여 사고를 방지하십시오. 각 경고의 마지막 부분에 있는 경고문 번호를 참조하여 이 장치와 함께 제공되는 번역된 안전 경고문에서 해당 번역문을 찾으십시오.

이 지시 사항을 보관하십시오.

Aviso **INSTRUÇÕES IMPORTANTES DE SEGURANÇA**

Este símbolo de aviso significa perigo. Você se encontra em uma situação em que há risco de lesões corporais. Antes de trabalhar com qualquer equipamento, esteja ciente dos riscos que envolvem os circuitos elétricos e familiarize-se com as práticas padrão de prevenção de acidentes. Use o número da declaração fornecido ao final de cada aviso para localizar sua tradução nos avisos de segurança traduzidos que acompanham o dispositivo.

GUARDE ESTAS INSTRUÇÕES

Advarsel **VIGTIGE SIKKERHEDSANVISNINGER**

Dette advarselssymbol betyder fare. Du befinder dig i en situation med risiko for legemesbeskadigelse. Før du begynder arbejde på udstyr, skal du være opmærksom på de involverede risici, der er ved elektriske kredsløb, og du skal sætte dig ind i standardprocedurer til undgåelse af ulykker. Brug erklæringsnummeret efter hver advarsel for at finde oversættelsen i de oversatte advarsler, der fulgte med denne enhed.

GEM DISSE ANVISNINGER

تحذير **إرشادات الأمان الهامة**

يوضح رمز التحذير هذا وجود خطر. وهذا يعني أنك متواجد في مكان قد ينتج عنه التعرض لإصابات. قبل بدء العمل، احذر مخاطر التعرض للصدمات الكهربائية وكن على علم بالإجراءات القياسية للحيلولة دون وقوع أي حوادث. استخدم رقم البيان الموجود في أخر كل تحذير لتحديد مكان ترجمته داخل تحذيرات الأمان المترجمة التي تأتي مع الجهاز.

قم بحفظ هذه الإرشادات

Upozorenje **VAŽNE SIGURNOSNE NAPOMENE**

Ovaj simbol upozorenja predstavlja opasnost. Nalazite se u situaciji koja može prouzročiti tjelesne ozljede. Prije rada s bilo kojim uređajem, morate razumjeti opasnosti vezane uz električne sklopove, te biti upoznati sa standardnim načinima izbjegavanja nesreća. U prevedenim sigurnosnim upozorenjima, priloženima uz uređaj, možete prema broju koji se nalazi uz pojedino upozorenje pronaći i njegov prijevod.

SAČUVAJTE OVE UPUTE

Upozornění **DŮLEŽITÉ BEZPEČNOSTNÍ POKYNY**

Tento upozorňující symbol označuje nebezpečí. Jste v situaci, která by mohla způsobit nebezpečí úrazu. Před prací na jakémkoliv vybavení si uvědomte nebezpečí související s elektrickými obvody a seznamte se se standardními opatřeními pro předcházení úrazům. Podle čísla na konci každého upozornění vyhledejte jeho překlad v přeložených bezpečnostních upozorněních, která jsou přiložena k zařízení.

USCHOVEJTE TYTO POKYNY

Προειδοποίηση

ΣΗΜΑΝΤΙΚΕΣ ΟΔΗΓΙΕΣ ΑΣΦΑΛΕΙΑΣ

Αυτό το προειδοποιητικό σύμβολο σημαίνει κίνδυνο. Βρίσκεστε σε κατάσταση που μπορεί να προκαλέσει τραυματισμό. Πριν εργαστείτε σε οποιοδήποτε εξοπλισμό, να έχετε υπόψη σας τους κινδύνους που σχετίζονται με τα ηλεκτρικά κυκλώματα και να έχετε εξοικειωθεί με τις συνήθεις πρακτικές για την αποφυγή ατυχημάτων. Χρησιμοποιήστε τον αριθμό δήλωσης που παρέχεται στο τέλος κάθε προειδοποίησης, για να εντοπίσετε τη μετάφρασή της στις μεταφρασμένες προειδοποιήσεις ασφαλείας που συνοδεύουν τη συσκευή.

ΦΥΛΑΞΤΕ ΑΥΤΕΣ ΤΙΣ ΟΔΗΓΙΕΣ

אזהרה

הוראות בטיחות חשובות

סימן אזהרה זה מסמל סכנה. אתה נמצא במצב העלול לגרום לפציעה. לפני שתעבוד עם ציוד כלשהו, עליך להיות מודע לסכנות הכרוכות במעגלים חשמליים ולהכיר את הנהלים המקובלים למניעת תאונות. השתמש במספר ההוראה המסופק בסופה של כל אזהרה כד לאתר את התרגום באזהרות הבטיחות המתורגמות שמצורפות להתקן.

שמור הוראות אלה

Opomena

ВАЖНИ БЕЗБЕДНОСНИ НАПАТСТВИЈА
Симболот за предупредување значи опасност. Се наоѓате во ситуација што може да предизвика телесни повреди. Пред да работите со опремата, бидете свесни за ризикот што постои кај електричните кола и треба да ги познавате стандардните постапки за спречување на несреќни случаи. Искористете го бројот на изјавата што се наоѓа на крајот на секое предупредување за да го најдете неговиот период во преведените безбедносни предупредувања што се испорачани со уредот.
ЧУВАЈТЕ ГИ ОВИЕ НАПАТСТВИЈА

Ostrzeżenie

WAŻNE INSTRUKCJE DOTYCZĄCE BEZPIECZEŃSTWA

Ten symbol ostrzeżenia oznacza niebezpieczeństwo. Zachodzi sytuacja, która może powodować obrażenia ciała. Przed przystąpieniem do prac przy urządzeniach należy zapoznać się z zagrożeniami związanymi z układami elektrycznymi oraz ze standardowymi środkami zapobiegania wypadkom. Na końcu każdego ostrzeżenia podano numer, na podstawie którego można odszukać tłumaczenie tego ostrzeżenia w dołączonym do urządzenia dokumencie z tłumaczeniami ostrzeżeń.

NINIEJSZE INSTRUKCJE NALEŻY ZACHOWAĆ

Obtaining Optical Networking Information

This section contains information that is specific to optical networking products. For information that pertains to all of Cisco, refer to the Obtaining Documentation and Submitting a Service Request section.

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 15454 system. It also includes translations of the safety warnings that appear in the ONS 15454 system documentation.

Cisco Optical Networking Product Documentation CD-ROM

Optical networking-related documentation, including Cisco ONS 15xxx 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.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation,* which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

CHAPTER 1

ML-Series Card Overview

This chapter provides an overview of the ML1000-2, ML100T-12, ML100X-8 and ML-MR-10 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

The ML-Series cards are independent Gigabit Ethernet (ML1000-2) or Fast Ethernet (ML100T-12 and ML100X-8) Layer 3 switches that process up to 5.7 million packets per second (Mpps). The ML-Series cards are integrated into the ONS 15454 SONET or the ONS 15454 SDH.

The Cisco IOS command-line interface (CLI) is the primary user interface for the ML-Series card. Most configuration for the card, such as Ethernet port, bridging, and VLAN, can be done only through the Cisco IOS CLI.

However, CTC, the ONS 15454 SONET/SDH graphical user interface (GUI), also supports the ML-Series card. SONET/SDH circuits cannot be provisioned through Cisco IOS, but must be configured through CTC or Transaction Language One (TL1). CTC offers ML-Series card status information, SONET/SDH alarm management, Cisco IOS Telnet session initialization, Cisco IOS configuration file management, provisioning, inventory, and other standard functions.

The ML100T-12 features twelve RJ-45 interfaces, and the ML100X-8 and ML1000-2 features two Small Form-factor Pluggable (SFP) slots supporting short wavelength (SX) and long wavelength (LX) optical modules. All three cards use the same hardware and software base and offer similar feature sets. For detailed card specifications, refer to the "Ethernet Cards" chapter of the *Cisco ONS 15454 Reference Manual* or the *Cisco ONS 15454 SDH Reference Manual*.

The ML-MR-10 card is a Multi-Rate Layer 2 mapping module that provides 1:1 mapping of Ethernet ports to virtual circuits. The ML-MR-10 has ten SFP connectors that support IEEE 802.3 compliant Ethernet ports at the ingress offering 10 Mbps, 100 Mbps, or 1000 Mbps rates.SFP modules are offered as separate orderable products for flexibility.

The ML-Series cards features two virtual packet-over-SONET/SDH (POS) ports, which function in a manner similar to OC-N/STM-N card ports. The SONET/SDH circuits are provisioned through CTC in the same manner as standard OC-N/STM-N card circuits. The ML-Series POS ports support virtual concatenation (VCAT) of SONET/SDH circuits and a software link capacity adjustment scheme (SW-LCAS). The ML-MR-10 supports only framed generic framing procedure (GFP-F) encapsulation for SONET.

ML-Series Feature List

The ML100T-12, ML100X-8, ML1000-2 and ML-MR-10 cards have the following features:

- Layer 1 data features:
  - 10/100BASE-TX half-duplex and full-duplex data transmission
  - 1000BASE-SX, 1000BASE-LX full-duplex data transmission
  - IEEE 802.3z (Gigabit Ethernet) and IEEE 802.3x (Fast Ethernet) Flow Control
  - IEEE 802.3ad Link Aggregation Control Protocol
- SONET/SDH features:
  - High-level data link control (HDLC) or frame-mapped generic framing procedure (GFP-F) framing mechanism for POS (ML-MR-10 supports only framed generic framing procedure (GFP-F) encapsulation for SONET)
  - Two POS virtual ports (ML100T-12, ML100X-8, ML1000-2)
  - LEX, Cisco HDLC, or Point-to-Point Protocol/Bridging Control Protocol (PPP/BCP) encapsulation for POS (ML100T-12, ML100X-8, ML1000-2)
  - VCAT with SW-LCAS (ML100T-12, ML100X-8, ML1000-2)
  - G-Series card and ONS 15327 E-Series card compatible (with LEX encapsulation only), (ML100T-12, ML100X-8, ML1000-2)
- Layer 2 bridging features (ML100T-12, ML100X-8, ML1000-2):
  - Transparent bridging
  - MAC address learning, aging, and switching by hardware
  - Protocol tunneling
  - Multiple Spanning Tree (MST) protocol tunneling
  - 255 active bridge group maximum
  - 60,000 MAC address maximum per card and 8,000 MAC address maximum per bridge group
  - Integrated routing and bridging (IRB)
  - IEEE 802.1P/Q-based VLAN trunking
  - IEEE 802.1Q VLAN tunneling
  - IEEE 802.1D Spanning Tree Protocol (STP) and IEEE 802.1W Rapid Spanning Tree Protocol (RSTP)
  - IEEE 802.1D STP instance per bridge group
  - Ethernet over Multiprotocol Label Switching (EoMPLS)
  - EoMPLS traffic engineering (EoMPLS-TE) with RSVP
  - VLAN-transparent and VLAN-specific services (Ethernet Relay Multipoint Service [ERMS])
- RPR-IEEE data path features supported:
  - Bridging is supported, as specified in the IEEE 802.17b spatially aware sublayer amendment.

  - Shortest path forwarding through topology discovery is supported.
  - Addressing is supported, including unicast, multicast, and simple broadcast data transfers.
  - Bidirectional multicast frames flood around the ring using both east and west ringlets.
  - The time to live (TTL) of the multicast frames is set to the equidistant span in a closed ring and the failed span in an open ring.
- RPR-IEEE service qualities supported:
  - Per-service-quality flow-control protocols regulate traffic introduced by clients.
  - Class A allocated or guaranteed bandwidth has low circumference-independent jitter.
  - Class B allocated or guaranteed bandwidth has bounded circumference-dependent jitter. This class allows for transmissions of excess information rate (EIR) bandwidths (with class C properties).
  - Class C provides best-effort services.
- RPR-IEEE design strategies increase effective bandwidths beyond those of a broadcast ring:
  - Clockwise and counterclockwise transmissions can be concurrent.
  - Bandwidths can be reallocated on nonoverlapping segments.
  - Bandwidth reclamation. Unused bandwidths can be reclaimed by opportunistic services.
  - Spatial bandwidth reuse. Opportunistic bandwidths are reused on nonoverlapping segments.
  - Temporal bandwidth reuse. Unused opportunistic bandwidth can be consumed by others.
- RPR-IEEE fairness features ensure proper partitioning of opportunistic traffic:
  - Weighted fairness allows a weighted fair access to available ring capacity.
  - Aggressive fairness is supported.
  - Single Choke Fairness Supports generation, termination, and processing of Single Choke Fairness frames on both spans.
- RPR-IEEE plug-and-play automatic topology discovery and advertisement of station capabilities allow systems to become operational without manual intervention.
- RPR-IEEE multiple features support robust frame transmissions:
  - Service restoration time is less than 60 milliseconds after a station or link failure.
  - Queue and shaper specifications avoid frame loss in normal operation.
  - Fully distributed control architecture eliminates single points of failure.
  - Operations, administration, and maintenance support service provider environments.
- RPR-IEEE non-supported features:
  - EoMPLS is not supported.
  - IP forwarding is not supported.
  - Wrapping, the optional IEEE 802.17b protection scheme, is not supported. Steering, the protection scheme mandated by the standard, is supported.
  - Layer 3 routing is not supported.
  - GFP-CSF is not supported. The ML and ML-MR cards do not generate the GFP-CSF Indication on any of the RPR spans. The behavior on receiving a GFP-CSF indication on the RPR interface is undefined.
- Cisco Proprietary RPR (ML100T-12, ML100X-8, ML1000-2):

- Ethernet frame check sequence (FCS) preservation for customers
- Cyclic redundancy check (CRC) error alarm generation
- FCS detection and threshold configuration
- Shortest path determination
- Keep alives

- 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
  - IEEE 802.1Q trunking
  - 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
  - IEEE 802.1Q trunking
  - Auto-negotiation with Remote Fault Indication (RFI)
- POS channel (ML100T-12, ML100X-8, ML1000-2):
  - Bundling the two POS ports
  - LEX encapsulation only
  - IRB
  - IEEE 802.1Q trunking
- Layer 3 routing, switching, and forwarding (ML100T-12, ML100X-8, ML1000-2):
  - Default routes
  - IP unicast and multicast forwarding
  - 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
  - Reverse Path Forwarding (RPF) multicast (not RPF unicast)
  - 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
  - IRB routing mode support
- Supported routing protocols (ML100T-12, ML100X-8, ML1000-2):
  - 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)
- Quality of service (QoS) features:
  - Multicast priority queuing classes
  - Service level agreements (SLAs) with 1-Mbps granularity
  - Input policing
  - Guaranteed bandwidth (weighted round-robin [WDRR] plus strict priority scheduling)
  - Low latency queuing support for unicast Voice-over-IP (VoIP)
  - Class of service (CoS) based on Layer 2 priority, VLAN ID, Layer 3 Type of Service/DiffServ Code Point (TOS/DSCP), and port
  - CoS-based packet statistics
- Ethernet Virtual Circuits (ML-MR-10)
  - Point-to-Point topology (UNI to UNI)
  - Attribute Discovery Frames (ATD) for VLAN mapping
  - VLAN ID (IEEE 801Q tag)
  - Ethernet Frame Check Sequence (FCS)
- Security features:
  - Cisco IOS login enhancements
  - Secure Shell connection (SSH Version 2)
  - Disabled console port
  - Authentication, Authorization, and Accounting/Remote Authentication Dial-In User Service (AAA/RADIUS) stand alone mode
  - AAA/RADIUS relay mode
- Additional protocols:
  - 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)
- Management features:
  - Cisco IOS

  - CTC
  - CTM
  - Remote monitoring (RMON)
  - Simple Network Management Protocol (SNMP)
  - Transaction Language 1 (TL1)
  - (Not applicable to ML-MR-10 cards) Simultaneous performance monitoring (PM) counter clearing in Cisco IOS, CTC, and TL1
- System features:
  - Automatic field programmable gate array (FPGA) Upgrade
  - Network Equipment Building Systems 3 (NEBS3) compliant
  - Multiple microcode images
  - Version up to independently upgrade individual ML-MR-10 cards
- CTC features:
  - Framing Mode Provisioning
  - Standard STS/STM and VCAT circuit provisioning for POS virtual ports
  - SONET/SDH alarm reporting for path alarms and other ML-Series card specific alarms, including RPR-WRAP
  - Raw port statistics
  - Standard inventory and card management functions
  - J1 path trace
  - Cisco IOS CLI Telnet sessions from CTC
  - Cisco IOS startup configuration file management from CTC

CHAPTER 2

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 IOS configuration file or to open a Cisco IOS command-line interface (CLI) session. See Chapter 3, "Initial Configuration."

This chapter contains the following major sections:

Displaying ML-Series POS And Ethernet Statistics on CTC

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

The Ethernet statistics window lists Ethernet port-level statistics. It is similar in appearance to the POS statistics window. 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. Table 2-1 describes the buttons in the POS Ports and Ether Ports window.

A different set of statistics appears for the ML-Series card depending on whether the card is using HDLC or GFP-F framing. For definitions of ML-Series card statistics, refer to the "Performance Monitoring" chapter of the *Cisco ONS 15454 Reference Manual* or the *Cisco ONS 15454 SDH Reference Manual.*

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

Button	Description
Refresh	Manually refreshes the statistics.
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 counters displaying the change from the temporary baseline are displayed by this CTC client. 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.
Auto-Refresh	Sets a time interval for the automatic refresh of statistics.

Displaying ML-Series Ethernet Ports Provisioning Information on CTC

The Ethernet port provisioning window displays the provisioning status of the Ethernet ports. Click the **Provisioning > Ether Ports** tabs to display this window.

The user must configure ML-Series ports using the Cisco IOS CL; however, the following fields can be provisioned using CTC: Port Name, Pre-Service Alarm Suppression (PSAS), and Soak Time. Click the Port Name field to assign a name to the port. For more information about provisioning these fields, refer to the "Change Card Settings" chapter in the *Cisco ONS 15454 Procedure Guide*.

"Auto" in a column indicates the port is set to autonegotiate capabilities with the attached link partner.

Not all ML-Series cards display all columns. Table 2-2 details the information displayed under the Provisioning > Ether Ports tab:

Table 2-2 CTC Display of Ethernet Port Provisioning Status

Column	Description	ML1000-2	ML100T-12	ML100X-8
Port	The fixed number identifier for the specific port.	0 or 1	0-11	0-7
Port Name	Configurable 12-character alphanumeric identifier for the port.	User specific	User specific	User specific
Admin State	Configured port state, which is administratively active or inactive.	UP and DOWN	UP and DOWN	UP and DOWN
Link State	Status between signaling points at port and attached device.	UP and DOWN	UP and DOWN	UP and DOWN
PSAS	A check indicates alarm suppression is set on the port for the time designated in the Soak Time column.			
Soak Time	Desired soak time in hours and minutes. Use this column when you have checked PSAS to suppress alarms. Once the port detects a signal, the countdown begins for the designated soak time. Soak time hours can be set from 0 to 48. Soak time minutes can be set from 0 to 45 in 15 minute increments.			
MTU	(Maximum Transmission Unit) Largest acceptable packet size configured for that port.	Default value is 1500	Default value is 1500	Default value is 1500

Table 2-2 CTC Display of Ethernet Port Provisioning Status (continued)

Column	Description	ML1000-2	ML100T-12	ML100X-8
Speed	Ethernet port transmission speed.	—	Auto, 10Mbps, or 100Mbps	100Mbps
Duplex	Setting of the duplex mode for the port.	—	Auto, Full, or Half	Full
Flow Control	Flow control mode negotiated with peer device. These values are displayed but not configurable in CTC.	Asymmetrical, Symmetrical or None	Symmetrical or None	Symmetrical or None
Optics	Small form-factor pluggable (SFP) physical media type.	Unplugged, 1000 SX, or 1000 LX	—	Unplugged, 100 FX, or 100 LX

Note The 100 FX value in the Optics column of the ML100X-8 represent the short wavelength (SX) SFP.

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 does not match, unless the same name is used to configure the port name in both CTC and Cisco IOS.

Displaying ML-Series POS Ports Provisioning Information on CTC

The POS ports provisioning window displays the provisioning status of the card's POS ports. Click the **Provisioning > POS Ports** tabs to display this window.

The user must configure ML-Series ports using the Cisco IOS CLI; however, the following fields can be provisioned using CTC: Port Name, Pre-Service Alarm Suppression (PSAS), and Soak Time. Click the Port Name field to assign a name to the port. For more information about provisioning these fields, refer to the "Change Card Settings" chapter in the *Cisco ONS 15454 Procedure Guide*.

Table 2-3 details the information displayed under the Provisioning > POS Ports tab.

Table 2-3 CTC Display of POS Port Provisioning Status

Column	Description
Port	The fixed number identifier for the specific port.
Port Name	Configurable 12-character alphanumeric identifier for the port.
Admin State	Configured port state, which is administratively 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 port and attached device. Possible values are UP and DOWN.

Table 2-3 CTC Display of POS Port Provisioning Status

Column	Description
PSAS	A check indicates alarm suppression is set on the port for the time designated in the Soak Time column.
Soak Time	Desired soak time in hours and minutes. Use this column when you have checked PSAS to suppress alarms. Once the port detects a signal, the countdown begins for the designated soak time. Soak time hours can be set from 0 to 48. Soak time minutes can be set from 0 to 45 in 15 minute increments.
MTU	The maximum transfer unit, which is the largest acceptable packet size configured for that port. The maximum setting is 9000. The default size is 1500 for the G-Series card compatible encapsulation (LEX) and 4470 for Cisco HDLC and Point-to-Point Protocol/Bridging Control Protocol (PPP/BCP) encapsulation.
Framing Type	HDLC or frame-mapped generic framing procedure (GFP-F) framing type shows the POS framing mechanism being employed on the port.

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 does not match, unless the same name is used to configure the port name in both CTC and Cisco IOS.

Provisioning Card Mode

The card mode provisioning window shows the mode currently configured on the ML-Series card and allows the user to change it to either HDLC, GFP-F, or 802.17 RPR. For more information on HDLC or GFP-F, see Chapter 22, "POS on ONS Ethernet Cards."

The user may also pre-provision the card mode of an ML-Series card before the card is physically installed. The ML-Series card will then boot up into the pre-provisioned mode. If the correct microcode image is not already loaded, setting the card mode to 802.17 will automatically download and enable the correct microcode image for IEEE compliant 802.17b.

Caution The ML-Series card reboots after the card mode is changed.

Click the **Provisioning** > **Card** tabs to display this window. Use the Mode drop-down list and then click **Apply** to provision the card mode type. Click **Yes** at the Reset Card dialog box that appears.

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 the *Cisco ONS 15454 SDH Troubleshooting Guide* for detailed information.

To view the window, click the **Provisioning > Alarm Profiles** tabs for the Ethernet and POS port alarm profile information.

Displaying the FPGA Information

CTC displays information for the field programmable gate array (FPGA) on the ML-Series card. Click the **Maintenance > Info** tabs to display this window.

The FPGA on the ML100T-12, ML100X-8 and ML1000-2 provides the interface and buffering between the card's network processor and the SONET/SDH cross-connect. FPGA Image Version 3.x supports HDLC framing, and FPGA Image Version 4.x supports GFP-F Framing. Both images support virtual concatenation (VCAT). In Release 5.0 and later, the correct FPGA is automatically loaded when the framing mode is changed by the user.

Note ML-Series cards manufactured prior to Software Release 4.6 need an updated version of the FPGA to support VCAT.

Caution Do not attempt to use current FPGA images with an earlier CTC software release.

Provisioning SONET/SDH Circuits

CTC provisions and edits STS/STM level circuits for the two virtual SONET/SDH ports of the ML-Series card in the same manner as it provisions other ONS 15454 SONET/SDH OC-N cards. The ONS 15454 ML-Series card supports both contiguous concatenation (CCAT) and virtual concatenation (VCAT) circuits.

For step-by-step instructions to configure an ML-Series card SONET CCAT or VCAT circuit, refer to the "Create Circuits and VT Tunnels" chapter of the *Cisco ONS 15454 Procedure Guide.* For step-by-step instructions to configure an ML-Series card SDH CCAT or VCAT circuit, refer to the "Create Circuits and Tunnels" chapter of the *Cisco ONS 15454 SDH Procedure Guide*. For more general information on VCAT circuits, refer to the "Circuits and Tunnels" chapter of the *Cisco ONS 15454 Reference Manual* or the *Cisco ONS 15454 SDH Reference Manual.*

J1 Path Trace

The J1 Path Trace is a repeated, fixed-length string comprised of 64 consecutive J1 bytes. You can use the string to monitor interruptions or changes to SONET/SDH circuit traffic. For information on J1 Path Trace, refer to the *Cisco ONS 15454 Reference Manual* or the *Cisco ONS 15454 SDH Reference Manual.*

CHAPTER 3

Initial Configuration

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

Hardware Installation

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

1. Install the ML-Series card into the ONS 15454 SONET/SDH. See the "Install Cards and Fiber-Optic Cable" chapter in the *Cisco ONS 15454 Procedure Guide* or the *Cisco ONS 15454 SDH Procedure Guide* for information.
2. Connect the cables to the front ports of the ML-Series card.
3. (Optional) Connect the console terminal to the ML-Series card.

Note A NO-CONFIG condition is reported in the Cisco Transport Controller (CTC) under the Alarms tab 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 3-7 for information on loading or creating the file.

Cisco IOS on the ML-Series Card

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 TCC2/TCC2P card. During a hard reset, when a card is physically removed and reinserted or power is otherwise lost to the card, the Cisco IOS software image is downloaded from the flash memory of the TCC2/TCC2P 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 the Cisco IOS command line interface (CLI) command **reload**, the ML-Series card checks its cache for a Cisco IOS image. If a valid and current Cisco 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 Cisco IOS image from the TCC2/TCC2P. Caching the Cisco IOS image provides a significant time savings when a warm reset is performed.

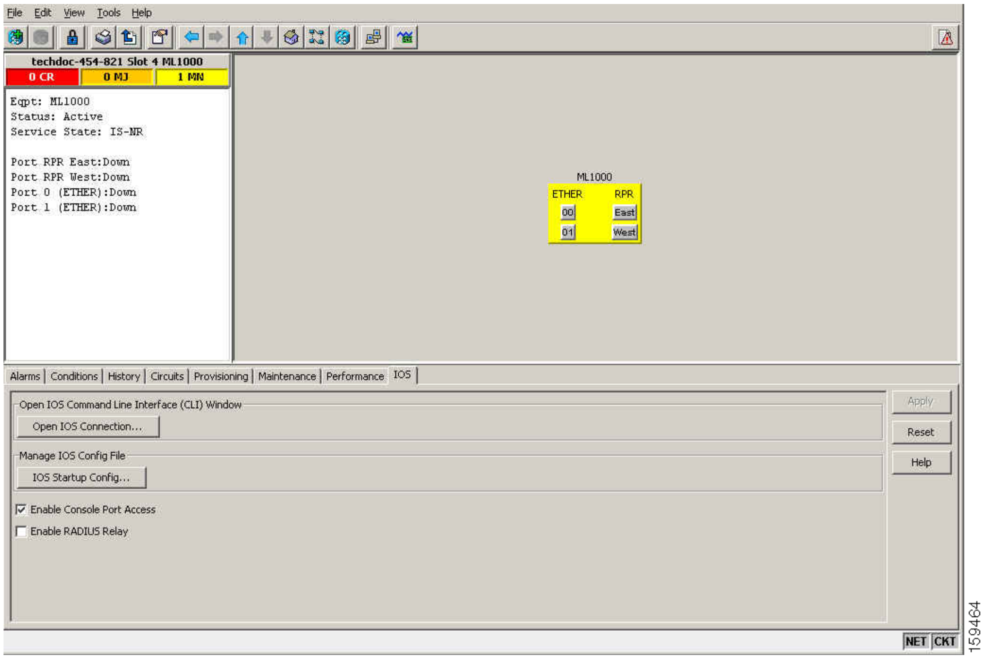
There are four ways to access the ML-Series card Cisco IOS configuration. The two out-of-band options are opening a Cisco IOS session on CTC and telnetting to the node IP address and slot number plus 2000. The two-in-band signalling options are telnetting to a configured management interface and directly 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. 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 A Cisco 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 3-7 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.

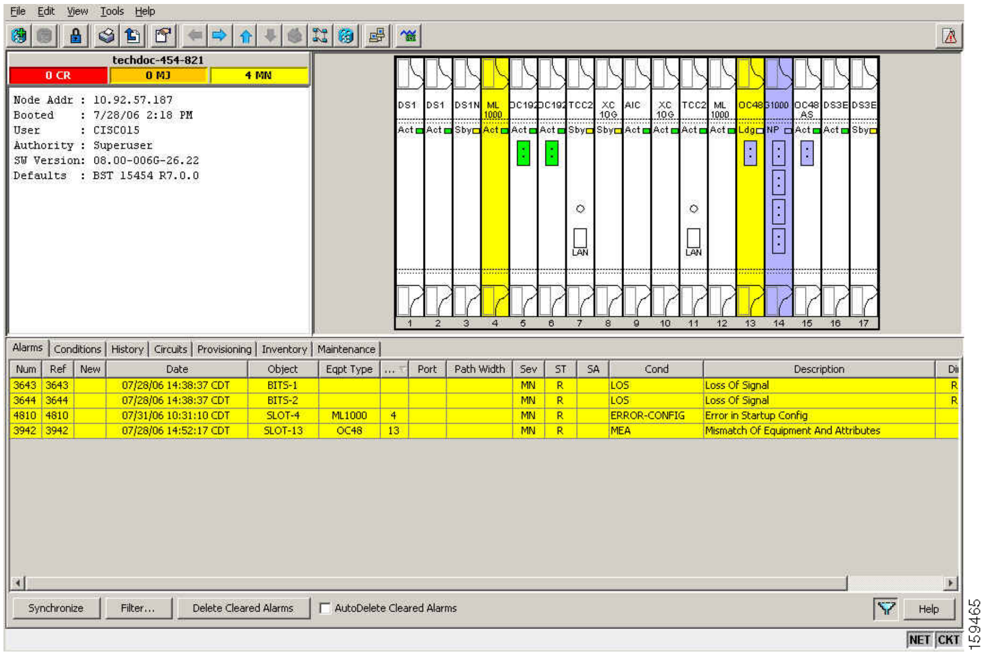
Note A Cisco IOS startup configuration file must be loaded and the ML-Series card must be installed and initialized prior to telnetting to the IP address and slot number plus 2000. See the "Startup Configuration File" section on page 3-7 for more information.

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



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 10.92.57.187 and Slot 13, you would enter or telnet `10.92.57.187 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 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 the "Configuring the Management Port" section on page 3-8.

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 CLI 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

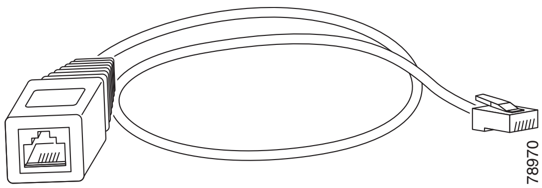


Table 3-1 shows the mapping of the RJ-11 pins to the RJ-45 pins.

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
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.

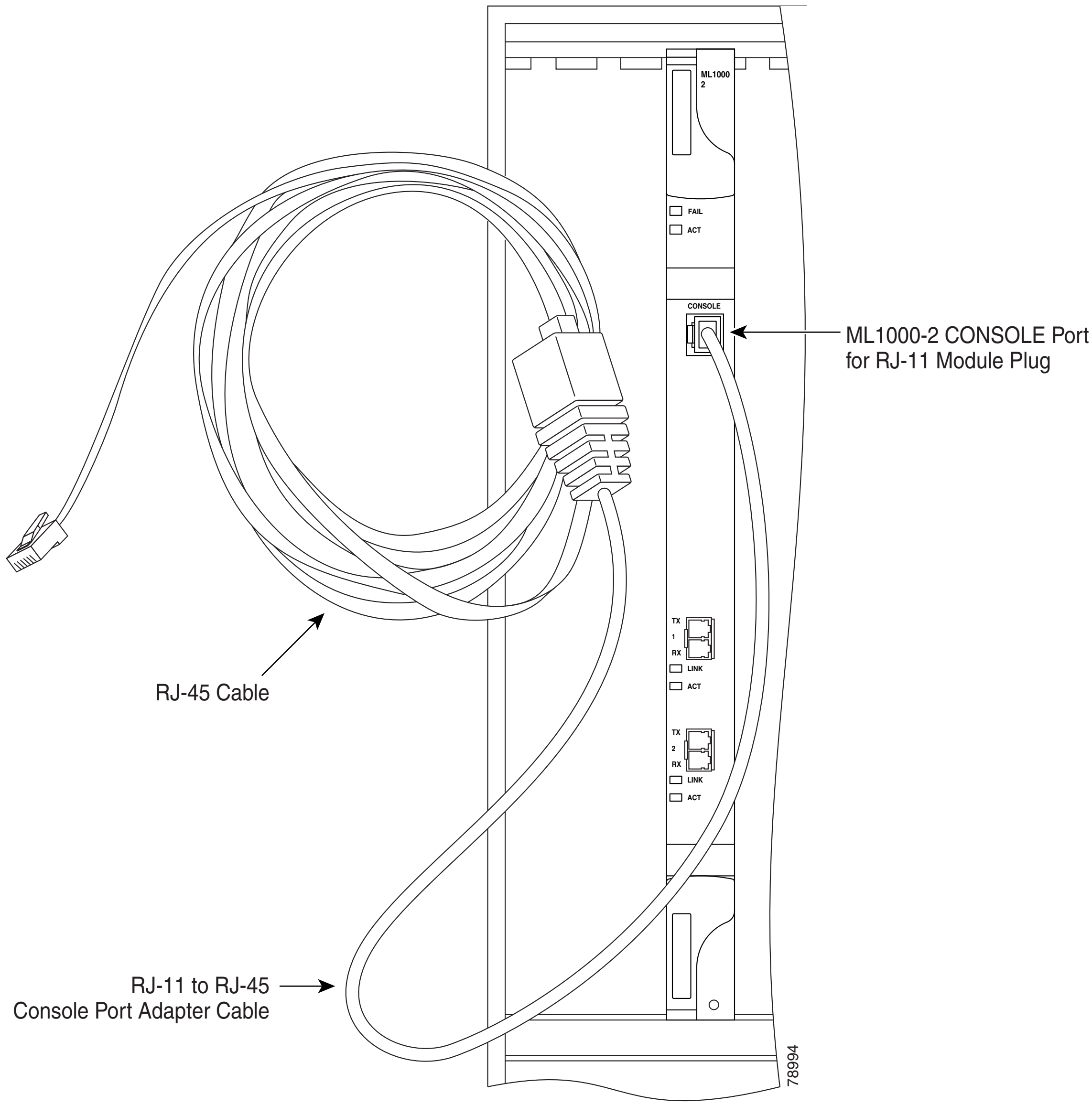
Step 1 Configure the data rate and character format of the PC or terminal to match these console port default settings:

- 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. For the ML100T-12 and ML100X-8, the console port 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 TCC2/TCC2P flash memory, then the card boots up to a default configuration. 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. A running configuration becomes a startup configuration file when saved with a **copy running-config startup-config** command.

It is not possible to establish a Telnet connection to the ML-Series card until a startup configuration file is loaded onto the ML-Series card. Access is available through the console port.

Caution The **copy running-config startup-config** command saves a startup configuration file to the flash memory on the ML-Series card. This operation is confirmed by the appearance of `[OK]` in the Cisco IOS CLI session. The startup configuration file is also saved to the ONS node's database restoration file after approximately 30 additional seconds.

Caution Accessing the read-only memory monitor mode (ROMMON) on the ML-Series card without the assistance of Cisco personnel is not recommended. This mode allows actions that can render the ML-Series card inoperable. The ML-Series card ROMMON is preconfigured to boot the correct Cisco IOS software image for the ML-Series card.

Caution The maximum size of the startup configuration file is 98356 bytes (characters).

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 saves 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 local, valid cached copy of Cisco IOS. It then either downloads the Cisco IOS software image from the TCC2/TCC2P or proceeds 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 "Configuring the Management Port" section on page 3-8.

Configuring the Management Port

Because there is no separate management port on ML-Series cards, any Fast Ethernet interface (0 to 11 on the ML100T-12 card and 0 to 7 on the ML100X-8), any Gigabit Ethernet interface (0 to 1 on the ML1000-2 card), or any POS interface (0 to 1 on any ML-Series card) can be configured as a management port. For the packet over SONET (POS) interface to exist, a synchronous transport signal (STS) or synchronous transport module (STM) circuit must first be created through CTC or translation language 1 (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 CLI through 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 **config t**. The Router(config)# prompt indicates that you are in global configuration mode.
Step 3	`Router(config)#` **`enable password`** *`password`*	Sets the enable password. See the "Passwords" section on page 3-8.
Step 4	`Router(config)#` **`enable secret`** *`password`*	Allows you to set an enable secret password. See the "Passwords" section on page 3-8. A user must enter the enable secret password to gain access to global configuration mode.
Step 5	`Router(config)#` **`interface`** *`type number`* `Router(config-if)#`	Activates interface configuration mode on the interface.

	Command	Purpose
Step 6	`Router(config-if)#` **`ip address`** *`ip-address subnetmask`*	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`** *`line-number`* `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`** *`password`*	Allows you to set 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`** *`name-string`*	Allows you to enter a system name. In this example, we set the hostname to "Router."
Step 3	*`name-string`*`(config)#` **`end`** *`name-string`*`#`	Returns to privileged EXEC mode.
Step 4	*`name-string`*`#` **`copy running-config startup-config`**	(Optional) Copies your configuration changes to NVRAM.

CTC and the Startup Configuration File

CTC allows a user to load the startup configuration file required by the ML-Series card. A Cisco-supplied sample Cisco IOS startup configuration file, named **Basic-IOS-startup-config.txt**, is available on the Cisco ONS 15454 SONET/SDH software CD. CISCO15 is the Cisco 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 TCC2/TCC2P 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 TCC2/TCC2P card flash, then the ML-Series card must be reset to use the Cisco IOS startup configuration file. The user can also 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.

Loading a Cisco IOS Startup Configuration File Through CTC

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.

Step 2 Click the **IOS Startup Config** button.

The config file dialog box appears.

Step 3 Click the **Local -> TCC** button.

Step 4 The sample Cisco 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 Cisco IOS startup config file and double-click the desired Cisco IOS startup config file.

Step 5 In the Are you sure? dialog box, click the **Yes** button.

The Directory and Filename fields on the configuration file dialog box update to reflect that the Cisco IOS startup config file is loaded onto the TCC2/TCC2P.

Step 6 Load the Cisco IOS startup config file from the TCC2/TCC2P 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 or card level CTC view and select **Reset Card**.

After the reset, the ML-Series card runs under the newly loaded Cisco IOS startup configuration file.

b. If the ML-Series card is not yet installed, installing the ML-Series card into the slot loads and runs the newly loaded Cisco 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 tab 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 Cisco IOS startup config file on the TCC2/TCC2P, but does not implement the Cisco IOS startup config on the ML-Series. See the "Database Restore of the Startup Configuration File" section on page 3-11 for additional information.

Database Restore of the Startup Configuration File

The ONS 15454 SONET/SDH includes a database restoration feature. Restoring the database will reconfigure a node and the installed line cards to the saved provisioning, except for the ML-Series card. The ML-Series card does not automatically restore the startup configuration file saved in the TCC2/TCC2P database.

A user can load the saved startup configuration file onto the ML-Series card in two ways. He can revert completely to the saved startup configuration and lose any additional provisioning in the unsaved running configuration, which is a restoration scheme similar to other ONS cards, or he can install the saved startup configuration file on top of the current running configuration, which is a merging restoration scheme used by many Cisco Catalyst devices.

To revert completely to the startup configuration file saved in the restored database, the user needs to reset the ML-Series card. Right-click the ML-Series card in CTC and choose **Reset** or use the Cisco IOS CLI **reload** command to reset the ML-Series card.

> **Caution** Resetting the ONS 15454 ML-Series card causes a loss of traffic and closes any Telnet sessions to the card.

To merge the saved startup configuration file with the running configuration, use the Cisco IOS CLI **copy startup-config running-config** command. This restoration scheme should only be used by experienced users with an understanding of the current running configuration and the Cisco IOS **copy** command. The **copy startup-config running-config** command will not reset the ML-Series card. The user also needs to use the Cisco IOS CLI **copy running-config startup-config** command to save the new merged running configuration to the startup configuration file.

Multiple Microcode Images

The primary packet processing and forwarding on the ML-Series card is done by the network processor, which is controlled by microcode. This microcode is a set of instructions (software) loaded into the network processor and executed at high speed. The network processor has limited microcode storage space.

Some of the ML-Series card features require significant amounts of microcode, and this additional microcode exceeds the storage capacity of the network processor. These features are added as new microcode images (separate microcode programs). The network processor can only hold one microcode image at a time, and changing the loaded microcode image requires resetting the network processor.

The user can choose from several microcode images for the ML-Series card. Table 3-2 compares the features available with the different microcode images.

Caution Configuring topology discovery or shortest path load balancing on an ML-Series card with the SW-RPR microcode image disables support for Cisco proprietary resilient packet ring (RPR) and dual RPR interconnect (DRPRI).

Table 3-2 Microcode Image Feature Comparison

Features	Base	Enhanced	EoMPLS[1]	SW-RPR	802.17
Packet Classification	Yes	Yes	Yes	Yes	Yes
Policing and Quality of Service (QoS)	Yes	Yes	Yes	Yes	Yes
Layer 2 Bridging	Yes	Yes	Yes	Yes	Yes
IP Unicast Switching	Yes	Yes	Yes	Yes	No
IP Fragmentation	Yes	No	No	No	No
IP Multicast Switching	Yes	No	No	No	No
EoMPLS	No	No	Yes	No	Future
Cisco Proprietary RPR Encapsulation	Yes	Yes	Yes	Yes	No
Cisco Proprietary RPR Resiliency Enhancements: • Cisco Proprietary RPR Keep Alive • Cisco Proprietary RPR CRC Threshold Configuration, Detection, and Wrap • Cisco Proprietary RPR Customer Ethernet FCS Preservation • Cisco Proprietary RPR CRC Error Alarm Generation • Cisco Proprietary RPR Shortest Path Determination and Topology Discovery	No	No	Yes	Yes	No
PPP/HDLC[2]/LEX[3]Encapsulation Support	Yes	Yes	Yes	No	No
IEEE 802.17b	No	No	No	No	Yes
Enhanced Performance Monitoring	No	Yes	No	Yes	Yes
Redundant Interconnect	No	No	Yes	Yes	Yes

1. Ethernet over multiprotocol label switching
2. high-level data link control
3. Ethernet over GFP-F according to ITU-T G.7041

Changing the Working Microcode Image

The user can change the microcode image using Cisco IOS CLI configuration and a reset of the ML-Series card. Using this configuration method, you can load any microcode image except 802.17. To automatically download and enable the 802.17 microcode image, use CTC to set the card mode to 802.17. For more information, see the "Provisioning Card Mode" section on page 2-4.

To configure a working microcode image, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# microcode {base | enhanced | fail | mpls | spr}`	Configures the ML-Series card with the selected microcode image: **base**—(Default) Enables base features only. Base features include Multicast routing and IP fragmentation. **enhanced**—Enables ERMS, enhanced packet statistics, and enhanced DRPRI. Disables multicast routing and IP fragmentation. **fail**—This command and feature are specific to ML-Series cards. In the event of a microcode failure, it configures the ML-Series card to save information to the flash memory and then reboot. The information is saved for use by the Cisco Technical Assistance Center (Cisco TAC). To contact Cisco TAC, see Appendix C, "Using Technical Support.". **mpls**—Enables MPLS. Disables IP multicast, IP fragmentation, and Ethernet relay multipoint service (ERMS) support. **spr**—Enables Cisco proprietary RPR encapsulation, Enhanced Packet Statistics, DRPRI, and Keepalives. Disables Multicast routing or IP fragmentation.
Step 2	`Router(config)# exit`	Exits global configuration mode.
Step 3	`Router# copy running-config startup-config`	Saves the configuration changes to flash memory. The running configuration file configured with the new microcode image choice must be saved as a startup configuration file for the ML-Series card to reboot with the new microcode image choice.
Step 4	`Router# reload`	Resets the ML-Series card and loads the new microcode image. **Caution** Resetting the ML-Series card causes a loss of traffic and closes any Telnet sessions to the card.
Step 5	`Router# show microcode`	Shows the microcode image currently loaded and the microcode image that loads when the ML-Series card resets.

Version Up Software Upgrade

The Version Up software upgrade feature allows users to independently upgrade ML-Series cards as part of an overall software upgrade process. With this feature enabled, the user first upgrades all the cards in the node that are not ML-Series cards, then in a second pass updates the ML-Series cards. Version Up is disabled by default.

The user can initiate individual upgrades for each ML-Series card or upgrade all the ML-Series cards at the same time. In the case of redundant ML-Series cards, individual upgrades allow time to verify the proper operation of the first card before the second card is upgraded. No ML-Series cards are updated until the user specifically requests it.

The user can perform a Version Up upgrade with CTC or Cisco Transport Manager (CTM). The Version Up feature is only supported on the ONS 15454 and SDH platforms. TL1 does not support the Version Up feature, and you cannot enter TL1 commands during the Version Up process.

Node and Card Behavior During Version Up

Between the upgrade of the non-ML-Series cards and the upgrade of the ML-Series cards, the node functions normally with regards to existing circuits but does not allow new provisioning or software downloads. Alarms still operate even with the ML-Series cards that are not yet upgraded.

The ML-Series card also continues to carry data traffic in the time span between the upgrade of the non-ML-Series cards and the upgrade of the ML-Series card, although this traffic drops when the ML-Series card resets to load the new software. You can telnet to the ML-Series card and configure the card using the Cisco IOS CLI, but the new configuration only exists in the running configuration file and cannot be saved to the startup configuration file.

During the Version Up upgrade, a SwMismatch condition appears for any cards running a different software version, even for non-ML-Series cards awaiting their turn to reset. When the card resets and loads the new software, the condition clears. The SwMismatch condition disappears on the ML-Series cards as they finish resetting and loading the new software. You can use the SwMismatch condition to keep track of ML-Series cards that still need upgrading. A SysBoot alarm is also raised during the upgrade. This alarm does not clear until all the ML-Series cards are upgraded.

Enabling and Completing Version Up

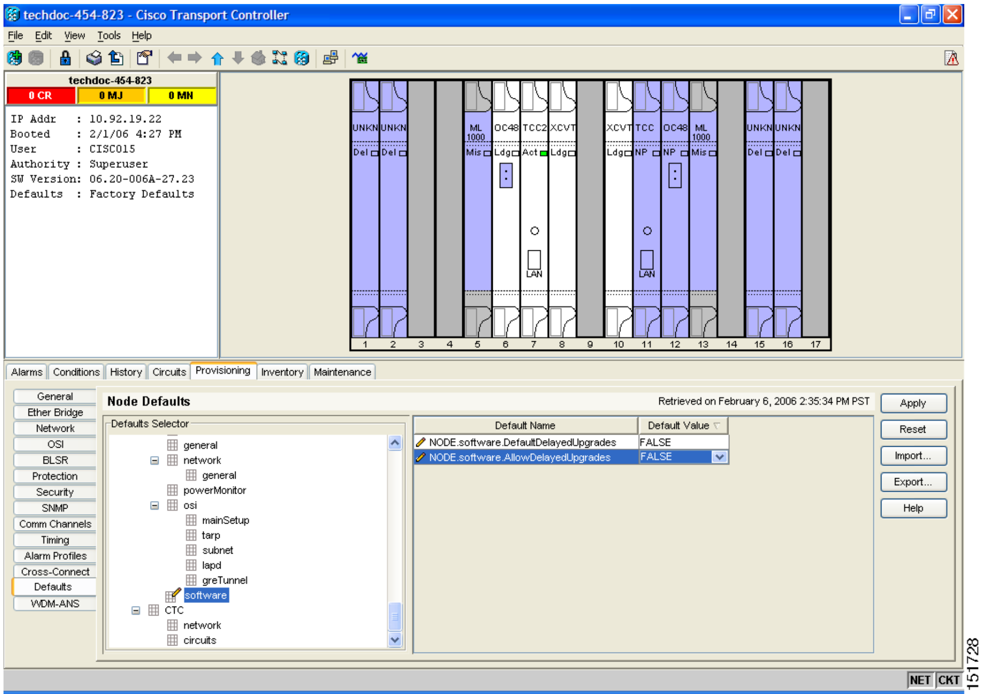
The default software upgrade behavior for the node is fully automatic. To enable Version Up, the NE defaults must be changed by a Superuser.

This procedure details enabling the Version Up feature through NE Defaults and completing the Version Up process.

Step 1 At the node view, click the **Provisioning > Defaults** tabs.

The Node Defaults window appears (Figure 3-5).

Figure 3-5 ***Node Defaults Delayed Upgrade Settings***



Step 2 In the Defaults Selector field, click **NODE** and then click **software.**

In the Default Name column, the *Node.Software.DefaultDelayedUpgrades* row and the *Node.Software.AllowDelayedUpgrades* row appear (Figure 3-5).

Step 3 Change the Default Value of the *Node.Software.AllowDelayedUpgrades* row to TRUE.

Step 4 Change the Default Value of the *Node.Software.DefaultDelayedUpgrades* row to TRUE.

Step 5 Click **Apply**.

The NE default is now set to enable Version Up.

Step 6 Begin the standard upgrade procedure for the node. Refer to the release-specific software upgrade document.

After clicking **Activate**, the Software Activation dialog box appears.

Step 7 Select the Delay automatic activation on the Software Activation dialog check box and click **OK**

Step 8 Accept the confirmation prompt to begin the Version Up activation.

Note Clearing the Delay automatic activation on the ML cards check box and clicking OK begins normal activation and upgrades all the cards in the node, including the ML-Series cards.

Step 9 After the new software load is activated on the node and all the non-ML-Series cards, you can activate this load on the ML-Series cards by resetting the ML-Series cards.

Caution Resetting the ML-Series card causes a loss of traffic and closes any Telnet sessions to the card.

To reset the ML-Series card through CTC, go to node view and click the ML-Series card to reveal a short cut menu, click **Reset Card**.

Step 10 After the ML-Series card reloads, verify that the correct software build is on the card using the Cisco IOS CLI privilege level **show version** command.

Example 3-1 shows a partial example of the **show version** command output with the Cisco IOS software version in bold.

Example 3-1 Output from show version Command

```
ML_Series# show version

Cisco IOS Software, ONS M-Series Software (DAYTONA-I7K91-M), Experimental Version
12.2(20050912:041138) [BLD-IOS_MARINER_MARINER_BF5_BUILD_6.amoayedi 105]
```

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 3-3 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 When a process makes unusually heavy demands on the CPU of the ML-Series card, it could 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.

Table 3-3 Cisco 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 POS port.	From global configuration mode, enter the **interface** *type number* command. For example, enter **interface fastethernet 0** for Fast Ethernet, **interface gigabitethernet 0** for Gigabit Ethernet interfaces, or **interface pos 0** for **POS** 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.

ROM monitor (ROMMON) mode 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 **config 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.

CHAPTER 4

Configuring Interfaces

This chapter describes basic interface configuration for the ML-Series card to help you get your ML-Series card up and running. Advanced packet-over-SONET/SDH (POS) interface configuration is covered in Chapter 5, "Configuring POS." 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:

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

General Interface Guidelines

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 that 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 that is composed 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, including the internal POS interfaces, has 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 that 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 twelve Fast Ethernet interfaces are Fast Ethernet 0 through 11. The ML100X-8 port IDs for the eight Fast Ethernet interfaces are Fast Ethernet 0 through 7. 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 card port IDs for the two POS interfaces are POS 0 and POS 1. You can use user-defined abbreviations such as f0 to configure the 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.

Basic Interface Configuration

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:

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 gigabitethernet number
```

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

The commands that 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
```

Basic Fast Ethernet, Gigabit Ethernet, and POS Interface Configuration

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 type number`	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 ip-address subnet-mask | bridge-group bridge-group-number}`	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 (TCC2/TCC2P) flash database.

Configuring the Fast Ethernet Interfaces for the ML100T-12

To configure the IP address or bridge-group number, speed, duplex, and flow control on an ML100T-12 Fast Ethernet interface, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface fastethernet number`	Activates interface configuration mode to configure the Fast Ethernet interface.
Step 2	`Router(config-if)# {ip address ip-address subnet-mask | bridge-group bridge-group-number}`	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] 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. In this case, the ML-Series card matches the speed and duplex mode of the partner node.
Step 4	`Router(config-if)# [no] duplex {full | half | auto}`	Sets full duplex, half duplex, or autonegotiate mode.

	Command	Purpose
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. ML-Series card Fast Ethernet port flow control is IEEE 802.3x compliant. **Note** Since Fast Ethernet ports support only symmetric flow control the **flowcontrol send** command controls both the receive and send flow control operations.
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 TCC2/TCC2P flash database.

Example 4-1 shows how to do the initial configuration of an ML100T-12 Fast Ethernet interface with an IP address and autonegotiation.

Example 4-1 Initial Configuration of a ML100T-12 Fast Ethernet Interface

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

Configuring the Fast Ethernet Interfaces for the ML100X-8

The ML100X-8 supports 100BASE-FX full-duplex data transmission. You cannot configure autonegotiation or speed on its Fast Ethernet interfaces. To configure the IP address or bridge-group number, or flow control on a Fast Ethernet interface, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface fastethernet`** *`number`*	Activates interface configuration mode to configure the Fast Ethernet interface.
Step 2	`Router(config-if)#` `{`**`ip address`** *`ip-address subnet-mask`* `|` **`bridge-group`** *`bridge-group-number`*`}`	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)#` **flowcontrol send** {**on** | **off** | **desired**}	(Optional) Sets the send flow control value for an interface. Flow control works only with port-level policing. ML-Series card Fast Ethernet port flow control is IEEE 802.3x compliant. **Note** Since Fast Ethernet ports support only symmetric flow control the **flowcontrol send** command controls both the receive and send flow control operations.
Step 4	`Router(config-if)#` **no shutdown**	Enables the interface by preventing it from shutting down.
Step 5	`Router(config)#` **end**	Returns to privileged EXEC mode.
Step 6	`Router#` **copy running-config startup-config**	(Optional) Saves your configuration changes to TCC2/TCC2P flash database.

Configuring the Gigabit Ethernet Interface for the ML1000-2

To configure IP address or bridge-group number, autonegotiation, and flow control on an ML1000-2 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** *number*	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.
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. ML-Series card Gigabit Ethernet port flow control is IEEE 802.3z compliant.
Step 5	`Router(config-if)#` **no shutdown**	Enables the interface by preventing it from shutting down.

	Command	Purpose
Step 6	`Router(config)# end`	Returns to privileged EXEC mode.
Step 7	`Router# copy running-config startup-config`	(Optional) Saves configuration changes to TCC2/TCC2P flash database.

Example 4-2 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
```

Configuring Gigabit Ethernet Remote Failure Indication (RFI)

Remote Failure Indication (RFI) is part of the IEEE 802.3z standard and is sent to exchange failure information as part of link negotiation. This feature improves communication between non-Cisco equipment and the ML1000-2. RFI is not on by default but can be turned on by the user. Disabling RFI is sometimes necessary when a non-Cisco piece of equipment does not support the IEEE 802.3z standard implementation of RFI.

RFI on the ML-Series card supports bidirectional RFI. When there is a a local fault on the ML-Series card, the ML-Series card will raise a local CARLOSS alarm and send its link partner an RFI. If an ML-Series card receives an RFI from its link partner, it raises the AUTONEG-RFI alarm and shuts down the Gigabit Ethernet port.

To enable RFI on a Gigabit Ethernet interface, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router (config)#` **`interface gigabitethernet`** *`number`*	Activates interface configuration mode to configure the Gigabit Ethernet interface.
Step 2	`Router(config-if)# [no] rfi auto`	Enables IEEE 802.3z standard RFI. The **no** form of the command disables RFI.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.
Step 4	`Router# copy running-config startup-config`	(Optional) Saves configuration changes to TCC2/TCC2P flash database.

Example 4-3 shows how to do an initial configuration of RFI on a Gigabit Ethernet interface.

Example 4-3 RFI Configuration of a Gigabit Ethernet Interface

```
Router(config)# interface gigabitethernet 0
Router(config-if)# rfi auto
Router(config-if)# end
```

Router# **copy running-config startup-config**

Monitoring and Verifying Gigabit Ethernet Remote Failure Indication (RFI)

After RFI is configured, you can verify that RFI is enabled by using the global command **show running configuration**. Example 4-4 shows the output from this command, and the "rfi auto" line under each of the Gigabit Ethernet port's output signifies RFI is enabled on these ports.

More specific RFI information is revealed with the global **show controller gigabit ethernet [0 | 1]** command**:**

- Example 4-5 shows the full output from this command on a near-end ML-Series card when no faults are detected at the near-end or far-end. The Remote Fault Indication is 00 or no error, and the Local Fault Indication is 00 or no error.
- Example 4-6 shows the partial output from this command on a near-end ML-Series card when a fault is detected at the near-end. The Remote Fault Indication is 00 or no error, but the Local Fault Indication is 01 or link error.
- Example 4-7 shows the partial output from this command on a far-end ML-Series card when a fault is detected at the near-end. The Remote Fault Indication is 01 or link error, and the Local Fault Indication is 00 or no error.

Note If the far-end link partner resets within approximately two minutes of the near-end ML-Series card sending an RFI signalling link error, the link partner will not display the RFI link error indication when back up.

Example 4-4 show run Command Output for RFI

```
Router# show running configuration
Building configuration...

Current configuration : 806 bytes
!
! No configuration change since last restart
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Interop-261-TOP-
!
boot-start-marker
boot-end-marker
!
enable password lab
!
clock timezone PST -8
clock summer-time PDT date Apr 2 2006 2:00 Oct 29 2006 2:00
ip subnet-zero
!
no mpls traffic-eng auto-bw timers frequency 0

interface GigabitEthernet0
 no ip address
 rfi auto
```

```
!
interface GigabitEthernet1
 no ip address
 rfi auto
```

Example 4-5 show controller Command Output for RFI on near-end card with no faults detected

```
Near_End# show controller gigabit ethernet 0
IF Name: GigabitEthernet0
Port Status UP
Port rxLosState Signal present
Remote Fault Indication 00  (no error)
Local Fault Indication 00  (no error)
Port 0 Gmac Loopback false
SFP EEPROM information
----------------------
0x0 : 03 04 07 00 00 00 02 12 00 01 01 01 0C 00 0A 64
0x10: 37 37 00 00 46 49 4E 49 53 41 52 20 43 4F 52 50
0x20: 2E 20 20 20 00 00 FFFFFF90 65 46 54 52 4A 2D 31 33 31
0x30: 39 2D 37 44 2D 43 53 43 00 00 00 00 05 1E 00 00

GBIC Type: GBIC_1000BASE_LH
Send Flow Control: Enabled (Port level policing required to send pause frames)
Receive Flow Control : Enabled
CRC-ALARM: FALSE

MAC registers:
GCR: 0x0         CMCR : 0x00000803 (Tx Enabled, Rx Enabled)

MII registers of External GMAC:
Control Register              (0x00): 0x1140 (Auto negotation Enabled)
Status Register               (0x01): 0x16D  (Link Status Up)
Auto Neg. Advt. Register      (0x04): 0x1A0  (Dir 1, Sym 1)
Auto Neg. Partner Ability Reg (0x05): 0x41A0 (Dir 1, Sym 1)
TR_IPG_TIME Register          (0x10): 0x7
PAUSE_TIME Register           (0x11): 0x100
PAUSE_SA1 Register            (0x13): 0x0
PAUSE_SA2 Register            (0x14): 0x0
PAUSE_SA3 Register            (0x15): 0x0
Pause Upper Threshold Reg.    (0x19): 0x80
Pause Lower Threshold Reg.    (0x1A): 0xFF
TX Full Threshold Register    (0x1B): 0x40
Memory Address Register       (0x1C): 0xF008
Sync Status Register          (0x1D): 0x40
Sys Status Register           (0x1E): 0x98
Sys Control Register          (0x1F): 0x14
Auto Neg Ctrl Register      (0xF004): 0x7
Rx Uinfo Registerter-GMAC   (0xF006): 0x0
RX control Register-GMAC    (0xF009): 0x3
RX Oversize Register-GMAC   (0xF00A): 0x5F4
Statistics control register (0xF008): 0x1

Counters :
MAC receive conters:
Bytes                    1952660
pkt64                    0
pkts64to127              0
pkts128to255             0
pkts256to511             5485
pkts512to1023            0
pkts1024to1518           0
pkts1519to1530           0
pkts_good_giants         0
```

```
pkts_error_giants         0
pkts_good_runts           0
pkts_error_runts          0
pkts_ucast                0
pkts_mcast                5485
pkts_bcast                0
Rx Sync Loss              0
Overruns                  0
FCS_errors                0
GMAC drop count           0
Symbol error              0
Rx Pause frames           0

MAC Transmit Counters
5d00h: %LINK-3-UPDOWN: Interface GigabitEthernet0, changed state to down
5d00h: %ETHERCHAN-5-MEMREMOVED: GigabitEthernet0 taken out of port-channel1
5d00h: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0, changed
 staBytes                     1952660
pkts64                    0
pkts65to127               0
pkts128to255              0
pkts256to511              5485
pkts512to1023             0
pkts1024to1518            0
pkts1519to1530            0
Good Giants               0
Unicast packets           0
Multicast packets         5485
Broadcast packets         0
FCS errors                0
Tx Pause frames           0
Ucode drops               0
```

Example 4-6 show controller Command Output for RFI on Near-end Card with Near-end Fault

```
Near_End# show controller gigabit ethernet 0
IF Name: GigabitEthernet0
Port Status DOWN
Port rxLosState No signal
Remote Fault Indication 00  (no error)
Local Fault Indication 01  (link error)
Port 0 Gmac Loopback false
```

Example 4-7 show controller Command Output for RFI on Far-end Card with Near-end Fault

```
Far_End# show controller gigabit ethernet 0
IF Name: GigabitEthernet0
Port Status DOWN
Port rxLosState Signal present
Remote Fault Indication 01  (link error)
Local Fault Indication 00  (no error)
Port 0 Gmac Loopback false
```

Configuring the POS Interfaces (ML100T-12, ML100X-8 and ML1000-2)

Encapsulation changes on POS ports are allowed only when the interface is in a manual shutdown (ADMIN_DOWN). For advanced POS interface configuration, see Chapter 5, "Configuring POS."

To configure the IP address, bridge group, or encapsulation for the POS interface, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface pos`** *`number`*	Activates interface configuration mode to configure the POS 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.
Step 3	`Router(config-if)#` **`shutdown`**	Manually shuts down the interface. Encapsulation changes on POS ports are allowed only when the interface is shut down (ADMIN_DOWN).
Step 4	`Router(config-if)#` **`encapsulation`** *`type`*	Sets the encapsulation type. Valid values are: • **hdlc**—Cisco HDLC • **lex**—(Default) LAN extension, special encapsulation for use with Cisco ONS Ethernet line cards • **ppp**—Point-to-Point Protocol
Step 5	`Router(config-if)#` **`no shutdown`**	Restarts the shutdown interface.
Step 6	`Router(config)#` **`end`**	Returns to privileged EXEC mode.
Step 7	`Router#` **`copy running-config startup-config`**	(Optional) Saves configuration changes to NVRAM.

CRC Threshold Configuration

You can configure a span shutdown when the ML-Series card receives CRC errors at a rate that exceeds the configured threshold and configured soak time. ML cards support CRC threshold configuration functionality on FE / GE / POS and RPR-IEEE interfaces. For configuration sample for RPR IEEE interfaces, see Chapter 18, "Configuring IEEE 802.17b Resilient Packet Ring."

To enable and configure the triggers for CRC errors on POS , perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	*`Router(config)#int pos0`* *`Router(config-if)#trigger crc-error threshold <threshold_value>`*	Sets the CRC threshold value. If the percentage of CRC errored frames received on this interface is greater than this value, we consider this interface as seeing excessive CRC. The valid values are 2, 3 and 4 indicating thresholds of 10e-2 (1%), 10e-3(0.1%) and 10e-4(.01%). The default value is 3.
Step 2	`Router(config-if)#no trigger crc-error threshold < threshold_value>`	Sets the threshold value back to the default value 3.

	Command	Purpose
Step 3	`Router(config)#int pos0` Router(config-if)#trigger crc-error delay <soak_time_in_minutes>	Sets the number of consecutive minutes for which excessive CRC errors should be seen to raise an excessive CRC indication. The valid values are from 3 minutes to 10 minutes. Default is 10minutes.
Step 4	Router(config-if)#no trigger crc-error delay <soak_time_in_minutes>	Sets the soak value back to the default of 10 minutes.
Step 5	`Router(config)#int pos0` `Router(config-if)#trigger crc-error action`	Enable trigger action. This configuration will bring the interface down on seeing CRC errors greater than configured <threshold value> for soak time period.
Step 6	`Router(config-if)#no trigger crc-error action`	Disables trigger action.

Monitoring Operations on the Fast Ethernet and Gigabit Ethernet Interfaces

To verify the settings after you have configured the interfaces, enter the **show interface** command. For additional information about monitoring the operations on POS interfaces, see the "Configuring POS" chapter.

Example 4-8 shows the output from the **show interface** command, which displays the status of the interface including port speed and duplex operation.

Example 4-8 show interface Command Output

```
Router# show interface fastEthernet 0
FastEthernet1 is administratively down, line protocol is down
Hardware is epif_port, address is 000d.bd5c.4c85 (bia 000d.bd5c.4c85)
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)
Auto-duplex, Auto Speed, 100BaseTX
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, 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
0 packets output, 0 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.

Example 4-9 shows the output from the **show controller** command, which shows statistics including initialization block information.

Example 4-9 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 interface** [*type number*] 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.

Example 4-10 shows output from the **show run interface** [*type number*] command, which includes information about the IP address or lack of IP address and the state of the interface.

Example 4-10 show run interface Command Output

```
daytona# show run interface FastEthernet 1
Building configuration...

Current configuration : 56 bytes
!
interface FastEthernet1
no ip address
shutdown
end
```

CHAPTER 5F

Configuring POS

This chapter describes advanced packet-over-SONET/SDH (POS) interface configuration for the ML-Series card. Basic POS interface configuration is included in Chapter 4, "Configuring Interfaces." For more information about the Cisco IOS commands used in this chapter, refer to the *Cisco IOS Command Reference* publication. POS operation on ONS Ethernet cards, including the ML-Series card, is described in Chapter 22, "POS on ONS Ethernet Cards."

This chapter contains the following major sections:

POS on the ML-Series Card

Ethernet and IP data packets need to be framed and encapsulated into SONET/SDH frames for transport across the SONET/SDH network. This framing and encapsulation process is known as POS and is done in the ML-Series card. Chapter 22, "POS on ONS Ethernet Cards," explains POS in greater detail.

The ML-Series card takes the standard Ethernet ports on the front of the card and the virtual POS ports and includes them all as switch ports. Under Cisco IOS, the POS port is an interface similar to the other Ethernet interfaces on the ML-Series card. It is usually used as a trunk port. Many standard Cisco IOS features, such as IEEE 802.1 Q VLAN configuration, are configured on the POS interface in the same manner as on a standard Ethernet interface. Other features and configurations are done strictly on the POS interface. The configuration of features limited to POS ports is shown in this chapter.

ML-Series SONET and SDH Circuit Sizes

SONET is an American National Standards Institute (ANSI) standard (T1.1051988) for optical digital transmission at hierarchical rates from 51.840 Mbps (STS-1) to 2.488 Gbps (STS-48) and greater. SDH is the international standard for optical digital transmission at hierarchical rates from 155.520 Mbps (STM-1) to 2.488 Gbps (STM-16) and greater.

Both SONET and SDH are based on a structure that has a basic frame and speed. The frame format used by SONET is the synchronous transport signal (STS), with STS-1 being the base level signal at 51.84 Mbps. A STS-1 frame can be carried in an OC-1 signal. The frame format used by SDH is the synchronous transport module (STM), with STM-1 being the base level signal at 155.52 Mbps. A STM-1 frame can be carried in an OC-3 signal.

Both SONET and SDH have a hierarchy of signaling speeds. Multiple lower level signals can be multiplexed together to form higher level signals. For example, three STS-1 signals can be multiplexed together to form a STS-3 signal, and four STM-1 signals can be multiplexed together to form a STM-4 signal.

SONET circuit sizes are defined as STS-n, where n is a multiple of 51.84 Mbps and n is equal to or greater than 1. SDH circuit sizes are defined as STM-n, where n is a multiple of 155.52 Mbps and n is equal to or greater than 0. Table 5-1 shows STS and STM line rate equivalents.

Table 5-1 SONET STS Circuit Capacity in Line Rate Mbps

SONET Circuit Size	SDH Circuit Size	Line Rate in Mbps
STS-1 (OC-1)	VC-3[1]	52 Mbps
STS-3c (OC-3)	STM-1 (VC4)	156 Mbps
STS-6c (OC-6)	STM-2 (VC4-2c)	311 Mbps
STS-9c (OC-9)	STM-3 (VC4-3c)	466 Mbps
STS-12c (OC-12)	STM-4 (VC4-4c)	622 Mbps
STS-24c (OC-24)	STM-8 (VC4-8c)	1244 Mbps (1.24 Gbps)

1. VC-3 circuit support requires an XC-VXx or XC-VCX-10G card to be installed.

For step-by-step instructions on configuring an ML-Series card SONET STS circuit, refer to the "Create Circuits and VT Tunnels" chapter of the *Cisco ONS 15454 Procedure Guide*. For step-by-step instructions on configuring an ML-Series card SDH STM circuit, refer to the "Create Circuits and Tunnels" chapter of the *Cisco ONS 15454 SDH Procedure Guide*.

VCAT

VCAT significantly improves the efficiency of data transport over SONET/SDH by grouping the synchronous payload envelopes (SPEs) of SONET/SDH frames in a nonconsecutive manner into VCAT groups. VCAT group circuit bandwidth is divided into smaller circuits called VCAT members. The individual members act as independent circuits.

Intermediate nodes treat the VCAT members as normal circuits that are independently routed and protected by the SONET/SDH network. At the terminating nodes, these member circuits are multiplexed into a contiguous stream of data. VCAT avoids the SONET/SDH bandwidth fragmentation problem and allows finer granularity for provisioning of bandwidth services.

The ONS 15454 SONET and ONS 15454 SDH ML-Series card VCAT circuits must also be routed over common fiber and be both bidirectional and symmetric. Only high order (HO) VCAT circuits are supported. The ML-Series card supports a maximum of two VCAT groups, with each group corresponding to one of the POS ports. Each VCAT group can contain two circuit members. A VCAT circuit originating from an ML-Series card must terminate on another ML-Series card or a CE-Series card. Table 5-2 shows supported VCAT circuit sizes for the ML-Series.

Caution Packet losses might occur when an optical fiber is reinserted or when a defect is cleared on members of the HW-LCAS split fiber routed circuits.

Table 5-2 VCAT Circuit Sizes Supported by ML100T-12, ML100X-8, and ML1000-2 Cards

SONET VCAT Circuit Size	SDH VCAT Circuit Size
STS-1-2v	VC-3-2v
STS-3c-2v	VC-4-2v
STS-12c-2v	VC-4-4c-2v

For step-by-step instructions on configuring an ML-Series card SONET VCAT circuit, refer to the "Create Circuits and VT Tunnels" chapter of the *Cisco ONS 15454 Procedure Guide*. For step-by-step instructions on configuring an ML-Series card SDH VCAT circuit, refer to the "Create Circuits and Tunnels" chapter of the *Cisco ONS 15454 SDH Procedure Guide*. For more general information on VCAT circuits, refer to the "Circuits and Tunnels" chapter of the *Cisco ONS 15454 Reference Manual* or the *Cisco ONS 15454 SDH Reference Manual*.

Note ML-Series card POS interfaces normally send an alarm for signal label mismatch failure in the ONS 15454 STS path overhead (PDI-P) to the far end when the POS link goes down or when RPR wraps. ML-Series card POS interfaces do not send PDI-P to the far-end when PDI-P is detected, when a remote defection indication alarm (RDI-P) is being sent to the far end, or when the only defects detected are generic framing procedure (GFP)-loss of frame delineation (LFD), GFP client signal fail (CSF), virtual concatenation (VCAT)-loss of multiframe (LOM), or VCAT-loss of sequence (SQM).

Note For nodes not connected by DCC (open ended nodes), VCAT must be configured through TL-1.

Note See section, "VCAT Circuit Provisioning Time Slot Limitations" of Chapter 21, CE-Series Ethernet Cards for information on VCAT Circuit Provisioning Time Slot Limitations.

SW-LCAS

A link capacity adjustment scheme (LCAS) increases VCAT flexibility by allowing the dynamic reconfiguration of VCAT groups without interrupting the operation of noninvolved members. Software link capacity adjustment scheme (SW-LCAS) is the software implementation of a LCAS-type feature. SW-LCAS differs from LCAS because it is not errorless and uses a different handshaking mechanism.

SW-LCAS on the ONS 15454 SONET/SDH ML-Series cards allows the automatic addition or removal of a VCAT group member in the event of a failure or recovery on a two-fiber bidirectional line switched ring (BLSR). The protection mechanism software operates based on ML-Series card link events. SW-LCAS allows service providers to configure VCAT member circuits on the ML-Series as protection channel access (PCA) circuits. This PCA traffic is dropped in the event of a protection switch, but is suitable for excess or noncommited traffic and can double the total available bandwidth on the circuit.

For step-by-step instructions on configuring SW-LCAS, refer to the "Create Circuits and VT Tunnels" chapter of the *Cisco ONS 15454 Procedure Guide* or the "Create Circuits and Tunnels" chapter of the *Cisco ONS 15454 SDH Procedure Guide*. For more general information on SW-LCAS, refer to the "Circuits and Tunnels" chapter of the *Cisco ONS 15454 Reference Manual* or the *Cisco ONS 15454 SDH Reference Manual*.

Terminal and Facility Loopback on LCAS Circuits In Split Fibre Routing

The following section lists guidelines to follow when the ML-MR-10 card includes a split fiber routing in a terminal and facility loopback on SW-LCAS circuits:

Note Make sure that you follow the guidelines and tasks listed in the following section. Not doing so will result in traffic going down on members passing through optical spans that do not have loopbacks.

- SW-LCAS circuit members must have J1 path trace set to manual.
- Transmit and receive traces must be unique.
- SW-LCAS circuits on ML-MR-10 must allow our of group (OOG) members on Trace Identifier Mismatch - Path (TIM-P).
- For members on split fiber routes, facility loopback must select the AIS option in CTC.
- Traffic hit is expected when loopback is applied. This is due to aysnchronous detection of VCAT defects and TIM-P detection on the other end of the circuit. This is acceptable since loopbacks are intrusive and affect traffic.

However, place members of an HW-LCAS circuit traversing an optical interface under maintenance in OOS,OOG (locked, outOfGroup) state before applying terminal/facility loopbacks.

Framing Mode, Encapsulation, and CRC Support

The ML-Series cards on the ONS 15454 and ONS 15454 SDH support two modes of the POS framing mechanism, GFP-F framing and HDLC framing (default). The framing mode, encapsulation, and CRC size on source and destination POS ports must match for a POS circuit to function properly. Chapter 22, "POS on ONS Ethernet Cards," explains the framing mechanisms, encapsulations, and cyclic redundancy check (CRC) bit sizes in detail.

Supported encapsulation and CRC sizes for the framing types are detailed in Table 5-3.

Table 5-3 Supported Encapsulation, Framing, and CRC Sizes for ML-Series Cards on the ONS 15454 and ONS 15454 SDH

	Encapsulations for HDLC Framing	CRC Sizes for HDLC Framing	Encapsulations for GFP-F Framing	CRC Sizes for GFP-F Framing
ML-Series	LEX (default) Cisco HDLC PPP/BCP	16-bit 32-bit (default)	LEX (default) Cisco HDLC PPP/BCP	32-bit (default)

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.

Configuring POS Interface Framing Mode

You configure framing mode on an ML-Series card only through CTC. For more information on configuring framing mode in CTC, see Chapter 2, “CTC Operations.”

Configuring POS Interface Encapsulation Type

The default Cisco EoS LEX is the primary encapsulation of ONS Ethernet cards. This encapsulation is used under HDLC framing with the protocol field set to the values specified in Internet Engineering Task Force (IETF) Request For Comments (RFC) 1841. Under GFP-F framing, the Cisco IOS CLI also uses the keyword lex. With GFP-F framing, the lex keyword is used to represent standard mapped Ethernet over GFP-F according to ITU-T G.7041.

To configure the encapsulation type for a ML-Series card, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface pos number`	Activates interface configuration mode to configure the POS interface.
Step 2	`Router(config-if)# shutdown`	Manually shuts down the interface. Encapsulation changes on POS ports are allowed only when the interface is shut down (ADMIN_DOWN).
Step 3	`Router(config-if)# encapsulation type`	Sets the encapsulation type. Valid values are: • **hdlc**—Cisco HDLC • **lex**—(default) LAN extension, special encapsulation for use with Cisco ONS Ethernet line cards. When the lex keyword is used with GFP-F framing it is standard Mapped Ethernet over GFP-F according to ITU-T G.7041. • **ppp**—Point-to-Point Protocol
Step 4	`Router(config-if)# no shutdown`	Restarts the shutdown interface.
Step 5	`Router(config)# end`	Returns to privileged EXEC mode.
Step 6	`Router# copy running-config startup-config`	(Optional) Saves configuration changes to NVRAM.

Configuring POS Interface CRC Size in HDLC Framing

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)# interface pos number`	Activates interface configuration mode to configure the POS interface.
Step 2	`Router(config-if)# crc {16 | 32}`	Sets the CRC value for HDLC framing. 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. **Note** The CRC value is fixed at a value of 32 under GFP-F framing.
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.

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`	Activates interface configuration mode to configure the POS interface.
Step 2	`Router(config-if)# mtu bytes`	Configures the MTU size up to a maximum of 9000 bytes. See Table 5-4 for default MTU sizes.
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.

Table 5-4 shows the default MTU sizes.

Table 5-4 Default MTU Size

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

Configuring Keep Alive Messages

To configure keep alive messages for the ML-Series card, perform the following steps beginning in global configuration mode:

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 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: • **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, as described in the "Monitoring and Verifying POS" section on page 5-10.

Note Cisco IOS alarm reporting commands apply only to the Cisco IOS CLI. SONET/SDH alarms reported to the TCC2/TCC2P are not affected.

	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 console 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: • **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, as described in the "Monitoring and Verifying POS" section on page 5-10.

Note Cisco IOS alarm reporting commands apply only to the Cisco IOS CLI. SONET/SDH alarms reported to the TCC2/TCC2P are not affected.

Configuring SONET/SDH Delay Triggers

You can set path alarms listed as triggers to bring down the line protocol of the POS interface. When you configure the path alarms as triggers, you can also specify a delay for the triggers using the **pos trigger delay** command. You can set the delay from 200 to 2000 ms. If you do not specify a time interval, the default delay is set to 200 ms.

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: • **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.

C2 Byte and Scrambling

One of the overhead bytes in the SONET/SDH frame is the C2 byte. The SONET/SDH 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. The C2 byte is not configurable. Table 5-5 provides C2 byte hex values.

Table 5-5 C2 Byte and Scrambling Default Values

Signal Label	SONET/SDH Payload Contents
0x01	LEX Encapsulation with 32-bit CRC with or without scrambling
0x05	LEX Encapsulation with 16-bit CRC with or without scrambling
0xCF	Cisco HDLC or PPP/BCP without scrambling

Table 5-5 C2 Byte and Scrambling Default Values (continued)

Signal Label	SONET/SDH Payload Contents
0x16	Cisco HDLC or PPP/BCP with scrambling
0x1B	GFP-F

Third-Party POS Interfaces C2 Byte and Scrambling Values

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 SPE Scrambling

SPE scrambling is on by default. To configure POS SONET/SDH Payload (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.
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.

Monitoring and Verifying POS

The **show controller pos** ***[0 | 1]*** command (Example 5-1) outputs the receive and transmit values and the C2 value. Thus, changing the value on the local end does not change the value in the **show controller** command output.

Example 5-1 show controller pos [0 | 1] Command

```
ML_Series# sh controllers pos 0
Interface POS0
Hardware is Packet/Ethernet over Sonet
Framing Mode: HDLC
```

```
Concatenation: CCAT
Alarms reportable to CLI: PAIS PLOP PUNEQ PTIM PPLM ENCAP PRDI PPDI BER_SF_B3 BER_SD_B3
VCAT_OOU_TPT LOM SQM
Link state change defects: PAIS PLOP PUNEQ PTIM PPLM ENCAP PRDI PPDI BER_SF_B3
Link state change time   : 200 (msec)
***************  Path  ***************
Circuit state: IS
    PAIS      = 0          PLOP      = 0          PRDI    = 0          PTIM = 0
    PPLM      = 0          PUNEQ     = 0          PPDI    = 0          PTIU = 0
    BER_SF_B3 = 0          BER_SD_B3 = 0          BIP(B3) = 0           REI = 0
    NEWPTR    = 0          PSE       = 0          NSE     = 0         ENCAP = 0
Active Alarms : PAIS
Demoted Alarms: None
Active Defects: PAIS
DOS FPGA channel number  : 0
Starting STS (0 based)   : 0
VT ID (if any) (0 based) : 255
Circuit size           : STS-3c
RDI Mode               : 1 bit
C2 (tx / rx)           : 0x01 / 0x01
Framing                : SONET
Path Trace
    Mode            : off
    Transmit String :
    Expected String :
    Received String :
    Buffer          : Stable
    Remote hostname :
    Remote interface:
    Remote IP addr  :
B3 BER thresholds:
SFBER = 1e-4,   SDBER = 1e-7
0 total input packets,  0 post-HDLC bytes
0 input short packets,  0 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  total output packets, 0  output pre-HDLC bytes
0 output post-HDLC bytes
Carrier delay is 200 msec
```

The **show interface pos** {**0 | 1**} command (Example 5-2) shows scrambling.

Example 5-2 show interface pos [0 | 1] Command

```
ML_Series# show interface pos 0
POS0 is administratively down, line protocol is down
    Hardware is Packet/Ethernet over Sonet, address is 0011.2130.b340 (bia 0011.2130.b340)
    MTU 1500 bytes, BW 145152 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation: Cisco-EoS-LEX, crc 32, loopback not set
    Keepalive set (10 sec)
    Scramble enabled
    ARP type: ARPA, ARP Timeout 04:00:00
    Last input 01:21:02, output never, output hang never
    Last clearing of "show interface" counters 00:12:01
    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 parity
  0 input errors, 0 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
```

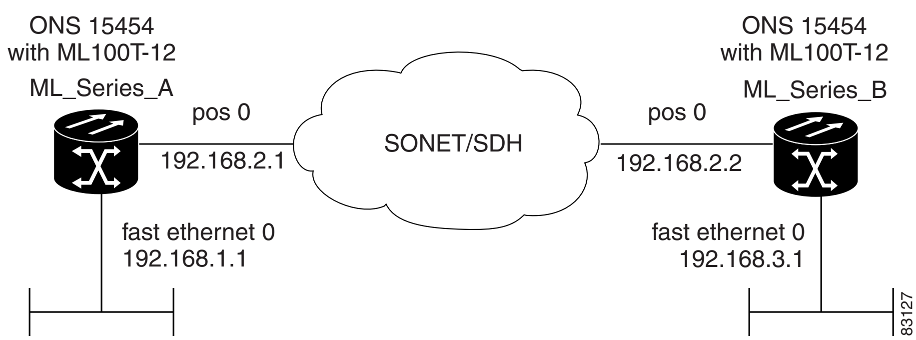
POS Configuration Examples

The following sections show ML-Series card POS configuration examples for connecting to other ONS Ethernet cards and POS-capable routers. These examples are only some of the ML-Series card configurations available to connect to other ONS Ethernet cards and POS-capable routers. For more specifics about the POS characteristics of ONS Ethernet cards, see Chapter 22, "POS on ONS Ethernet Cards."

ML-Series Card to ML-Series Card

Figure 5-1 illustrates a POS configuration between two ONS 15454 or ONS 15454 SDH ML-Series cards.

Figure 5-1 ML-Series Card to ML-Series Card POS Configuration



Example 5-3 shows the commands associated with the configuration of ML-Series card A.

Example 5-3 ML-Series Card A Configuration

```
hostname ML_Series_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 5-4 shows the commands associated with the configuration of ML Series B.

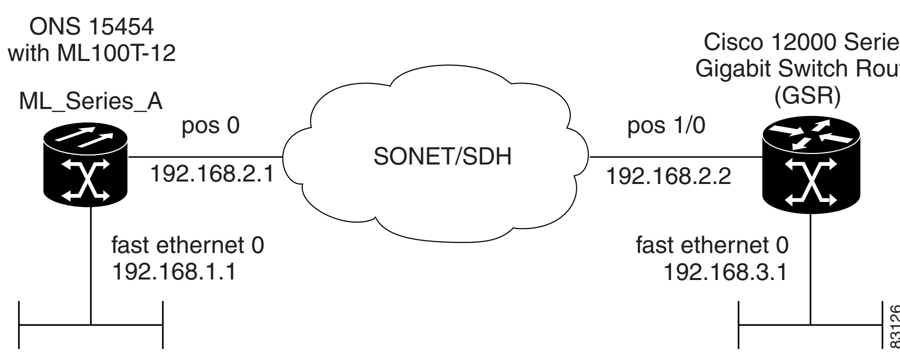
Example 5-4 ML-Series Card B Configuration

```
hostname ML_Series_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 5-2 illustrates a POS configuration between an ML-Series card and a Cisco 12000 GSR-Series router. PPP/BCP encapsulation or Cisco HDLC encapsulation may be used for interoperation.

Figure 5-2 ML-Series Card to Cisco 12000 Series Gigabit Switch Router (GSR) POS Configuration



Example 5-5 shows the commands associated with configuration of ML-Series card A.

Example 5-5 ML-Series Card A Configuration

```
hostname ML_Series_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 5-6 shows the commands associated with the configuration of the GSR-12000.

Example 5-6 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
!
```

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 5-6.

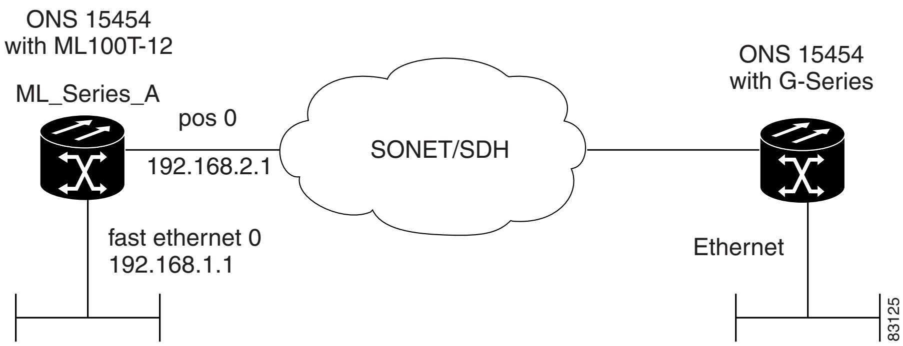
Table 5-6 ML-Series Parameter Configuration for Connection to a Cisco 12000 GSR-Series Router

Command	Parameter
`Router(config-if)#` **`encapsulation ppp`** or `Router(config-if)#` **`encapsulation hdlc`**	Encapsulation—Default encapsulation on the Cisco 12000 GSR Series is HDLC, which is supported by the ML-Series. PPP is also supported by both the ML-Series card and the Cisco 12000 GSR Series. The Cisco 12000 GSR Series does not support LEX, which is the default encapsulation on the ML-Series card.
`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 5-3 illustrates a POS configuration between an ML-Series card and a G-Series card.

Figure 5-3 ML-Series Card to G-Series Card POS Configuration



Example 5-7 shows the commands associated with the configuration of ML-Series card A.

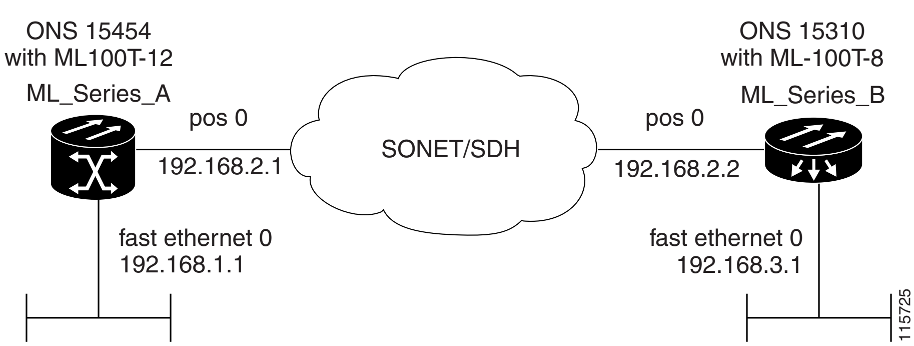
Example 5-7 ML-Series Card A Configuration

```
hostname ML_Series_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
```

ML-Series Card to ONS 15310 ML-100T-8 Card

Figure 5-4 illustrates a POS configuration between an ML-Series card and an ONS 15310 ML-100T-8 card. For step-by-step circuit configuration procedures for the connected ML-100T-8 card, refer to the *Cisco ONS 15310-CL and Cisco ONS 15310-MA Ethernet Card Software Feature and Configuration Guide*.

Figure 5-4 ML-Series Card to ONS 15310 CE-100T-8 Card Configuration



Example 5-8 shows the commands associated with the configuration of ML-Series card A.

Example 5-8 ML-Series Card A Configuration

```
hostname ML_Series_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
```

CHAPTER 6

Configuring Bridges

This chapter describes how to configure bridging for ML1000-2 Gigabit Ethernet cards, ML100T-12 Fast Ethernet cards, and ML100X-8 Fast Ethernet cards. 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:

Caution Cisco Inter-Switch Link (ISL) and Cisco Dynamic Trunking Protocol (DTP) are not supported by ML1000-2, ML100T-12, and ML100X-8 cards, but their 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 Basic Bridging

ML1000-2, ML100T-12, and ML100X-8 cards support transparent bridging for Fast Ethernet, Gigabit Ethernet and POS ports. They support a maximum of 255 active bridge groups. For information on the modes of transparent bridging, see the "Transparent Bridging Modes of Operation" section on page 6-5.

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 (STP) (optional).
- In interface configuration mode:
  - Determine which interfaces belong to the same bridge group.

    ML1000-2, ML100T-12, or ML100X-8 cards bridge all nonrouted traffic among the network interfaces comprising the bridge group. If spanning tree is enabled, the interfaces became part of the same spanning tree. 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.

Spanning tree is not mandatory for an ML1000-2, ML100T-12, or ML100X-8 bridge group. But if it is configured, a separate spanning-tree process runs for each configured bridge group. A bridge group establishes a spanning tree based on the bridge protocol data units (BPDUs) it receives on only its member interfaces.

Configuring Basic Bridging

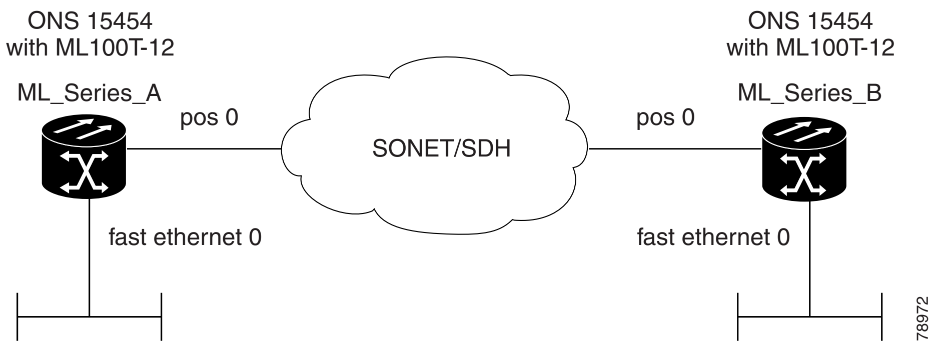
Use the following steps to configure bridging:

	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`** *`bridge-group-number`* `[`**`protocol {drpi-rstp | rstp | ieee}`**`]`	Assigns a bridge group number and defines the appropriate spanning-tree type: bridge-group-number can range from 1 to 4096. • **drpri-rstp** is the protocol used to interconnect dual RPR interconnect to protect from node failure • **rstp** is the IEEE 802.1W Rapid Spanning Tree. • **ieee** is the IEEE 802.1D Spanning Tree Protocol. **Note** Spanning tree is not mandatory for an ML1000-2, ML100T-12, or ML100X-8 bridge group. But configuring spanning tree blocks network loops.
Step 3	`Router(config)#` **`bridge`** *`bridge-group-number`* **`priority`** *`number`*	(Optional) Assigns a specific priority to the bridge, to assist in the spanning-tree root definition. Lowering the priority of a bridge makes it more likely the bridge is selected as the root.
Step 4	`Router(config)#` **`interface`** *`type number`*	Enters interface configuration mode to configure the interface of the ML1000-2, ML100T-12, or ML100X-8 card.
Step 5	`Router(config-if)#` **`bridge-group`** *`bridge-group-number`*	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.

Bridging Examples

The ML1000-2, ML100T-12, and ML100X-8 cards all have bridging capability. In the following figures, an ML100T-12 configuration is shown as a representative model for all three cards. Figure 6-1 shows a basic bridging example. Example 6-1 shows the configuration of the east M100T-12 card. Example 6-2 shows the configuration of the west ML100T-12.

Figure 6-1 Bridging Example



Example 6-1 East Router 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 6-2 West Router 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 Basic Bridging

After you have set up an ML1000-2, ML100T-12, or ML100X-8 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`** *`bridge-group-number`*	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`** `{`*`bridge-group-number`* `|` *`interface-address`*`}`	Displays classes of entries in the bridge forwarding database.
Step 3	`Router#` **`show bridge verbose`**	Displays detailed information about configured bridge groups.
Step 4	`ML_Series#` **`show spanning-tree`** `[`*`bridge-group-number`*`] [`**`brief`**`]`	Displays detailed information about spanning tree. **bridge-group-number** restricts the spanning tree information to specific bridge groups. **brief** displays summary information about spanning tree.

Example 6-3 shows an example of monitoring and verifying bridging.

Example 6-3 Monitoring and Verifying Bridging

```
ML-Series# 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


ML-Series# 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
unt
 1 60/0   0000.0001.6000 forward  FastEthernet0        -
 1 61/0   0000.0001.6100 forward  POS0                 -

Flood ports
FastEthernet0
POS0

ML-Series# show spanning-tree brief
```

```
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
```

Transparent Bridging Modes of Operation

The transparent bridging feature in the Cisco IOS software combines bridge-groups and IP routing. This combination provides the speed of an adaptive spanning-tree bridge, along with the functionality, reliability, and security of a router. ML1000-2, ML100T-12, and ML100X-8 cards support transparent bridging in the same general manner as other Cisco IOS platforms.

Transparent bridging processes IP frames in four distinct modes, each with different rules and configuration options. The modes are IP routing, no IP routing, bridge crb, and bridge irb. This section covers the configuration and operation of these four modes on ML1000-2, ML100T-12, and ML100X-8 cards.

For additional general Cisco IOS user documentation on configuring transparent bridging, see the "Configuring Transparent Bridging" chapter of the *Cisco IOS Bridging and IBM Networking Configuration Guide, Release 12.2* at:
http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca767.html

IP Routing Mode

IP routing mode is the default mode. It disables the other modes (no IP routing, bridge crb, and bridge irb). The global command **ip routing** enables IP routing mode.

In IP routing mode, the bridge-groups do not process IP packets. The IP packets are either routed or discarded.

The following rules help describe packet handling in this mode:

- An input interface or subinterface configured with only a bridge-group will bridge non-IP packets and discard IP packets (Example 6-4).
- An input interface or subinterface configured with only an IP address will route IP packets and discard non-IP packets (Example 6-5).
- An input interface or subinterface configured with both an IP address and a bridge-group routes IP packets and bridges non-IP packets (Example 6-6). This configuration is sometimes referred to as fallback bridging. If a protocol cannot be routed, then the interface falls back to bridging.

- All of the interfaces or subinterfaces belonging to a specific bridge-group need consistent configuration with regards to configuring or not configuring IP addresses. Mixing interfaces configured with IP addresses and interfaces not configured with IP addresses in the same bridge group can cause inconsistent or unpredictable routing at the network level.
- All the interfaces and subinterface belonging to the same bridge-group need consistent configuration with regard to IP addresses. Either all of the bridge group's interfaces should be configured with IP addresses or none of the bridge group's interfaces should be configured with IP addresses.

Example 6-4 shows card interfaces configured in a bridge group with no IP addresses.

Example 6-4 Bridge Group with No IP Address

```
ip routing
bridge 1 proto rstp

int f0
bridge-group 1

int pos 0
bridge-group 1
```

Example 6-5 shows card interfaces configured with IP addresses but not in a bridge group.

Example 6-5 IP Addresses with No Bridge Group

```
ip routing

int f0
ip address 10.10.10.2 255.255.255.0

int pos 0
ip address 20.20.20.2 255.255.255.0
```

Example 6-6 shows card interfaces configured with IP addresses and in a bridge group.

Example 6-6 IP Addresses with Bridge Group

```
ip routing
bridge 1 proto rstp

int f0
ip address 10.10.10.2 255.255.255.0
bridge-group 1

int pos 0
ip address 20.20.20.2 255.255.255.0
bridge-group 1
```

No IP Routing Mode

The no IP routing mode bridges all packets, both IP and non-IP, and prevents routing. Although Cisco IOS can use the IP addresses for interfaces configured as management ports, it will not route between these IP addresses.

The global command **no ip routing** enables this feature, and enabling no ip routing disables the other modes.

The following rules help describe packet handling in this mode:

- An input interface or subinterface configured with only a bridge-group and no ip addresses bridges all packets (Example 6-7).
- An input interface or subinterface configured with only an IP address discards all packets, except packets with the destination MAC and IP address of the input interface, which are processed by Cisco IOS. This is not a valid configuration.
- An input interface or subinterface configured with both an IP address and a bridge group bridges all packets, except packets sent to the input interface MAC address. Packets sent to the input interface MAC address and the interface IP address are processed by Cisco IOS. Other packets sent to the input interface MAC address are discarded. This is not a valid configuration for the IP addresses.
- All of the interfaces or subinterfaces belonging to a specific bridge-group need consistent configuration in regards to configuring or not configuring IP addresses. Mixing interfaces configured with IP addresses and interfaces not configured with IP addresses in the same bridge group can cause inconsistent or unpredictable routing at the network level.

Example 6-7 shows card interfaces configured in a bridge group with no IP addresses.

Example 6-7 Bridge Group with No IP Address

```
no ip routing
bridge 1 proto rstp

int f0
bridge-group 1

int pos 0
bridge-group 1
```

Bridge CRB Mode

In bridge crb mode, the default sub-mode for every bridge group is to bridge but not route the IP packets. This is similar to the no ip routing mode behavior. But with bridge crb, packet handling is configured not globally but for the specific bridge group. You can selectively disable bridge groups to block IP packets or configure fallback bridging for a group of routed interfaces.

Concurrent routing and bridging is enabled with the global command **bridge crb.** Enabling bridge crb disables the other modes.

The following rules help describe packet handling in this mode:

- The command **bridge *x* bridge ip** (where *x* is a bridge-group number) configures a bridge-group to bridge IP packets. Input interfaces and sub-interfaces belonging to the bridge-group will follow the rules for no IP routing mode.
- The command **bridge *x* route IP** (where *x* is a bridge-group number) configures a bridge-group to ignore IP packets. Input interfaces and sub-interfaces belonging to this bridge-group will follow the rules for IP routing mode (Example 6-8).
- When you enable bridge crb with pre-existing bridge groups, it will generate a **bridge x route IP** configuration command for any pre-existing bridge groups with an interface configured for routing (configured with an IP address). This is a precaution when crb is first enabled.

- All of the interfaces or subinterfaces belonging to a specific bridge-group need consistent configuration in regards to configuring or not configuring IP addresses. Mixing interfaces configured with IP addresses and interfaces not configured with IP addresses in the same bridge group can cause inconsistent or unpredictable routing at the network level.
- Routing between interfaces or subinterfaces that do not belong to the same bridge group could result in inconsistent network behavior. This mode is for routing between members of a bridge-group, but never for routing into or out of a bridge group.

Example 6-8 shows card interfaces configured with IP addresses and multiple bridge groups.

Example 6-8 IP Addresses and Multiple Bridge Group

```
bridge crb
bridge 1 proto rstp
bridge 1 route ip
bridge 2 proto rstp

int f0
ip address 10.10.10.2 255.255.255.0
bridge-group 1

int pos 0
ip address 20.20.20.2 255.255.255.0
bridge-group 1

int f1
bridge-group 2

int pos 1
bridge-group 2
```

Tip When troubleshooting a bridge crb configuration, make sure the interfaces are not assigned IP addresses belonging to the same subnet. Routing requires IP addresses to be in different subnets.

Bridge IRB Mode

The integrated routing and bridging mode is enabled with the global command **bridge irb**. Enabling bridge irb disables the other modes.

Bridge irb mode is a super-set of the bridge crb mode. Only IRB mode supports a bridged virtual interface (BVI), which is a virtual Layer 3 interface belonging to a specific bridge-group. A BVI requires an IP address to function and is visible to all member interfaces of that bridge-group. The only proper way to route into and out of a bridge-group is with a BVI.

Bridge irb behaves like bridge crb with the following additions:

- If a BVI interface is configured for a bridge-group, the BVI IP address should be the only one configured on any member of that bridge-group (Example 6-9).
- If both an IP address and a bridge-group are configured on a single interface, enable either IP bridging or IP routing, but not both (Example 6-10).
- If IP routing is disabled in a bridge-group, all packets will be bridged, and BVI interfaces will not route IP. This is the default for each bridge-group.

- If IP bridging and IP routing are both enabled in a bridge-group with a BVI, then IP packets can be bridged between bridge-group members (bridging within the same subnet), and they can be routed in and out of the bridge-group via the BVI.
- If IP bridging is disabled, but IP routing is enabled in a bridge-group, IP packets can be routed in and out of the bridge-group through the BVI but cannot be bridged between the Layer 2 interfaces. The global command **bridge x route ip** in combination with the global command **no bridge *x* bridge ip** disables IP bridging while enabling IP routing.

Example 6-9 shows card interfaces configured in a bridge group and the BVI configured with an IP address. Both bridging and routing are enabled.

Example 6-9 Bridge irb with Routing and Bridging Enabled

```
bridge irb
bridge 1 proto rstp
bridge 1 route ip

int f0
bridge-group 1

int pos 0
bridge-group 1

int bvi 1
ip address 10.10.10.1 255.255.255.0
```

Example 6-10 shows card interfaces configured with both an IP address and a bridge-group. IP routing is enabled and IP bridging is disabled.

Example 6-10 IP Addresses and Multiple Bridge Group

```
bridge irb
bridge 1 proto rstp
bridge 1 route ip
no bridge 1 bridge ip

int f0
ip address 10.10.10.1 255.255.255.0
bridge-group 1

int pos 0
ip address 20.20.20.2 255.255.255.0
bridge-group 2
```

Tip When troubleshooting bridge irb, make sure the BVI is configured with an IP address and the BVI bridge members are not configured with IP addresses.

CHAPTER 7

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

These sections describe how the spanning-tree features work:

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.

BPDUs contain 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 an 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 IEEE 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 7-1, 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 7-1 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 7-2 describes the timers that affect the entire spanning-tree performance.

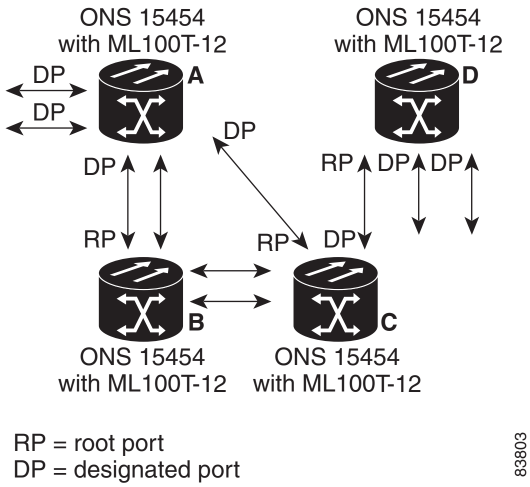
Table 7-2 Spanning-Tree Timers

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 7-1, Switch A is elected as the root switch because the switch priority of all the switches is set to the default (32768) and Switch A has the lowest MAC address. However, because of traffic patterns, number of forwarding interfaces, or link types, Switch 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 7-1 Spanning-Tree Topology



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:

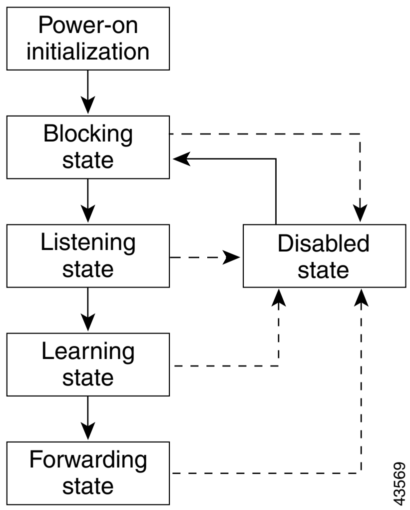
- Blocking—The interface does not participate in frame forwarding.
- Listening—The first transitional state after the blocking state when the spanning tree determines that the interface should participate in frame forwarding.
- Learning—The interface prepares to participate in frame forwarding.
- Forwarding—The 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 7-2 illustrates how an interface moves through the states.

Figure 7-2 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 01000CCCCCCD) when they are being tunneled via the protocol tunneling feature.

STP and IEEE 802.1Q Trunks

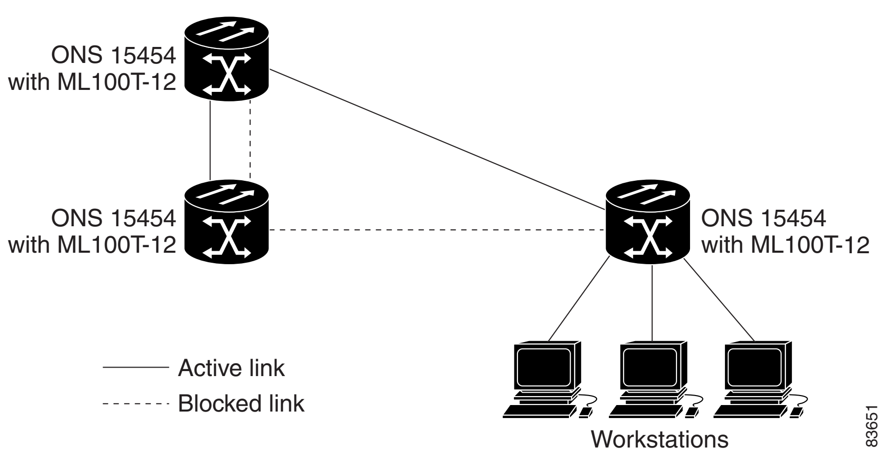
When you connect a Cisco switch to a non-Cisco device through an IEEE 802.1Q trunk, the Cisco switch uses PVST+ to provide spanning-tree interoperability. PVST+ is automatically enabled on IEEE 802.1Q trunks after users assign a protocol to a bridge group. The external spanning-tree behavior on access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

For more information on IEEE 802.1Q trunks, see Chapter 8, "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 7-3. 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 7-3 Spanning Tree and Redundant Connectivity



You can also create redundant links between switches by using EtherChannel groups. For more information, see Chapter 10, "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) IEEE 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 IEEE 802.1D spanning tree), which is critical for networks carrying delay-sensitive traffic such as voice and video.

These sections describe how RSTP works:

- Supported RSTP Instances, page 7-9
- Port Roles and the Active Topology, page 7-9
- Rapid Convergence, page 7-10
- Synchronization of Port Roles, page 7-12
- Bridge Protocol Data Unit Format and Processing, page 7-13
- Topology Changes, page 7-14

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 the "Election of the Root Switch" section on page 7-3. 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 IEEE 802.1D). The port state controls the operation of the forwarding and learning processes. Table 7-3 provides a comparison of IEEE 802.1D and RSTP port states.

Table 7-3 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 7-3 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 7-4, 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.

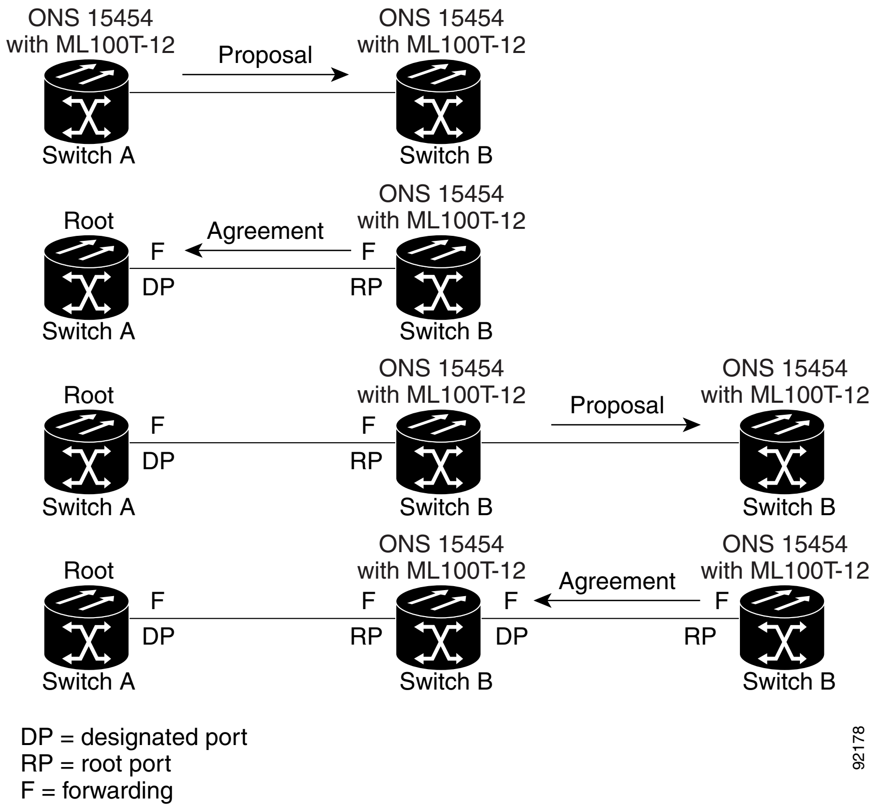
  After receiving the proposal message, Switch B selects as its new root port the port from which the proposal message was received, forces all non edge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

  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 7-4 Proposal and Agreement Handshaking for Rapid Convergence



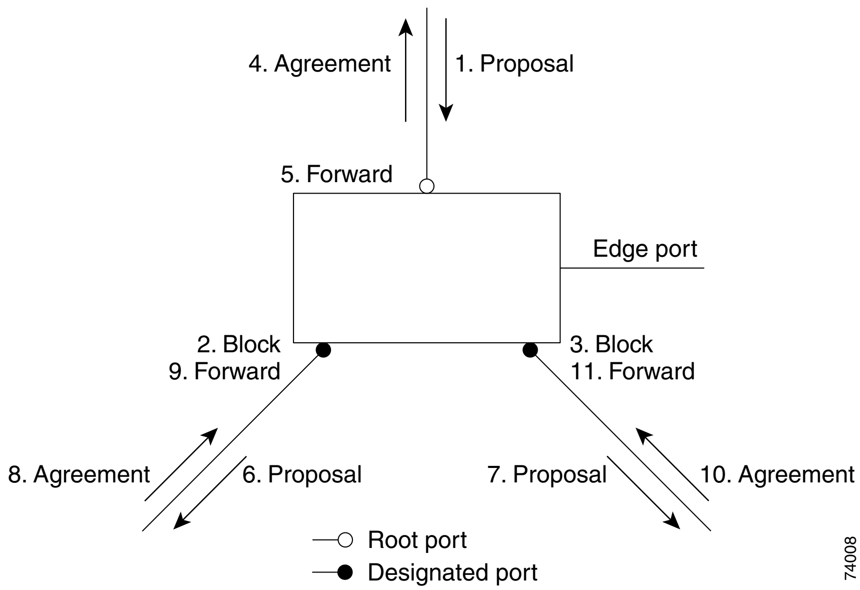
Synchronization of Port Roles

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 7-5.

Figure 7-5 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 7-4 shows the RSTP flag fields.

Table 7-4 RSTP BPDU Flags

Bit	Function
0	Topology change (TC)
1	Proposal
2–3: 00 01 10 11	Port role: Unknown Alternate port Root port 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 IEEE 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 IEEE 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 IEEE 802.1D in handling spanning-tree topology changes.

- Detection—Unlike IEEE 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 IEEE 802.1D, which uses TCN BPDUs, the RSTP does not use them. However, for IEEE 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 IEEE 802.1D switch, it replies with an IEEE 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 IEEE 802.1D) is active on a root port connected to an IEEE 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 IEEE 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 IEEE 802.1D switches, RSTP selectively sends IEEE 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 IEEE 802.1D BPDU after the port's migration-delay timer has expired, it assumes that it is connected to an IEEE 802.1D switch and starts using only IEEE 802.1D BPDUs. However, if the RSTP switch is using IEEE 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 IEEE 802.1D STP

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

However, the switch does not automatically revert to the RSTP mode if it no longer receives IEEE 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 7-16
- Disabling STP and RSTP, page 7-16
- Configuring the Root Switch, page 7-17
- Configuring the Port Priority, page 7-17
- Configuring the Path Cost, page 7-18
- Configuring the Switch Priority of a Bridge Group, page 7-19
- Configuring the Hello Time, page 7-19
- Configuring the Forwarding-Delay Time for a Bridge Group, page 7-20
- Configuring the Maximum-Aging Time for a Bridge Group, page 7-20

Default STP and RSTP Configuration

Table 7-5 shows the default STP and RSTP configuration.

Table 7-5 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 native 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	`Router#` **`configure terminal`**	Enters the global configuration mode.
Step 2	`Router(config)#` **`interface`** *`interface-id`*	Enters the interface configuration mode.
Step 3	`Router(config-if)#` **`bridge-group`** *`bridge-group-number`* **`spanning disabled`**	Disables STP or RSTP on a per-interface basis.
Step 4	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.

To reenable 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	`Router#` **`configure terminal`**	Enters the global configuration mode.
Step 2	`Router(config)#` **`interface`** *`interface-id`*	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	`Router(config-if)#` **`bridge-group`** *`bridge-group-number priority-value`*	Configures the port priority for an interface that is an access port. For the *priority-value*, the range is 0 to 255; the default is 128 in increments of 16. The lower the number, the higher the priority.
Step 4	`Router(config-if)#` **`end`**	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	`Router#` **`configure terminal`**	Enters the global configuration mode.
Step 2	`Router(config)#` **`interface`** *`interface-id`*	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*).
Step 3	`Router(config-if)#` **`bridge-group`** *`bridge-group-number`* **`path-cost`** *`cost`*	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 *cost*, the range is 0 to 65535; the default value is derived from the media speed of the interface.
Step 4	`Router(config-if)#` **`end`**	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 *priority*, 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 *seconds*, 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 *seconds*, 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 *seconds*, 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 7-6:

Table 7-6 Commands for Displaying Spanning-Tree Status

Command	Purpose
`ML_Series# show spanning-tree`	Displays detailed STP or RSTP information.
`ML_Series# show spanning-tree brief`	Displays summary of STP or RSTP information.

Table 7-6 Commands for Displaying Spanning-Tree Status (continued)

Command	Purpose
ML_Series# **show spanning-tree interface** *interface-id*	Displays STP or RSTP information for the specified interface.
ML_Series# **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 7-1 show spanning-tree Commands

```
Router# show spanning-tree brief

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
```

CHAPTER 8

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 IOS commands used in this chapter, refer to the *Cisco IOS Command Reference* publication.

This chapter contains the following major sections:

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

Understanding VLANs

VLANs 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.

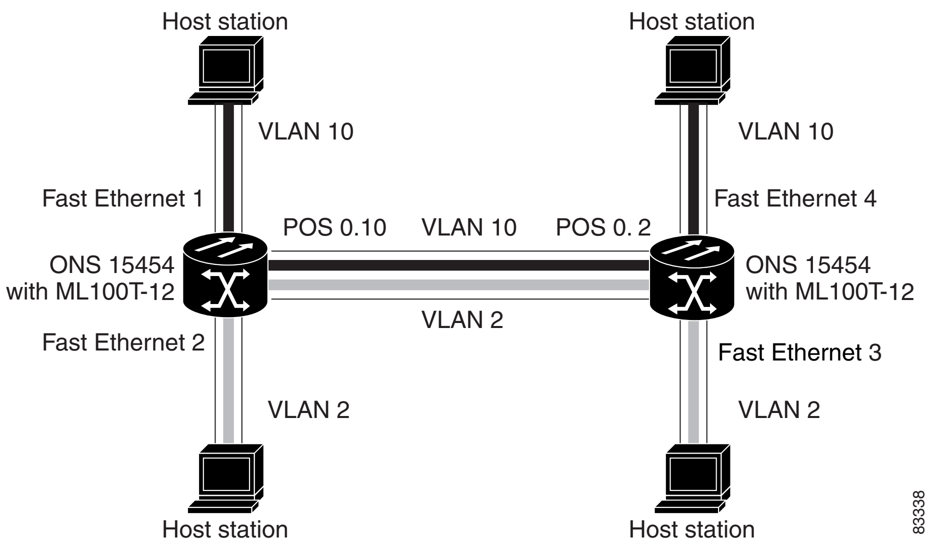
VLANs enable efficient traffic separation and provide excellent bandwidth utilization. VLANs also alleviate scaling issues by logically segmenting the physical LAN structure into different subnetworks so that packets are switched only between ports within the same VLAN. This can be very useful for security, 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 card software supports VLAN frame encapsulation through the IEEE 802.1Q standard. The Cisco Inter-Switch Link (ISL) VLAN frame encapsulation is not supported. ISL frames are 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 four 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 to 4095 range. Figure 8-1 shows a network topology in which two VLANs span two ONS 15454s with ML-Series cards.

Figure 8-1 VLANs Spanning Devices in a Network



Configuring IEEE 802.1Q VLAN Encapsulation

You can configure IEEE 802.1Q VLAN encapsulation on either type of ML-Series card interfaces, Ethernet or Packet over SONET/SDH (POS). VLAN encapsulation is not supported on POS interfaces configured with HDLC encapsulation.

The native VLAN is always VLAN ID 1 on ML-Series cards. Frames on the native VLAN are normally transmitted and received untagged. On an trunk port, all frames from VLANs other than the native VLAN are transmitted and received tagged.

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`** *`bridge-group-number`* **`protocol`** *`type`*	Assigns a bridge group (VLAN) number and define the appropriate spanning tree type.
Step 2	`Router(config)#` **`interface`** *`type number`*	Enters interface configuration mode to configure the interface.
Step 3	`Router(config-if)#` **`no ip address`**	Disables IP processing.
Step 4	`Router(config)#` **`interface`** *`type number.subinterface-number`*	Enters subinterface configuration mode to configure the subinterface.

	Command	Purpose
Step 5	`Router(config-subif)# encap dot1q vlan-number`	Sets the encapsulation on the VLAN to IEEE 802.1Q.
Step 6	`Router(config-subif)# bridge-group bridge-group-number`	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 know as VLAN translation.

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

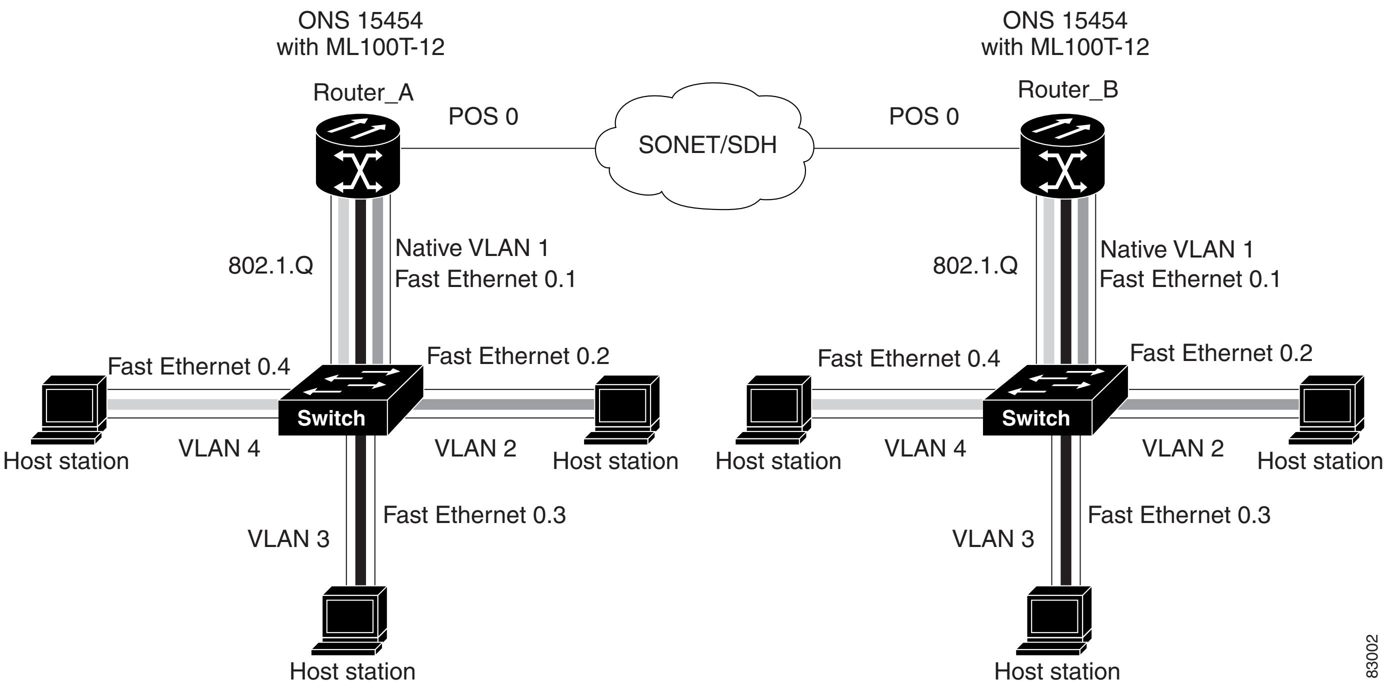
Note Native VLAN frames transmitted on the interface are normally untagged. All untagged frames received on the interface are associated with the native VLAN, which is always VLAN 1. Use the command **encapsulation dot1q 1 native**.

IEEE 802.1Q VLAN Configuration

The VLAN configuration example for the ML100T-12 shown in Figure 8-2 depicts the following VLANs:

- 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 8-2 Bridging IEEE 802.1Q VLANs



Example 8-1 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 8-2:

Example 8-1 Configure VLANs for IEEE 802.1Q 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).

An example of the **show vlans** privileged EXEC command commands are shown here:

Example 8-2 show vlans Commands

```
ML1000-121#show vlans
Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)
    vLAN Trunk Interfaces:  POS1
GigabitEthernet0
    This is configured as native Vlan for the following interface(s) :
POS1
GigabitEthernet0
    Protocols Configured:   Address:              Received:        Transmitted:
Virtual LAN ID:  5 (IEEE 802.1Q Encapsulation)
    vLAN Trunk Interfaces:  POS1.1
GigabitEthernet0.1
    Protocols Configured:   Address:              Received:        Transmitted:
    Bridging        Bridge Group 2                157                   0
    Bridging        Bridge Group 2                157                   0
```

CHAPTER 9

Configuring IEEE 802.1Q Tunneling 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 impairing the traffic of other customers. The ML-Series cards support IEEE 802.1Q tunneling and Layer 2 protocol tunneling.

This chapter contains the following sections:

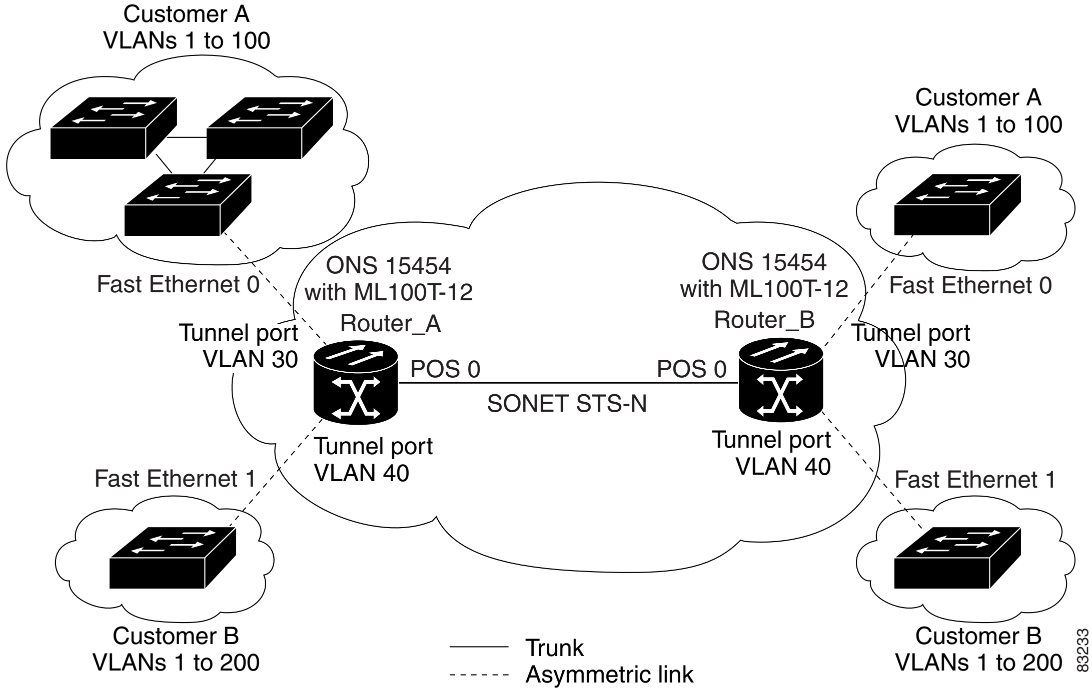
Understanding IEEE 802.1Q Tunneling

Business customers of service providers often have specific requirements for VLAN IDs and the number of supported VLANs. 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 specification VLAN limit of 4096.

Using the IEEE 802.1Q tunneling (QinQ) 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 (Figure 9-1).

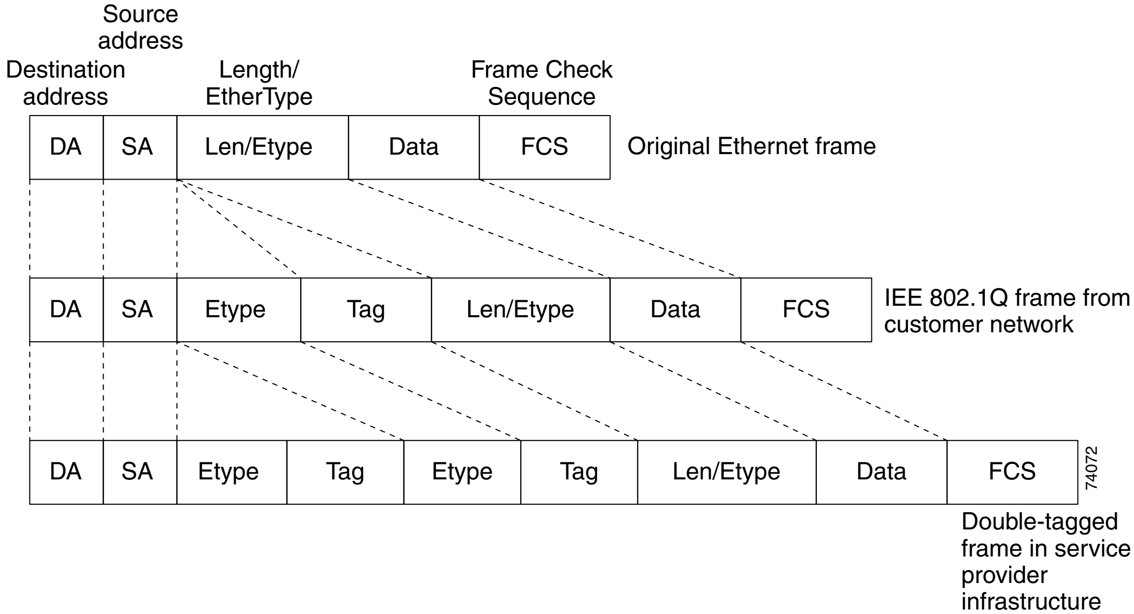
Figure 9-1 IEEE 802.1Q Tunnel Ports in a Service-Provider Network



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, they 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 9-2 shows the structure of the double-tagged packet.

Figure 9-2 Normal, IEEE 802.1Q, and IEEE 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 9-1 on page 9-2, 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 the native VLAN (VLAN 1) is used in the service provider network as a metro tag, this tag must 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 **vlan dot1q tag native** command 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 the following information about configuring IEEE 802.1Q tunneling:

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

IEEE 802.1Q Tunneling and Compatibility with 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	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`bridge`** *`bridge-number`* **`protocol`** *`bridge-protocol`*	Creates a bridge number and specifies a protocol.
Step 3	`Router(config)#` **`interface fastethernet`** *`number`*	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	Router(config-if)# **bridge-group** *number*	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 subinterfaces on a provider trunk interface.
Step 5	Router(config-if)# **mode dot1q-tunnel**	Sets the interface as an IEEE 802.1Q tunnel port.
Step 6	Router(config)# **end**	Returns to privileged EXEC mode.
Step 7	Router# **show dot1q-tunnel**	Displays the tunnel ports on the switch.
Step 8	Router# **copy running-config startup-config**	(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 to be used as a metro tag, 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.

IEEE 802.1Q Example

The following examples show how to configure the example in Figure 9-1 on page 9-2. Example 9-1 applies to Router A, and Example 9-2 applies to Router B.

Example 9-1 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 9-2 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 VLAN-Transparent and VLAN-Specific Services

The ML-Series card supports combining VLAN-transparent services and one or more VLAN-specific services on the same port. All of these VLAN-transparent and VLAN-specific services can be point-to-point or multipoint-to-multipoint.

This allows a service provider to combine a VLAN-transparent service, such as IEEE 802.1Q tunneling (QinQ), with VLAN-specific services, such as bridging specific VLANs, on the same customer port. For example, one customer VLAN can connect to Internet access and the other customer VLANs can be tunneled over a single provider VLAN to another customer site, all over a single port at each site. Table 9-1 outlines the differences between VLAN-transparent and VLAN-specific services.

Table 9-1 VLAN-Transparent Service Versus VLAN-Specific Services

VLAN-Transparent Services	VLAN-Specific Services
Bridging only	Bridging or routing
One service per port	Up to 254 VLAN-specific services per port
Applies indiscriminately to all VLANs on the physical interface	Applies only to specified VLANs

Note VLAN-transparent service is also referred to as Ethernet Wire Service (EWS). VLAN-specific service is also referred to as QinQ tunneling trunk UNI in Metro Ethernet terminology.

A VLAN-specific service on a subinterface coexists with the VLAN-transparent service, often IEEE 802.1Q tunneling, on a physical interface. VLANs configured for a VLAN-transparent service and a VLAN-specific service follow the VLAN-specific service configuration. If you need to configure 802.1Q tunneling, configure this VLAN-transparent service in the normal manner, see the "Configuring IEEE 802.1Q Tunneling" section on page 9-4.

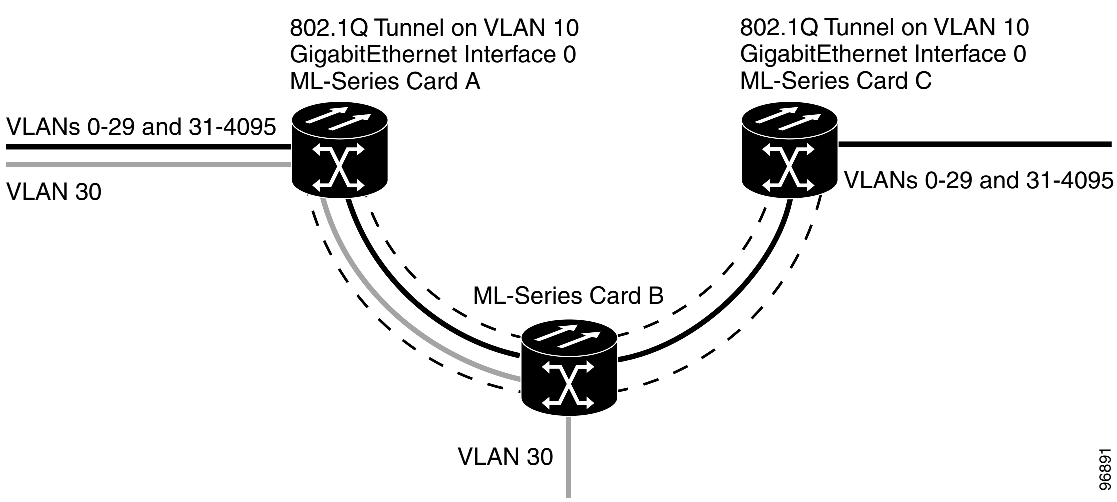
A VLAN-specific service can be any service normally applicable to a VLAN. To configure an ERMS VLAN-specific service, configure the service in the normal manner.

VLAN-Transparent and VLAN-Specific Services Configuration Example

In this example, the Gigabit Ethernet interfaces 0 on both the ML-Series card A and ML-Series card C are the trunk ports in an IEEE 802.1Q tunnel, a VLAN-transparent service. VLAN 10 is used for the VLAN-transparent service, which would normally transport all customer VLANs on the ML-Series card A's Gigabit Ethernet interface 0. All unspecified VLANs and VLAN 1 would also be tunneled across VLAN 10.

VLAN 30 is prevented from entering the VLAN-transparent service and is instead forwarded on a specific-VLAN service, bridging Gigabit Ethernet interface 0 on ML-Series card A and Gigabit Ethernet interface 0 on ML-Series card B. Figure 9-3 is used as an example to performing configuration examples 9-3, 9-4, and 9-5.

Figure 9-3 ERMS Example



Example 9-3 applies to ML-Series card A.

Example 9-3 ML-Series Card A Configuration

```
hostname ML-A
bridge 10 protocol rstp
```

```
bridge 30 protocol ieee
!
!
interface GigabitEthernet0
    no ip address
    no ip route-cache
    mode dot1q-tunnel
    bridge-group 10
     bridge-group 10 spanning-disabled
!
interface GigabitEthernet0.3
 encapsulation dot1Q 30
 no ip route-cache
bridge-group 30
!
interface POS0
 no ip address
 no ip route-cache
 crc 32
!
interface POS0.1
 encapsulation dot1Q 10
 no ip route-cache
 bridge-group 10
!
interface POS0.3
 encapsulation dot1Q 30
 no ip route-cache
 bridge-group 30
```

Example 9-4 applies to ML-Series card B.

Example 9-4 ML-Series Card B Configuration

```
hostname ML-B
!
bridge 10 protocol rstp
bridge 30 protocol ieee
!
!
interface GigabitEthernet0
 no ip address
!
interface GigabitEthernet0.3
 encapsulation dot1Q 30
 bridge-group 30
!
interface GigabitEthernet1
 no ip address
 shutdown
!
interface POS0
 no ip address
 crc 32
!
interface POS0.1
 encapsulation dot1Q 10
 bridge-group 10
!
interface POS0.3
 encapsulation dot1Q 30
 bridge-group 30
!
```

```
interface POS1
 no ip address
 crc 32
!
interface POS1.1
 encapsulation dot1Q 10
 bridge-group 10
!
interface POS1.3
 encapsulation dot1Q 30
 bridge-group 30
```

Example 9-5 applies to ML-Series card C.

Example 9-5 ML-Series Card C Configuration

```
hostname ML-C
bridge 10 protocol rstp
!
!
interface GigabitEthernet0
    no ip address
    no ip route-cache
    mode dot1q-tunnel
    bridge-group 10
     bridge-group 10 spanning-disabled
!
interface POS0
 no ip address
 no ip route-cache
 crc 32
!
interface POS0.1
 encapsulation dot1Q 10
 no ip route-cache
 bridge-group 10
```

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. Spanning Tree Protocol (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. CDP, STP, or VTP Layer 2 protocol data units (PDUs) 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 or on specific VLANs, 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 or on specific tunnel VLANs that are connected to the customer by 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 at the interface and subinterface level. Multiple STP (MSTP) Tunneling support is achieved through subinterface protocol tunneling. 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 the following information about configuring Layer 2 protocol tunneling:

Default Layer 2 Protocol Tunneling Configuration

Table 9-2 shows the default Layer 2 protocol tunneling configuration.

Table 9-2 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 ML-Series card supports Per-VLAN Protocol Tunneling (PVPT), which allows protocol tunneling to be configured and run on a specific subinterface (VLAN). PVPT configuration is done at the subinterface level.
- PVPT should be configured on VLANs that carry multi-session transport (MST) BPDUs on the connected devices.
- The ML-Series card supports tunneling of CDP, STP (including MSTP and VTP protocols). Protocol tunneling is disabled by default but can be enabled for the individual protocols on IEEE 802.1Q tunnel ports or on specific VLANs.
- 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 affect 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 configured 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 Layer 2 Tunneling on a Port

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

	Command	Purpose
Step 1	Router# **configuration terminal**	Enters global configuration mode.
Step 2	Router(config)# **bridge** *bridge-group-number* **protocol** *type*	Creates a bridge group number and specifies a protocol.
Step 3	Router(config)# **l2protocol-tunnel cos** ***cos-value***	Associates a CoS value with the Layer 2 tunneling port. Valid numbers for a *cos-value* range from 0 to 7.
Step 4	Router(config)# **interface** ***type*** *number*	Enters interface configuration mode for the interface to be configured as a tunnel port.
Step 5	Router(config-if)# **bridge-group** *bridge-group-number*	Assigns a bridge group to the interface.
Step 6	Router(config-if)# **mode dot1q tunnel**	Sets the interface as an IEEE 802.1Q tunnel VLAN.
Step 7	Router(config-if)# **l2protocol-tunnel** {**all** | **cdp** | **stp** | **vtp]**}	Sets the interface as a Layer 2 protocol tunnel port and enables all three protocols or specifically enables CDP, STP, or VTP. These protocols are off by default.
Step 8	Router(config-if)# **end**	Returns to privileged EXEC mode.
Step 9	Router# **show dot1q-tunnel**	Displays the tunnel ports on the switch.
Step 10	Router# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring Layer 2 Tunneling Per-VLAN

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

	Command	Purpose
Step 1	Router# **configuration terminal**	Enters global configuration mode.
Step 2	Router(config)# **bridge** *bridge-group-number* **protocol** *type*	Creates a bridge group number and specifies a protocol.
Step 3	Router(config)# **l2protocol-tunnel cos** ***cos-value***	Associates a CoS value with the Layer 2 tunneling VLAN. Valid numbers for a *cos-value* range from 0 to 7.
Step 4	Router(config)# **interface** ***type*** *number.subinterface-number*	Enters subinterface configuration mode and the subinterface to be configured as a tunnel VLAN.
Step 5	Router(config-subif)# **encapsulation dot1q** ***bridge-group-number***	Sets the subinterface as an IEEE 802.1Q tunnel VLAN.
Step 6	Router(config-subif)# **bridge-group** ***bridge-group-****number*	Assigns a bridge group to the interface.
Step 7	Router(config-subif)# **end**	Returns to privileged EXEC mode.
Step 8	Router# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Monitoring and Verifying Tunneling Status

Table 9-3 shows the privileged EXEC commands for monitoring and maintaining IEEE 802.1Q and Layer 2 protocol tunneling.

Table 9-3 Commands for Monitoring and Maintaining Tunneling

Command	Purpose
show dot1q-tunnel	Displays IEEE 802.1Q tunnel ports on the switch.
show dot1q-tunnel interface *interface-id*	Verifies if a specific interface is a tunnel port.
show l2protocol-tunnel	Displays information about Layer 2 protocol tunneling ports.
show vlan dot1q tag native	Displays IEEE 802.1Q tunnel information.

CHAPTER 10

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

The ML-Series card offers both EtherChannel and POS channel. Traditionally EtherChannel is a trunking technology that groups together multiple full-duplex IEEE 802.3 Ethernet interfaces to provide fault-tolerant high-speed links between switches, routers, and servers. 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. POS channel is only supported with LEX encapsulation.

Link aggregation provides the following benefits:

- Logical aggregation of bandwidth
- Load balancing
- Fault tolerance

Port channel is a term for both POS channel and EtherChannel. The port channel interface is treated as a single logical interface although it consists of multiple interfaces. Each port channel interfaces consists of one type of interface, either Fast Ethernet, Gigabit Ethernet, or POS. You must perform all port channel configurations on the port channel (EtherChannel or POS channel) interface rather than on the individual member Ethernet or POS interfaces. You can create the port channel interface by entering the **interface port-channel** interface configuration command.

Note You must perform all IOS configurations—such as bridging, routing, or parameter changes such as an MTU change—on the port channel (EtherChannel or POS channel) interface rather than on individual member Ethernet or POS interfaces.

Port channel connections are fully compatible with IEEE 802.1Q trunking and routing technologies. IEEE 802.1Q trunking can carry multiple VLANs across a port channel.

Each ML100T-12, ML100X-8, or ML1000-2 card supports one POS channel, a port channel made up of the two POS ports. A POS channel combines the two POS port capacities into a maximum aggregate capacity of STS-48c or VC4-16c.

Each ML100T-12 supports up to six FECs and one POS channel. Each ML100X-8 supports up to four FECs and one POS channel. A maximum of four Fast Ethernet ports can bundle into one Fast Ethernet Channel (FEC) and provide bandwidth scalability up to 400-Mbps full-duplex Fast Ethernet.

Each ML1000-2 supports up to two port channels, including the POS channel. A maximum of two Gigabit Ethernet ports can bundle into one Gigabit Ethernet Channel (FEC) and provide 2-Gbps full-duplex aggregate capacity on the ML1000-2.

Each ML-MR-10 card supports up to ten port channel interfaces. A maximum of ten Gigabit Ethernet ports can be added into one Port-Channel.

Note If the number of POS ports configured on the ML-MR-10 are 26, the MLMR-10 card supports two port channel interfaces. However, a maximum of ten Gigabit Ethernet ports can be added into one port channel.

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.

Caution Before a physical interface is removed from an EtherChannel (port channel) interface, the physical interface must be disabled. To disable a physical interface, use the **shutdown** command in interface configuration mode.

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

Note Policing is not supported on port channel interfaces.

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

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** *channel-number*	Creates the EtherChannel interface. You can configure up to 6 FECs on the ML100T-12, 4 FECs on the ML100X-8, and 1 GEC on the ML1000-2.
Step 2	Router(config-if)# **ip address** *ip-address subnet-mask*	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*.

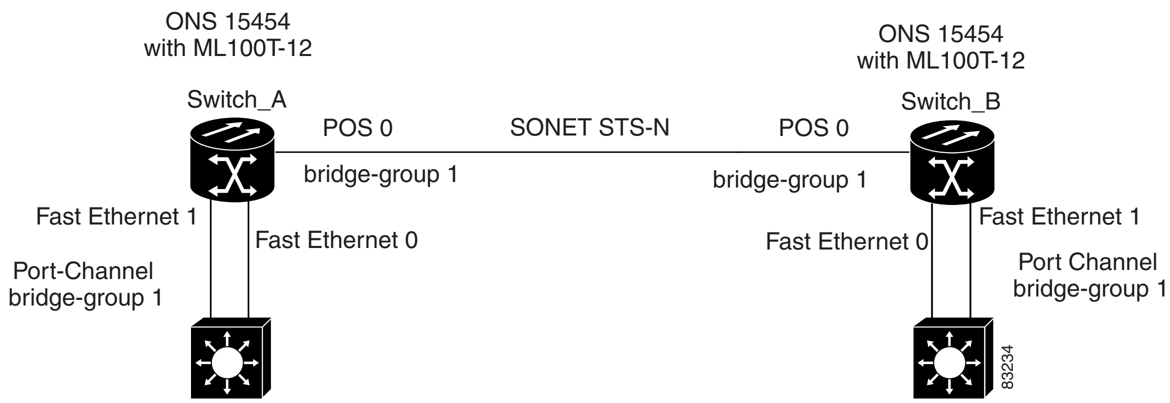
To assign Ethernet interfaces to the EtherChannel, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface fastethernet** *number* or Router(config)# **interface gigabitethernet** *number*	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 Ethernet interface on the system to the EtherChannel, but both interfaces must be either FEC or GEC.
Step 2	Router(config-if)# **channel-group** *channel-number*	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

Figure 10-1 shows an example of EtherChannel. The associated commands are provided in Example 10-1 (Switch A) and Example 10-2 (Switch B).

Figure 10-1 EtherChannel Example



Example 10-1 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 10-2 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 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`** *`channel-number`*	Creates the POS channel interface. You can configure one POS channel on the ML-Series card.
Step 2	`Router(config-if)#` **`ip address`** *`ip-address subnet-mask`*	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:

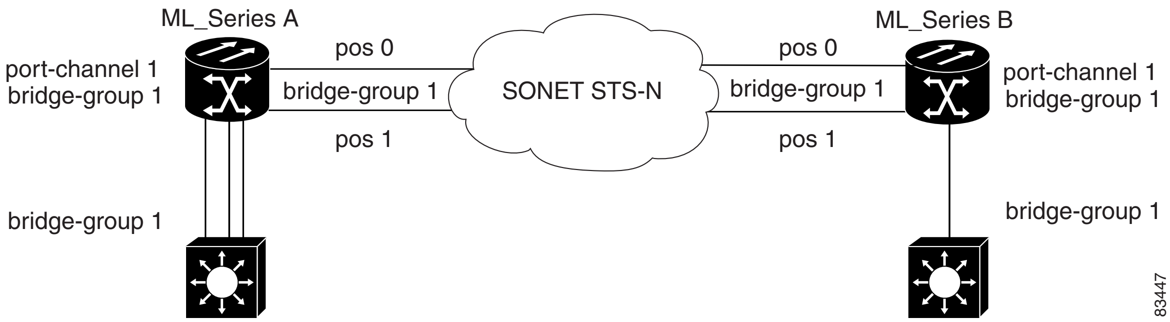
	Command	Purpose
Step 1	`Router(config)#` **`interface pos`** *`number`*	Enters the interface configuration mode to configure the POS interface that you want to assign to the POS channel.
Step 2	`Router(config-if)#` **`channel-group`** *`channel-number`*	Assigns the POS interface to the POS channel. The channel number must be the same channel number that you assigned to the POS channel interface.

	Command	Purpose
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

Figure 10-2 shows an example of POS channel configuration. The associated code is provided in Example 10-3 (Switch A) and Example 10-4 (Switch B).

Figure 10-2 POS Channel Example



Example 10-3 Switch A Configuration

```
bridge irb
bridge 1 protocol ieee
!
!
interface Port-channel1
 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 10-4 Switch B Configuration

```
bridge irb
bridge 1 protocol ieee
```

```
!
!
interface Port-channel1
 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 IEEE 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 IEEE 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:

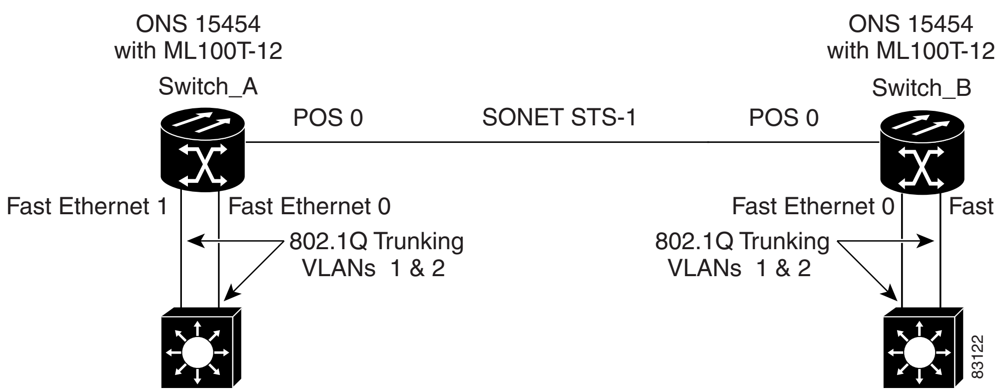
	Command	Purpose
Step 1	`Router(config)#` **`interface port-channel`** *`channel-number.subinterface-number`*	Configures the subinterface on the created port channel.
Step 2	`Router(config-subif)#` **`encapsulation dot1q`** *`vlan-id`*	Assigns the IEEE 802.1Q encapsulation to the subinterface.
Step 3	`Router(config-subif)#` **`bridge-group`** *`bridge-group-number`*	Assigns the subinterface to a bridge group.

	Command	Purpose
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 10-3 shows an example of encapsulation over EtherChannel. The associated code is provided in Example 10-5 (Switch A) and Example 10-6 (Switch B).

Figure 10-3 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 IEEE 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 10-5 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 10-6 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 10-7 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.
```

Understanding Link Aggregation Control Protocol

In Software Release 8.0.0, and later, ML100T-12, ML1000-2, ML100T-8, and CE-100T-8 cards can utilize the link aggregation control protocol (LACP) to govern reciprocal peer packet transmission with respect to LACP's detection of flawed packets. The cards' ports transport a signal transparently (that is, without intervention or termination). However, this transparent packet handling is done only if the LACP is not configured for the ML series card.

Passive Mode and Active Mode

Passive or active modes are configured for a port and they differ in how they direct a card to transmit packets: In passive mode, the LACP resident on the node transmits packets only after it receives reciprocal valid packets from the peer node. In active mode, a node transmits packets irrespective of the LACP capability of its peer.

LACP Functions

LACP performs the following functions in the system:

- Maintains configuration information in order to control aggregation
- Exchanges configuration information with other peer devices
- Attaches or detaches ports from the link aggregation group based on the exchanged configuration information
- Enables data flow when both sides of the aggregation group are synchronized

In addition, LACP provides the following benefits:

- Logical aggregation of bandwidth
- Load balancing
- Fault tolerance

LACP Parameters

LACP utilizes the following parameters to control aggregation:

System Identifier—A unique identification assigned to each system. It is the concatenation of the system priority and a globally administered individual MAC address.

Port Identification—A unique identifier for each physical port in the system. It is the concatenation of the port priority and the port number.

Port Capability Identification—An integer, called a key, that identifies one port’s capability to aggregate with another port. There are two types of key: administrative and operational. An administrative key is configured by the network administrator, and an operational key is assigned by LACP to a port based on its aggregation capability.

Aggregation Identifier—A unique integer that is assigned to each aggregator and is used for identification within the system.

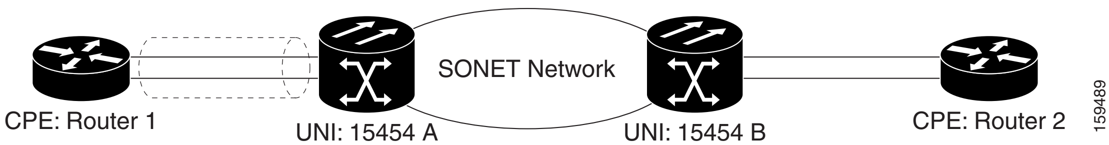
LACP Usage Scenarios

In Software Release 8.0.0, and later, LACP functions on ML-Series cards in termination mode and on the CE-Series cards in transparent mode.

Termination Mode

In termination mode, the link aggregation bundle terminates or originates at the ML card. To operate in this mode, LACP should be configured on the Ethernet interface. One protect SONET or SDH circuit can carry the aggregated Ethernet traffic of the bundle. The advantage of termination mode over transparent mode is that the network bandwidth is not wasted. However. the disadvantage is that there is no card protection between the CPE and UNI (ONS 15454) because all the links in the ML card bundle belong to the same card.

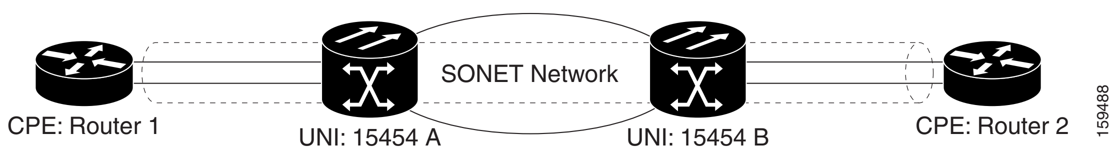
Figure 10-4 LACP Termination Mode Example



Transparent Mode

In Figure 10-5, the link aggregation bundle originates at router 1 and terminates at router 2. Transparent mode is enabled when the LACP packets are transmitted without any processing on a card. While functioning in this mode, the CE-100T-8 cards pass through LACP packets transparently so that the two CPE devices perform the link aggregation. To operate in this mode, no LACP configuration is required on the CE-100T-8 cards.

Figure 10-5 LACP Transparent Mode Example



Configuring LACP

To configure LACP over the EtherChannel, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`int port`** *`<interface-number>`*	Accesses the port interface where you will create the LACP.
Step 2	`Router(config-if)#` **`int fa`** *`<facility-number>`*	Access the facility number on the port.
Step 3	`Router(config-if)#` **`channel`**	Accesses the channel group of commands.
Step 4	`Router(config-if)#` **`channel-group`** *`<channel-number>`* **`mode ?`**	Queries the current mode of the channel group. Options include active and passive.

	Command	Purpose
Step 5	Router(config-if)# **channel-group** *<channel-number>* **mode active**	Places the channel group in active mode.
Step 6	Router(config-if)# **exit**	Exits the channel group configuration.
Step 7	Router(config-if)# **int fa** *<facility-number>*	Accesses the facility.
Step 8	Router(config-if)# **lacp-port**	Access the link aggregation control protocol commands for the port.
Step 9	Router(config-if)# **lacp port-priority** *<priority number>*	Sets the LACP port's priority. Range of values is from 1 through 65535. For example, `lacp port-priority 100`
Step 10	Router(config-if)# **exit**	Exits the port's configuration mode.
Step 11	Router(config)#**lacp sys**	Accesses the system LACP settings.
Step 12	Router(config)#**lacp system-priority** *<system priority>*	Sets the LACP system priority in a range of values from 1 through 65535. For example, `lacp system-priority 100`
Step 13	Router(config)# **exit**	Exits the global configuration mode.
Step 14	Router# **copy running-config startup-config**	(Optional) Saves the configuration changes to NVRAM.

In Example 10-8, the topology includes two nodes with a GEC or FEC transport between them. This example shows one GEC interface on Node 1. (Up to four similar types of links per bundle are supported.)

Example 10-8 LACP Configuration Example

```
ML2-Node1#sh run int gi0
Building configuration...

Current configuration : 150 bytes
!
interface GigabitEthernet0
 no ip address
 no keepalive
 duplex auto
 speed auto
 negotiation auto
 channel-group 1 mode active
 no cdp enable
end

ML2-Node1#
ML2-Node1#sh run int por1
Building configuration...

Current configuration : 144 bytes
!
interface Port-channel1
 no ip address
 no negotiation auto
 service instance 30 ethernet[1]
```

1. This is optional, required only when the IEEE 802.1q configuration is needed.

```
  encapsulation dot1q 30[1]
  bridge-domain 30
 !
end

ML2-Node1#
ML2-Node1#sh lacp int
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 1
                            LACP port     Admin     Oper    Port     Port
Port      Flags   State     Priority      Key       Key     Number   State
Gi0       SA      bndl      32768         0x1       0x1     0x5      0x3D
ML2-Node1#
Configuration  remains same for the ML2-Node2 also.
```

Load Balancing on the ML-Series cards

The load balancing for the Ethernet traffic on the portchannel is performed while sending the frame through a port channel interface based on the source MAC and destination MAC address of the Ethernet frame.

On a 2 port port channel interface, the Unicast Ethernet traffic (Learned MAC with unicast SA and DA) is transmitted on either first or second member of the port-channel based on the result of the "Exclusive OR" (XOR) operation applied on the second least significant bits (bit 1) of DA-MAC and SA-MAC. So, if the "XOR" result of the Ethernet frames SA-MAC second least significant bit and DA-MAC second least significant bit is 0 then the frame is sent on the first member and if the result is 1 then the frame is transmitted on the second member port of the port channel.

Table 10-1 MAC Based - 2- Port Channel Interface

Second Least Significant bit of the MAC-DA	Second Least Significant bit of the MAC-SA	XOR Result	Used Member Interface for the Frame Forwarding to the EtherChannel and/or Port Channel
0	0	0	Port 1
0	1	1	Port 2
1	0	1	Port 2
1	1	0	Port 1

Table 10-2 IP Based - 2- Port Channel Interface

Second Least Significant bit of the IP-DA	Second Least Significant bit of the IP-SA	XOR Result	Used Member Interface for the Frame Forwarding to the EtherChannel and/or Port Channel
0	0	1	Port 1
0	1	1	Port 2
1	0	1	Port 2
1	1	0	Port 1

The Flood Ethernet traffic (Unknown MAC, Multicast and Broadcast frames) is transmitted on the first active member of the port-channel.

The routed IP Unicast traffic from the ML-Series towards the port channel ports is transmitted on either interface based on the result of the "Exclusive OR" (XOR) operation applied on the second least significant bits of the source and destination IP address of the IP packet. So if the "XOR" result of the IP packets Source Address least significant bit and Destination Address least significant bit is 0 then the frame is on the first member port and if the result is 1 then the frame is transmitted on the second member port.

Table 10-3 MAC Based - 4-Port Channel Interface

Third Least Significant bit of the MAC-DA	Third Least Significant bit of the MAC-SA	Second Least Significant bit of the MAC-DA	Second Least Significant bit of the MAC-SA	XOR Result	Used Member Interface for the Frame Forwarding to the EtherChannel and/or Port Channel
0	0	0	0	00	First
0	0	0	1	01	Second
0	0	1	0	01	Second
0	0	1	1	00	First
0	1	0	0	10	Third
0	1	0	1	11	Fourth
0	1	1	0	11	Fourth
0	1	1	1	10	Second
1	0	0	0	10	Second
1	0	0	1	11	Third
1	0	1	0	11	Third

Table 10-3 MAC Based - 4-Port Channel Interface

Third Least Significant bit of the MAC-DA	Third Least Significant bit of the MAC-SA	Second Least Significant bit of the MAC-DA	Second Least Significant bit of the MAC-SA	XOR Result	Used Member Interface for the Frame Forwarding to the EtherChannel and/or Port Channel
1	0	1	1	10	Second
1	1	0	0	00	First
1	1	0	1	01	Second
1	1	1	0	01	Second
1	1	1	1	00	First

Table 10-4 IP Based - 4-Port Channel Interface

Third Least Significant bit of the IP-DA	Third Least Significant bit of the IP-SA	Second Least Significant bit of the IP-DA	Second Least Significant bit of the IP-SA	XOR Result	Used Member Interface for the Frame Forwarding to the EtherChannel and/or Port Channel
0	0	0	0	00	First
0	0	0	1	01	Second
0	0	1	0	01	Second
0	0	1	1	00	First
0	1	0	0	10	Third
0	1	0	1	11	Fourth
0	1	1	0	11	Fourth
0	1	1	1	10	Second
1	0	0	0	10	Second
1	0	0	1	11	Third
1	0	1	0	11	Third
1	0	1	1	10	Second
1	1	0	0	00	First
1	1	0	1	01	Second
1	1	1	0	01	Second
1	1	1	1	00	First

On the 4 port channel, the second and third least significant bits are used for load balancing.

The routed IP Multicast traffic from the ML-Series towards the RPR ring is transmitted on the first active member of the port channel.

Load Balancing on the ML-MR-10 card

The load balancing on the ML-MR-10 card can be configured through the following options:

- source and destination MAC addresses
- VLAN ID contained in the SVLAN (outer) tag

The default load balancing mechanism on ML-MR-10 card is the source and destination MAC address.

MAC address based load balancing

The MAC address based load balancing is achieved by performing "XOR" (exclusive OR) operation on the last 4 least significant bits of the source MAC address and the destination MAC address.

Table 10-5 displays the ethernet traffic with 4 Gigabit Ethernet members on the port channel interfaces.

Table 10-5 4 Gigabit Ethernet Port Channel Interface

XOR Result	Member Interface used for Frame Forwarding on the Port Channel Interface
0	member-0
1	member-1
2	member-2
3	member-0
4	member-1
5	member-2
6	member-0
7	member-1
8	member-2
9	member-0
10	member-1
11	member-2
12	member-0
13	member-1
14	member-2
15	member-0

Table 10-6 displays the ethernet traffic with 3 Gigabit Ethernet members on the port channel interfaces.

Table 10-6 3 Gigabit Ethernet Port Channel Interface

XOR Result	Member Interface used for Frame Forwarding on the Port Channel Interface
0	member-0
1	member-1
2	member-2
3	member-0
4	member-1
5	member-2
6	member-0
7	member-1
8	member-2
9	member-0
10	member-1
11	member-2
12	member-0
13	member-1
14	member-2
15	member-0

The member of the port channel interface depends on the order in which the Gigabit Ethernet becomes an active member of the port channel interface. The order in which the members are added to the port channel can be found using the show interface port channel <port channel number> command in the EXEC mode.

VLAN Based Load Balancing

VLAN based load balancing is achieved by using the last 4 least significant bits of the incoming VLAN ID in the outer VLAN.

Table 10-7 displays the ethernet traffic with 3 Gigabit Ethernet members on the port channel interfaces.

Table 10-7 3 Gigabit Ethernet members

Last 4 bits in VLAN	Member Interface used for the Frame forwarding on the Port-Channel Interface
0	member-0
1	member-1

Table 10-7 3 Gigabit Ethernet members

Last 4 bits in VLAN	Member Interface used for the Frame forwarding on the Port-Channel Interface
2	member-2
3	member-3
4	member-0
5	member-1
6	member-2
7	member-3
8	member-0
9	member-1
10	member-2
11	member-3
12	member-0
13	member-1
14	member-2
15	member-3

The member of the port channel interface depends on the order in which the Gigabit Ethernet becomes an active member of the port channel interface. The order in which the members are added to the port channel can be found using the show interface port-channel <port-channel number> command in the EXEC mode.

With the 4 Gigabit Ethernet members, if the incoming VLAN ID is 20, the traffic will be sent on member-0. If the incoming VLAN ID is 30, the traffic will be sent on member-2.

Configuration Commands for Load Balancing

Table 10-8 details the commands used to configure load balancing on the ML-Series cards and the ML-MR-10 card.

Table 10-8 Configuration Commands for Load Balancing

	Command	Purpose
Step 1	**Router(config) #int port-channel 10**	Accesses the port interface
Step 2	**Router(config-if)#load -balance vlan**	To change the load-balancing based on outer vlan

Table 10-8 Configuration Commands for Load Balancing

	Command	Purpose
Step 3	**Router(config)#exi**	tExits the global configuration mode.
Step 4	**Router# copy running-config startup-config**	(Optional) Saves the configuration changes to NVRAM.

Example 10-9 show command configuration

```
Configuration:
!
interface Port-channel10
 no ip address
 no negotiation auto
 load-balance vlan
 service instance 20 ethernet
  encapsulation dot1q 20
   bridge-domain 20
 !
 service instance 30 ethernet
  encapsulation dot1q 30
   bridge-domain 30
 !
!
!
interface GigabitEthernet1
 no ip address
 speed auto
 duplex auto
 negotiation auto
 channel-group 10
 no keepalive
!
interface GigabitEthernet2
 no ip address
 speed auto
 duplex auto
 negotiation auto
 channel-group 10
 no keepalive
!
interface GigabitEthernet9
 no ip address
 speed auto
 duplex auto
 negotiation auto
 channel-group 10
 no keepalive


Router#sh int port-channel 10
Port-channel10 is up, line protocol is up
  Hardware is GEChannel, address is 001b.54c0.2643 (bia 0000.0000.0000)
  MTU 9600 bytes, BW 2100000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
```

```
    No. of active members in this channel: 3
        Member 0 : GigabitEthernet9 , Full-duplex, 100Mb/s
        Member 1 : GigabitEthernet1 , Full-duplex, 1000Mb/s
        Member 2 : GigabitEthernet2 , Full-duplex, 1000Mb/s
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/225/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/120 (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 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     48 packets output, 19080 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 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
Router#
```

```
Router#show port-channel load-balance interface Port-channel 10 hash-table
Hash-value             Interface
  0                GigabitEthernet9
  1                GigabitEthernet1
  2                GigabitEthernet2
  3                GigabitEthernet9
  4                GigabitEthernet1
  5                GigabitEthernet2
  6                GigabitEthernet9
  7                GigabitEthernet1
  8                GigabitEthernet2
  9                GigabitEthernet9
  10                GigabitEthernet1
  11                GigabitEthernet2
  12                GigabitEthernet9
  13                GigabitEthernet1
  14                GigabitEthernet2
  15                GigabitEthernet9
Router#
```

CHAPTER 11

Configuring Networking Protocols

This chapter 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.

This chapter contains the following major sections:

Basic IP Routing Protocol Configuration

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.

To configure IP routing protocols to run on a Fast Ethernet, Gigabit Ethernet, or Packet-over-SONET/SDH (POS) interface, perform one of the following procedures, depending on the protocol you are configuring.

RIP

To configure the Routing Information Protocol (RIP), 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`** *`net-number`*	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 Enhanced Interior Gateway Routing Protocol (EIGRP), perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`router eigrp`** *`autonomous-system-number`*	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`** *`net-number`*	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 Open Shortest Path First (OSPF) protocol, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`router ospf`** *`process-ID`*	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`** *`net-address wildcard-mask`* **`area`** *`area-ID`*	Assigns an interface to a specific area. • The net-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 Border Gateway Protocol (BGP), perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`router bgp`** *`autonomous-system-number`*	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`** *`net-number`*	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

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

Note By default, IP routing is already enabled.

	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** *ip-routing-protocol*`	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 *Cisco IOS IP and IP Routing Configuration Guide.*
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 (Example 11-1) to disable routing.

Example 11-1 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 11-4
- Configuring OSPF, page 11-9
- Configuring EIGRP, page 11-20
- Configuring BGP, page 11-27
- Configuring IS-IS, page 11-29
- Configuring Static Routes, page 11-31

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 nonupdating 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 11-1 shows the default RIP configuration.

Table 11-1 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
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	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`ip routing`**	Enables IP routing. (Required only if IP routing is disabled.)
Step 3	`Router(config)#` **`router rip`**	Enables a RIP routing process, and enters router configuration mode.
Step 4	`Router(config-router)#` **`network`** *`network-number`*	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	`Router(config-router)#` **`neighbor`** *`ip-address`*	(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	`Router(config-router)#` **`offset list`** `{[`*`access-list-number`*` | `*`name`*`]} {`**`in`**` | `**`out`**`}` *`offset`* `[`*`type-number`*`]`	(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	`Router(config-router)#` **`timers basic`** *`update invalid holddown flush`*	(Optional) Adjusts routing protocol timers. Valid ranges for all timers are 0 to 4294967295 seconds. • update—The time (in seconds) between sending of routing updates. The default is 30 seconds. • invalid—The timer interval (in seconds) after which a route is declared invalid. The default is 180 seconds. • holddown—The time (in seconds) that must pass before a route is removed from the routing table. The default is 180 seconds. • flush—The amount of time (in seconds) for which routing updates are postponed. The default is 240 seconds.
Step 8	`Router(config-router)#` **`version`** `{`**`1`**` | `**`2`**`}`	(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 **ip rip** {**send** | **receive**} **version** {**1** | **2** | **1 2**} to control what versions are used for sending and receiving on interfaces.
Step 9	`Router(config-router)#` **`no auto summary`**	(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	`Router(config-router)#` **`no validate-update-source`**	(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	`Router(config-router)#` **`output-delay`** *`delay`*	(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	`Router(config-router)#` **`end`**	Returns to privileged EXEC mode.

	Command	Purpose
Step 13	Router# **show ip protocols**	Verifies your entries.
Step 14	Router# **copy running-config startup-config**	(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 11-2).

Example 11-2 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 (Example 11-3).

Example 11-3 show ip rip database Command Output

```
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 message-digest key (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`** *`interface-id`*	Enters interface configuration mode, and specifies the interface to configure.
Step 3	`Router(config-if)#` **`ip rip authentication key-chain`** *`name-of-chain`*	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`** `[`*`interface-id`*`]`	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`** *`interface-id`*	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 3	`Router(config-if)#` **`ip address`** *`ip-address subnet-mask`*	Configures the IP address and IP subnet.

	Command	Purpose
Step 4	Router(config-if)# **ip summary-address rip** *ip-address ip-network-mask*	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** *interface-id*	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 the 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 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 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 11-2 shows the default OSPF configuration.

Table 11-2 Default OSPF Configuration

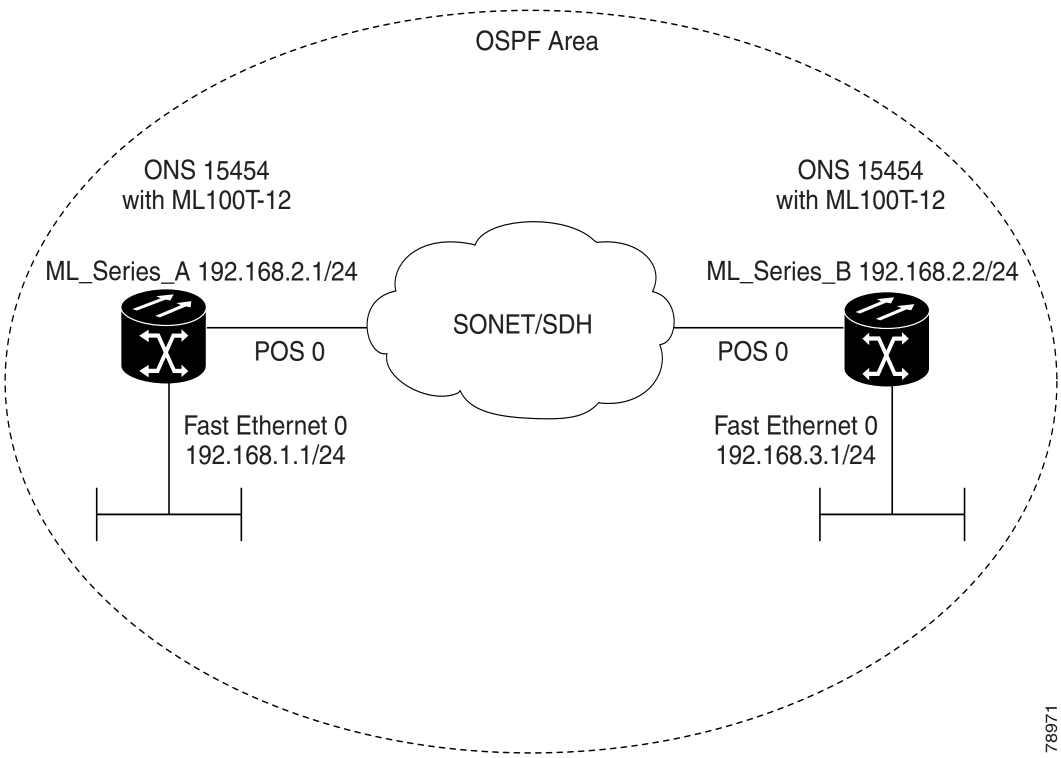
Feature	Default Setting
Interface parameters	Cost: No default cost predefined. Retransmit interval: 5 seconds. Transmit delay: 1 second. Priority: 1. Hello interval: 10 seconds. Dead interval: 4 times the hello interval. No authentication. No password specified. MD5 authentication disabled.
Area	Authentication 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 dist3 (routes from other routing domains): 110
OSPF database filter	Disabled. All outgoing link-state advertisements (LSAs) are flooded to the interface.
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.

Table 11-2 Default OSPF Configuration (continued)

Feature	Default Setting
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 11-1 shows an example of an IP routing protocol using OSPF.

Figure 11-1 IP Routing Protocol Example Using OSPF



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	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`router ospf`** *`process-id`*	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	`Router(config)#` **`network`** *`address wildcard-mask`* **`area`** *`area-id`*	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	`Router(config)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Router#` **`show ip protocols`**	Verifies your entries.
Step 6	`Router#` **`copy running-config startup-config`**	(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 11-4 shows an example of configuring an OSPF routing process. In the example, a process number of 1 is assigned. Example 11-5 shows the output of the command used to verify the OSPF process ID.

Example 11-4 Configuring an OSPF Routing Process

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

Example 11-5 show ip protocols Privileged EXEC Command Output

```
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.
Step 7	`Router(config-if)#` **`ip ospf hello-interval`** *`seconds`*	(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	`Router(config-if)#` **`ip ospf dead-interval`** *`seconds`*	(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	`Router(config-if)#` **`ip ospf authentication-key`** *`key`*	(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	`Router(config-if)#` **`ip ospf message digest-key`** *`keyid`* **`md5`** *`key`*	(Optional) Enables authentication. • keyid—Identifier from 1 to 255. • key—Alphanumeric password of up to 16 bytes.

	Command	Purpose
Step 11	Router(config-if)# **ip ospf database-filter all out**	(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	Router(config-if)# **end**	Returns to privileged EXEC mode.
Step 13	Router# **show ip ospf interface** [*interface-name*]	Displays OSPF-related interface information.
Step 14	Router# **copy running-config startup-config**	(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 11-6 shows the output of the **show ip ospf interface** privileged EXEC command.

Example 11-6 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 NSSAs. Stub areas are areas into which information about external routes is not sent. Instead, the 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	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`router ospf`** *`process-id`*	Enables OSPF routing, and enters router configuration mode.
Step 3	`Router(config)#` **`area`** *`area-id`* **`authentication`**	(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	`Router(config)#` **`area`** *`area-id`* **`authentication message-digest`**	(Optional) Enables MD5 authentication on the area.
Step 5	`Router(config)#` **`area`** *`area-id`* **`stub`** `[`**`no-summary`**`]`	(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	`Router(config)#` **`area`** *`area-id`* **`nssa`** `{`**`no-redistribution`** `|` **`default-information-originate`** `|` **`no-summary`**`}`	(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: • **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	`Router(config)#` **`area`** *`area-id`* **`range`** *`address-mask`*	(Optional) Specifies an address range for which a single route is advertised. Use this command only with area border routers.
Step 8	`Router(config)#` **`end`**	Returns to privileged EXEC mode.
Step 9	`Router#` **`show ip ospf`** `[`*`process-id`*`]`	Displays information about the OSPF routing process in general or for a specific process ID to verify configuration.
Step 10	`Router#` **`copy running-config startup-config`**	(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 11-7 shows the output of the **show ip ospf database** and the **show ip ospf** privileged EXEC commands.

Example 11-7 show ip ospf database and show ip ospf Privileged EXEC Command Ouputs

```
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 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 (intra-area), 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	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# router ospf process-id`	Enables OSPF routing, and enters router configuration mode.
Step 3	`Router(config)# summary-address address-mask`	(Optional) Specifies an address and IP subnet mask for redistributed routes so that only one summary route is advertised.
Step 4	`Router(config)# area area-id virtual-link router-id [hello-interval seconds] [retransmit-interval seconds] [trans] {[authentication-key key] | [message-digest-key key-id md5 key]}`	(Optional) Establishes a virtual link and set its parameters. See the "OSPF Interface Parameters" section on page 11-13 for parameter definitions and Table 11-2 on page 11-10 for virtual link defaults.
Step 5	`Router(config)# default-information originate [always] [metric metric-value] [metric-type type-value] [route-map map-name]`	(Optional) Forces the ASBR to generate a default route into the OSPF routing domain. Parameters are all optional.
Step 6	`Router(config)# ip ospf name-lookup`	(Optional) Configures DNS name lookup. The default is disabled.
Step 7	`Router(config)# ip auto-cost reference-bandwidth ref-bw`	(Optional) Specifies an address range for which a single route will be advertised. Use this command only with area border routers.
Step 8	`Router(config)# distance ospf {[inter-area dist1] | [inter-area dist2] | [external dist3]}`	(Optional) Changes the OSPF distance values. The default distance for each type of route is 110. The range is 1 to 255.

	Command	Purpose
Step 9	Router(config)# **passive-interface** *type number*	(Optional) Suppresses the sending of hello packets through the specified interface.
Step 10	Router(config)# **timers spf** *spf-delay spf-holdtime*	(Optional) Configures route calculation timers. • spf-delay—Enter an integer from 0 to 65535. The default is 5 seconds; 0 means no delay. • spf-holdtime—Enter an integer from 0 to 65535. The default is 10 seconds; 0 means no delay.
Step 11	Router(config)# **ospf log-adj-changes**	(Optional) Sends syslog message when a neighbor state changes.
Step 12	Router(config)# **end**	Returns to privileged EXEC mode.
Step 13	Router# **show ip ospf** [*process-id* [*area-id*]] **database**	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 11-19.
Step 14	Router# **copy running-config startup-config**	(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 four-minute default pacing interval, and you do 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	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **router ospf** *process-id*	Enables OSPF routing, and enters router configuration mode.
Step 3	Router(config)# **timers lsa-group-pacing** *seconds*	Changes the group pacing of LSAs.
Step 4	Router(config)# **end**	Returns to privileged EXEC mode.
Step 5	Router# **show running-config**	Verifies your entries.
Step 6	Router# **copy running-config startup-config**	(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	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)# interface loopback 0`	Creates a loopback interface, and enters interface configuration mode.
Step 3	`Router(config)#` **`ip address`** *`address mask`*	Assigns an IP address to this interface.
Step 4	`Router(config)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Router#` **`show ip interface`**	Verifies your entries.
Step 6	`Router#` **`copy running-config startup-config`**	(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 11-3 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 11-3 Show IP OSPF Statistics Commands

Command	Purpose
`Router(config)#` **`show ip ospf`** `[`*`process-id`*`]`	Displays general information about OSPF routing processes.
`Router(config)#` **`show ip ospf`** `[`*`process-id`*`]` **`database`** `[`**`router`**`] [`*`link-state-id`*`]`	Displays lists of information related to the OSPF database.
`Router(config)# show ip ospf border-routes`	Displays the internal OSPF routing ABR and ASBR table entries.
`Router(config)#` **`show ip ospf interface`** `[`*`interface-name`*`]`	Displays OSPF-related interface information.
`Router(config)#` **`show ip ospf neighbor`** `[`*`interface-name`*`] [`*`neighbor-id`*`]` **`detail`**	Displays OSPF interface neighbor information.
`Router(config)# show ip ospf virtual-links`	Displays OSPF-related virtual links information.

Configuring EIGRP

Enhanced IGRP (EIGRP) is a Cisco proprietary enhanced version of the Interior Gateway Routing Protocol (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 the following 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 do not need to 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 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 11-4 shows the default EIGRP configuration.

Table 11-4 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: • 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 11-4 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	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`router eigrp`** *`autonomous-system-number`*	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	`Router(config)#` **`network`** *`network-number`*	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	`Router(config)#` **`eigrp log-neighbor-changes`**	(Optional) Enables logging of EIGRP neighbor changes to monitor routing system stability.
Step 5	`Router(config)#` **`metric weights`** *`tos k1 k2 k3 k4 k5`*	(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	Router(config)# **offset list** [{*access-list-number* | *name*}] { **in** | **out** } *offset* [*type-number*]	(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	Router(config)# **no auto-summary**	(Optional) Disables automatic summarization of subnet routes into network-level routes.
Step 8	Router(config)# **ip summary-address eigrp** *autonomous-system-number address-mask*	(Optional) Configures a summary aggregate.
Step 9	Router(config)# **end**	Returns to privileged EXEC mode.
Step 10	Router# **show ip protocols**	Verifies your entries.
Step 11	Router# **copy running-config startup-config**	(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 11-8 shows the output for the **show ip protocols** privileged EXEC command.

Example 11-8 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	`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)# `**`ip bandwidth-percent eigrp`** *`autonomous-system-number percent`*	(Optional) Configures the maximum percentage of bandwidth that can be used by EIGRP on an interface. The default is 50 percent.
Step 4	`Router(config)# `**`ip summary-address eigrp`** *`autonomous-system-number address mask`*	(Optional) Configures a summary aggregate address for a specified interface (not usually necessary if autosummary is enabled).
Step 5	`Router(config)# `**`ip hello-interval eigrp`** *`autonomous-system-number seconds`*	(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	`Router(config)# `**`ip hold-time eigrp`** *`autonomous-system-number seconds`*	(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	`Router(config)# `**`no ip split-horizon eigrp`** *`autonomous-system-number`*	(Optional) Disables split horizon to allow route information to be advertised by a router out any interface from which that information originated.
Step 8	`Router# `**`end`**	Returns to privileged EXEC mode.
Step 9	`Router# `**`show ip eigrp interface`**	Displays the interfaces that EIGRP is active on and information about EIGRP relating to those interfaces.
Step 10	`Router# `**`copy running-config startup-config`**	(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 11-9 shows the output of the **show ip eigrp interface** privileged EXEC command.

Example 11-9 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
```

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	`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 authentication mode eigrp`** *`autonomous-system-number`* **`md5`**	Enables MD5 authentication in IP EIGRP packets.
Step 4	`Router(config-if)#` **`ip authentication key-chain eigrp`** *`autonomous-system-number key-chain`*	Enables authentication of IP EIGRP packets.
Step 5	`Router(config-if)#` **`exit`**	Returns to global configuration mode.
Step 6	`Router(config)#` **`key chain`** *`name-of-chain`*	Identifies a key chain and enter key-chain configuration mode. Match the name configured in Step 4.
Step 7	`Router(config-keychain)#` **`key`** *`number`*	In key-chain configuration mode, identifies the key number.
Step 8	`Router(config-keychain)#` **`key-string`** *`text`*	In key-chain key configuration mode, identifies the key string.
Step 9	`Router(config-keychain-key)#` **`accept-lifetime`** *`start-time`* {**`infinite`** | *`end-time`* | **`duration`** *`seconds`*}	(Optional) Specifies the time period during which the key can be received. The *start-time* and *end-time* syntax can be either *hh*:*mm*:*ss Month date year* or *hh*:*mm*:*ss date Month year*. The default *start-time* (and earliest acceptable day) is January 1, 1993. The default *end-time* and **duration** is infinite.
Step 10	`Router(config-keychain-key)#` **`send-lifetime`** *`start-time`* {**`infinite`** | *`end-time`* | **`duration`** *`seconds`*}	(Optional) Specifies the time period during which the key can be sent. The *start-time* and *end-time* syntax can be either *hh*:*mm*:*ss Month day year* or *hh*:*mm*:*ss day Month year*. The default *start-time* (and earliest acceptable day) is January 1, 1993. The default *end-time* and **duration** is **infinite**.
Step 11	`Router(config)#` **`end`**	Returns to privileged EXEC mode.
Step 12	`Router#` **`show key chain`**	Displays authentication key information.
Step 13	`Router#` **`copy running-config startup-config`**	(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 11-5 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 11-5 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 11-10 shows the output of the **show ip eigrp interface** privileged EXEC command.
Example 11-11 shows the output of the **show ip eigrp neighbors** privileged EXEC command.
Example 11-12 shows the output of the **show ip eigrp topology** privileged EXEC command.
Example 11-13 shows the output of the **show ip eigrp traffic** privileged EXEC command.

Example 11-10 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 11-11 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 11-12 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 11-13 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`** *`autonomous-system`*	Defines BGP as the routing protocol and starts the BGP routing process.
Step 3	`Router(config-router)#` **`network`** *`network-number`* `[`**`mask`** *`network-mask`*`] [`**`route-map`** *`route-map-name`*`]`	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 11-14 shows and example of configuring BGP routing.

Example 11-14 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 11-6 lists some common EXEC commands used to view the BGP configuration. Example 11-15 shows the output of the commands listed in Table 11-6.

Table 11-6 BGP Show Commands

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

Example 11-15 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 NLRIs 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 Intermediate System-to-Intermediate System (IS-IS) routing, perform the following steps, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router isis [tag]`	Defines IS-IS as the IP routing protocol.
Step 2	`Router(config-router)# net network-entity-title`	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 interface-type interface-id`	Enters interface configuration mode.
Step 4	`Router(config-if)# ip address ip-address mask`	Assigns an IP address to the interface.
Step 5	`Router(config-if)# ip router isis [tag]`	Specifies that this interface should run IS-IS.
Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.

Example 11-16 shows an example of IS-IS routing configuration.

Example 11-16 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 11-7. Example 11-17 shows examples of the commands in Table 11-7 and their output.

Table 11-7 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 11-17 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`** *`prefix mask`* `{` *`address`* `|` *`interface`* `} [`*`distance`*`]`	Establishes a static route. Illustrated in Example 11-18.
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 11-18 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. Use the show ip route privileged EXEC command to view information about the static IP route (Example 11-19).

Example 11-19 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 11-8 shows the default administrative distances for these routing protocols.

Table 11-8 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 (Example 11-20). 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.

Example 11-20 show ip route 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
```

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 11-9 to clear routes or display status.

Table 11-9 Commands to Clear IP Routes or Display Route Status

Command	Purpose
Router# **clear ip route** {**network** [*mask* | *****]}	Clears one or more routes from the IP routing table.
Router# **show ip protocols**	Displays the parameters and state of the active routing protocol process.
Router# **show ip route** [{*address* [*mask*] [**longer-prefixes**] | [*protocol* [*process-id*]}]	Displays the current state of the routing table.
Router# **show ip interface** *interface*	Displays detailed information about the interface.
Router# **show ip interface brief**	Displays summary status information about all interfaces.
Router# **show ip route summary**	Displays the current state of the routing table in summary form.
Router# **show ip route supernets-only**	Displays supernets.
Router# **show ip cache**	Displays the routing table used to switch IP traffic.
Router# **show route-map** [*map-name*]	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`** *`type number`*	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`** *`rendezvous-point ip-address`*	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 11-10, from privileged EXEC mode.

Table 11-10 IP Multicast Routing Show Commands

Command	Purpose
`Router# show ip mroute`	Shows the complete multicast routing table and the combined statistics of packets processed.
`Router# show ip pim neighbor`	When used in EXEC mode, lists the PIM neighbors discovered by the Cisco IOS software.
`Router# show ip pim interface`	Displays information about interfaces configured for PIM.
`Router# show ip pim rp`	When used in EXEC mode, displays the active rendezvous points (RPs) that are cached with associated multicast routing entries.

CHAPTER 12

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:

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 maximum transmission unit (MUTT) 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`** *`bridge-group`* **`protocol {ieee	rstp}`**	Defines one or more bridge groups.
Step 2	`Router(config)#` **`interface`** *`type number`*	Enters interface configuration mode.
Step 3	`Router(config-if)#` **`bridge-group`** *`bridge-group`*	Assigns the interface to the specified bridge group.
Step 4	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.

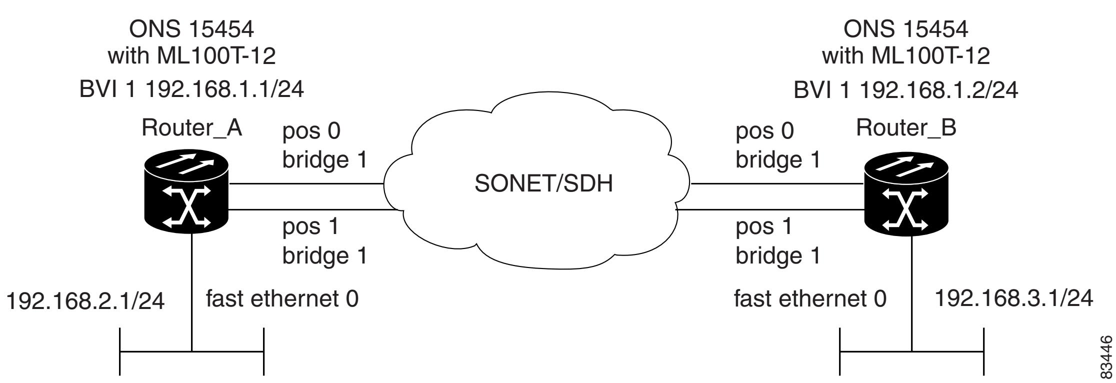
To enable and configure IRB and BVI, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`bridge irb`**	Enables IRB. Allows bridging of traffic.
Step 2	`Router(config)#` **`interface bvi`** *`bridge-group`*	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`** *`ip-address ip-address-subnet-mask`*	Configures IP addresses on routed interfaces.
Step 4	`Router(config-if)#` **`exit`**	Exits the interface configuration mode.
Step 5	`Router(config)#` **`bridge`** *`bridge-group`* **`route`** *`protocol`*	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.

IRB Configuration Example

Figure 12-1 shows an example of IRB configuration. Example 12-1 shows the configuration code for Router A, and Example 12-2 shows the configuration code for Router B.

Figure 12-1 Configuring IRB



Example 12-1 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 12-2 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 12-1 shows the privileged EXEC commands for monitoring and verifying IRB.

Table 12-1 Commands for Monitoring and Verifying IRB

Command	Purpose
Router# **show interfaces bvi** ***bvi-interface-number***	Shows BVI information, such as the BVI MAC address and processing statistics. The ***bvi-interface-number*** is the number of the bridge group assigned to the BVI interface.
Router# **show interfaces [*type-number*] irb**	Shows BVI information for the following: • 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 12-3 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 12-1 describes significant fields shown in the display.

Table 12-2 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.

CHAPTER 13

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:

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 its 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. 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*** *vrf-name*	Enters VRF configuration mode and assigns a VRF name.
Step 2	`Router(config-vrf)#` ***rd*** *route-distinguisher*	Creates a VPN route distinguisher (RD). An RD creates routing and forwarding tables and specifies the default route distinguisher for a VPN. The RD is added to the beginning of the customer's IPv4 prefixes to change them into globally unique VPN-IPv4 prefixes. Either RD is an ASN-relative RD, in which case it is composed of an autonomous system number and an arbitrary number, or it is an IP-address-relative RD, in which case it is composed of an IP address and an arbitrary number. You can enter a *route-distinguisher* in either of these formats: 16-bit AS number: your 32-bit number For example, 101:3. 32-bit IP address: your 16-bit number For example, 192.168.122.15:1.
Step 3	`Router(config-vrf)#` **route-target** {**import** | **export** | **both**} *route-distinguisher*	Creates a list of import and/or export route target communities for the specified VRF.
Step 4	`Router(config-vrf)#` **import map** *route-map*	(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** *vrf-name*	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 13-1 shows an example of configuring a VRF. In the example, the VRF name is customer_a, the route-distinguisher is 1:1, and the interface type is Fast Ethernet, number 0.1.

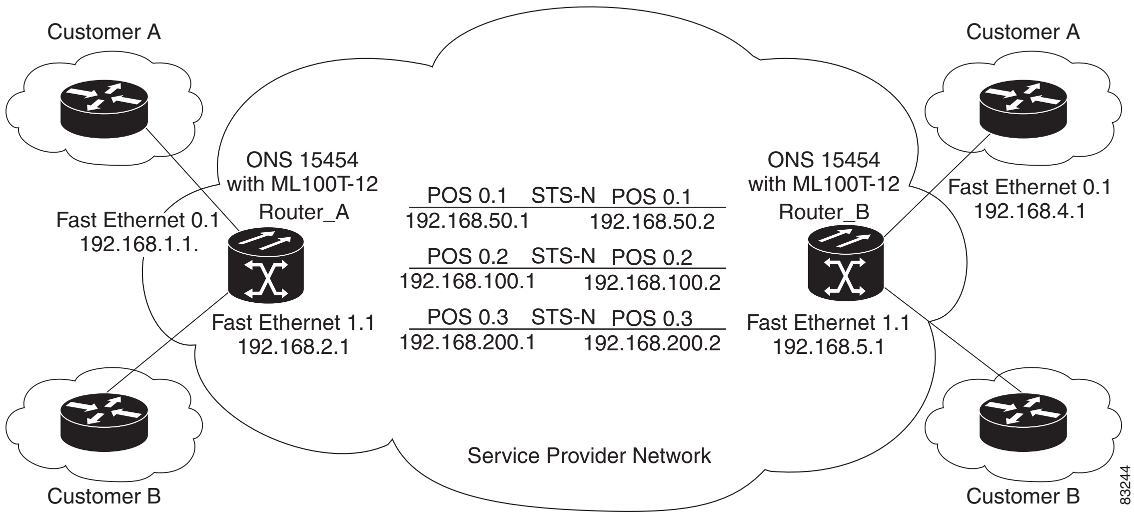
Example 13-1 Configuring a VRF

```
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 13-1 shows an example of a VRF Lite configuration. The configurations for Router A and Router B are provided in Example 13-2 and Example 13-3 on page 13-4, respectively. The associated routing tables are shown in Example 13-4 on page 13-5 through Example 13-9 on page 13-7.

Figure 13-1 VRF Lite—Sample Network Scenario



Example 13-2 **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 13-3 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 13-4 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 13-5 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 13-6 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 13-7 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 13-8 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 13-9 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 13-1 shows the privileged EXEC commands for monitoring and verifying VRF Lite.

Table 13-1 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** *ip-address*	Pings an ip address that has a specific VRF.

CHAPTER 14

Configuring Quality of Service

This chapter describes the quality of service (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:

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

- Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2
- Cisco IOS Quality of Service Solutions Command Reference, Release 12.2

Understanding QoS

QoS is the ability of the network to provide better or special treatment to a set of services to the detriment of less critical services. The ML-Series card uses QoS to dynamically allocate transmission bandwidth for the different services it multiplexes onto the SONET/SDH circuit. Through QoS, you can configure the ML-Series card to provide different levels of treatment to the different services. The different levels are defined through the service elements of bandwidth, including loss and delay. A service-level agreement (SLA) is a guaranteed level of these service elements.

The QoS mechanism has three basic steps. It classifies types of traffic, specifies what action to take against a type of traffic, and specifies where the action should take place. The following sections explain how the ML-Series card accomplishes these steps for unicast traffic. QoS for priority-multicast traffic and traffic with unknown destination addresses is handled with a different mechanism, detailed in the "Understanding Multicast QoS and Priority Multicast Queuing" section on page 14-24.

Priority Mechanism in IP and Ethernet

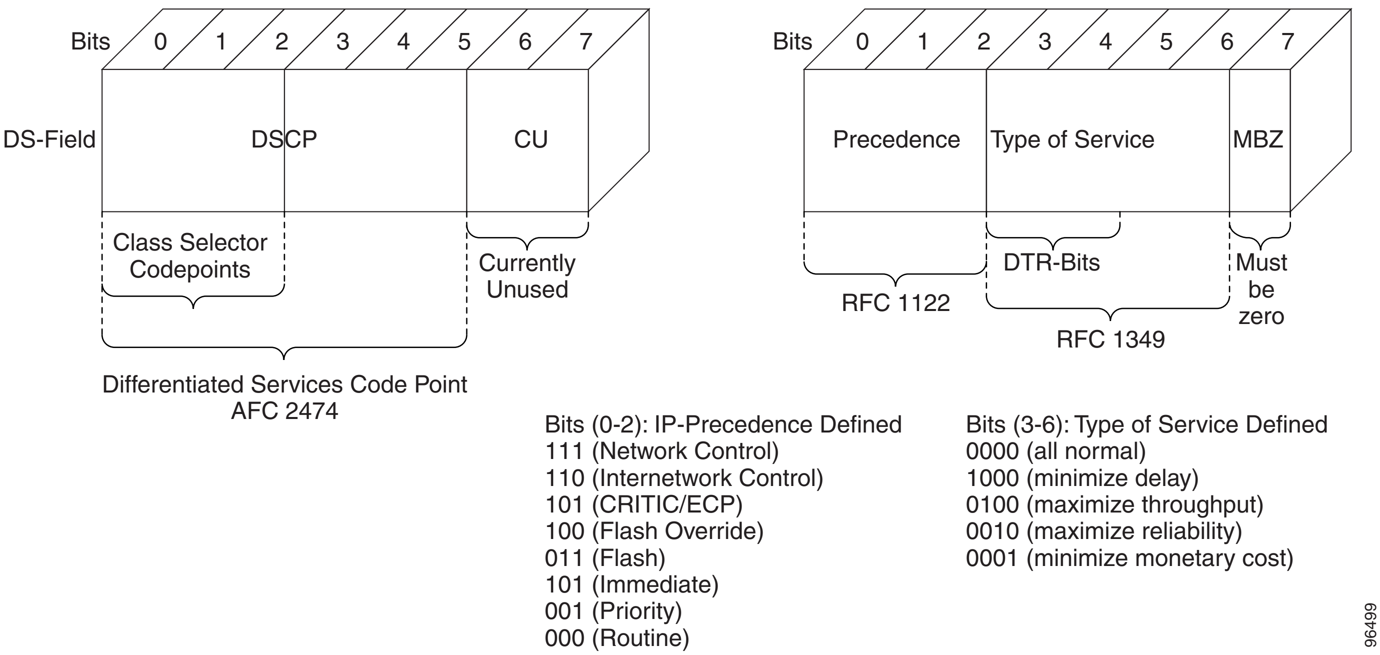
For any QoS service to be applied to data, there must be a way to mark or identify an IP packet or an Ethernet frame. When identified, a specific priority can be assigned to each individual IP packet or Ethernet frame. The IP Precedence field or the IP Differentiated Services Code Point (DSCP) field prioritizes the IP packets, and the Ethernet class of service (IEEE 802.1p defined class of service [CoS]) is used for the Ethernet frames. IP precedence and Ethernet CoS are further described in the following sections.

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 (IETF 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 SLA.

IP DSCP uses the six bits in the IPv4 header to specify class of service for each IP packet (IETF RFC 2474). Figure 14-1 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 SLA.

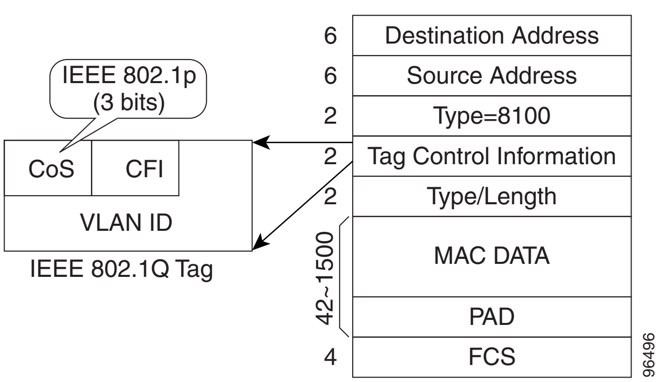
Figure 14-1 IP Precedence and DSCP



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 IEEE 802.1Q header are commonly referred to as the IEEE 802.1p bits. There are three CoS bits that provide eight classes, matching the number delivered by IP precedence. In many real-world networks, a packet might 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) can be mapped into each of the eight individual Ethernet CoS values. Figure 14-2 shows 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.

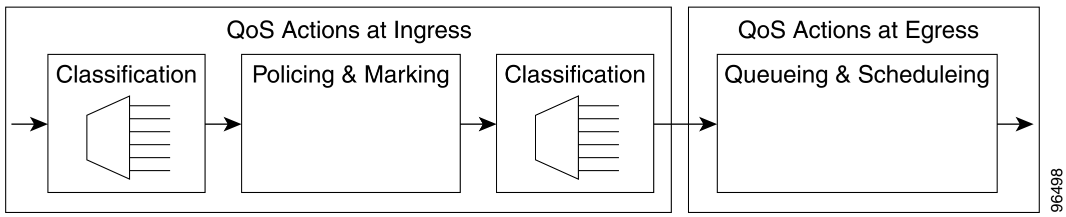
Figure 14-2 Ethernet Frame and the CoS Bit (IEEE 802.1p)



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 Cisco proprietary resilient packet ring RPR-CoS. After they are classified into class flows, further QoS functions can be applied to each packet as it traverses the card. Figure 14-3 illustrates the ML-Series QoS flow.

Figure 14-3 ML-Series QoS Flow



Policing provided by the ML-Series card ensures that attached equipment does not submit more than a predefined 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 that the peak bandwidth is over-subscribed with an appropriate ratio. The policing action is applied per classification.

Priority marking can set the Ethernet IEEE 802.1p CoS bits or RPR-CoS bits as they exit the ML-Series card. The marking feature operates on the outer IEEE 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 overcommitted. 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

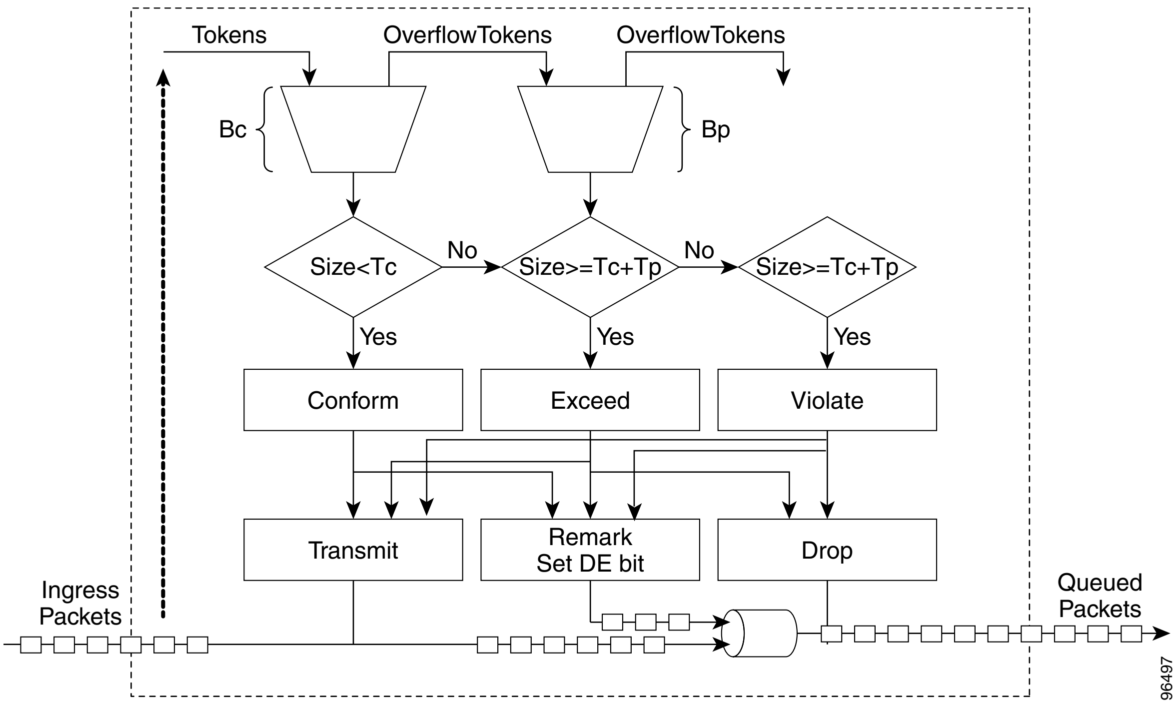
Classification can be based on any single packet classification criteria or a combination (logical AND and OR). A total of 254 classes, not including the class default, can be defined on the card. Classification of packets is configured using the Modular CLI **class-map** command. For traffic transiting the Cisco Proprietary 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 14-4 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 nonconforming 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 nonconform 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 14-4 Dual Leaky Bucket Policer Model



Marking and Discarding with a Policer

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 2l, 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.

In some cases, it might be desirable to discard all traffic of a specific ingress class. This can be accomplished by using a police command of the following form with the class: **police 96000 conform-action drop exceed-action drop**.

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 is received, it is re-marked. If a marked packet is transported over the Cisco proprietary 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 the case of a packet without any Q-tag, the marking does not have any significance.

The local scheduler treats all nonconforming packets as discard eligible regardless of their CoS setting or the global CoS commit definition. For Cisco proprietary RPR implementation, the discard eligible (DE) packets are marked using the DE bit on the Cisco proprietary 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 Cisco proprietary 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 of memory for the buffer pool. Ethernet ports share 6 MB of the memory, and packet-over-SONET/SDH (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 percent to 50 percent of the shared buffer capacity. Dynamic buffer allocation to each queue can be reduced based on the number of queues that need 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 by committing 100 percent of the 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 is 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 according to 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 WDRR as well as by priority scheduling mechanisms from the queued traffic associated with each egress port.

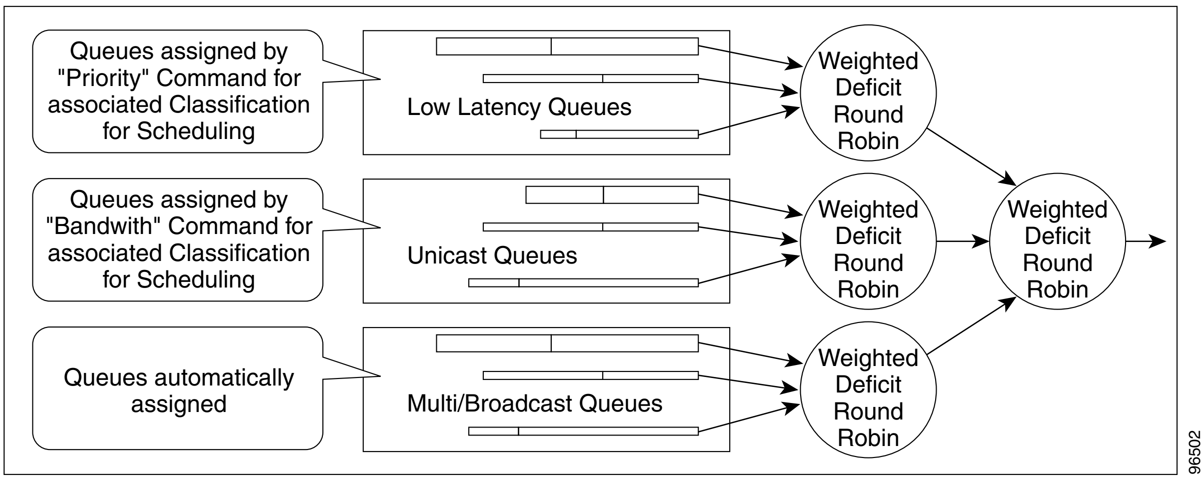
Though ordinary round robin servicing of queues can be done in constant time, unfairness occurs when different queues use different packet sizes. Deficit Round Robin (DRR) scheduling solves this problem. If a queue was not able to send a packet in its previous round because its packet size was too large, 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 14-5 illustrates the ML-Series card's queuing and scheduling.

Figure 14-5 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 Gigabit Ethernet port, approximately 293 kbps for traffic exiting an OC-12c port, and approximately 49 kbps for traffic exiting a FastEthernet port.

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 percent 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 DE allows some packets to be treated as committed and some as discard-eligible on the scheduler. For Ethernet frames, the CoS (IEEE 802.1p) bits are used to identify committed and discard eligible packets, where the RPR-CoS and the DE bits are used for Cisco proprietary RPR traffic. When congestion occurs and a queue begins to fill, the DE 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.

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 than other packets in the multicast/broadcast queue. The Ethernet CoS (IEEE 802.1p) for Layer 2-tunneled protocols can be assigned by the ML-Series card.

Egress Priority Marking

Egress priority marking allows the operator to assign the IEEE 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 IEEE 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 IEEE 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 IEEE 802.1Q tag, then the **set-cos** command has no effect on that packet. If an IEEE 802.1Q tag is inserted in the packet (either a normal tag or a QinQ tag), the inserted tag has the set-cos priority. If an IEEE 802.1Q tag is present on packet ingress and retained on packet egress, the priority of that tag is modified. If the ingress interface is a QinQ access port and the **set-cos** policy-map classifies based on ingress tag priority, this classifies based on the user priority. This is a way to allow the user-tag priority to determine the SP 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 IEEE 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.

Ingress Priority Marking

Ingress priority marking can be done for all input packets of a port, for all input packets matching a classification, or based on a measured rate. Marking of all packets of an input class can also be done with a policing command of the form **police 96000 conform-action set-cos-transmit exceed-action set-cos-transmit**. Using this command with a policy map that contains only the "class-default" will mark all ingress packets to the value. Rate based priority marking is discussed in the "Marking and Discarding with a Policer" section on page 14-5.

QinQ Implementation

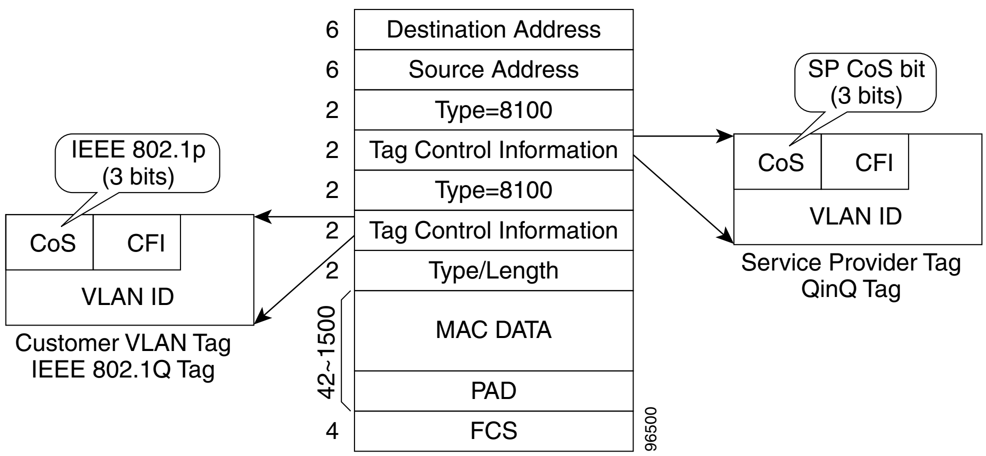
The hierarchical VLAN or IEEE 802.1Q tunneling feature enables the service provider to transparently carry the customer VLANs coming from any specific port (UNI) and transport them over the service provider network. This feature is also known as QinQ, which is performed by adding an additional IEEE 802.1Q 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 (SP) tag is added, the QinQ network typically loses any visibility to the IP header or the customer Ethernet IEEE 802.1Q tag on the QinQ encapsulated frames.

On the ML-Series cards, the QinQ access ports (IEEE 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 the 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 IEEE 802.1p bit of the SP tag. The ML-Series cards do not have visibility into the customer CoS, IP precedence, or DSCP values after the packet is double-tagged (because it is beyond the entry point of the QinQ service).

Figure 14-6 illustrates the QinQ implementation on the ML-Series card.

Figure 14-6 QinQ



The ML-Series cards can be used as the IEEE 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 frame.

Flow Control Pause and QoS

When flow control and policy-map are enabled for an interface and the policy-map is configured only with 'class-default' having policer action, flow control handles the bandwidth. For all the packets, which match policers drop criteria, PAUSE frames are sent upstream so that far end device can reduce it's transmit rate accordingly. If the far end device honours the received PAUSE frames, there will not be any drops on ML card due to policer configuration. However, if the policer gets noncompliant flow, the packets are dropped or demarked using the policer definition of the interface.

The above statement is valid for an interface, which has a policer with not only a class-default (i.e with non-default class) configured. When the policy-map is configured only with class-default, the policer acts instead of allowing the flow control to drop or demark the frames.

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 Cisco Proprietary RPR

For VLAN bridging over Cisco proprietary RPR , all ML-Series cards on the ring must be configured with the base Cisco proprietary RPR and Cisco proprietary RPR QoS configuration. SLA and bridging configurations are only needed at customer Cisco proprietary RPR access points, where IEEE 802.1Q VLAN CoS is copied to the Cisco proprietary RPR CoS. This IEEE 802.1Q VLAN CoS copying can be overwritten with a **set-cos** *action* command. The CoS commit rule applies at Cisco proprietary RPR ring ingress.

- If the packet does not have a VLAN header, the Cisco proprietary RPR CoS for non-VLAN traffic is set using the following rules:
- The default CoS is 0.
- If the packet comes in with an assigned CoS, the assigned CoS replaces the default. If an IP packet originates locally, the IP precedence setting replaces the CoS setting.
- The input policy map has a **set-cos** action.
- The output policy map has a **set-cos** action (except for broadcast or multicast packets).

The Cisco proprietary RPR header contains a CoS value and DE indicator. The Cisco proprietary RPR DE is set for noncommitted traffic.

The ML-Series card Cisco proprietary RPR transit traffic, which is defined as traffic going from POS port to POS port around the Cisco proprietary RPR , can only be classified by Layer 2 CoS. Other match rules are ignored. This is a ML-Series card specific implementation of QoS on Cisco proprietary RPR designed for the CoS based QoS model of the Cisco Metro Ethernet Solution.

This Layer 2 CoS dependence prevents DSCP-based output policy maps from working properly with Cisco proprietary RPR on the ML-Series card. Using a DSCP based policy-map causes all transit traffic to be incorrectly treated as class-default. This results in a discard of the transit traffic without any regard for the DSCP priority when transit station congestion occurs.

The DSCP based output policy map limitation has a work around. Each Cisco proprietary RPR frame has it's own three bit CoS marking, which is normally copied from the VLAN CoS. This is the field on which "match cos" classification is done for transit Cisco proprietary RPR traffic. The Cisco proprietary RPR CoS can be marked based on the DSCP match at the input station, and then classified based on the Cisco proprietary RPR CoS at transit stations. This method can support a maximum of eight classes. If you are using nine classes (including class-default), two of them would need to be combined to use this work-around.

Example 14-1 shows a class and policy-map definition configuration that would overcome the DSCP limitation. The example also changes nine classes into eight by combining the Voice and Call-Sig classes.

Caution "Match cos 0" should not be included in the definition of any class-map, because non-VLAN-tagged Ethernet packets are always treated as CoS 0 on input from Ethernet. Using “match cos 0” might incorrectly match all traffic coming from Ethernet.

Example 14-1 Class and Policy-map Definition Configuration Overcoming the DSCP Limitation

```
class-map match-any Bulk-Data
  match ip dscp af11
  match cos 3
class-map match-any Crit-Data
  match ip dscp af21  af31
  match cos 7
class-map match-any Net-Management
  match ip dscp cs2
  match cos 2
class-map match-any Video
  match ip dscp cs4  af41
  match cos 4
class-map match-any Voice
  description  Includes Voice and Call Signalling
  match ip dscp ef
  match ip dscp cs3
  match cos 5
class-map match-any Routing
  match ip dscp cs6
  match cos 6
class-map match-any Scavenger
  match ip dscp cs1
  match cos 1
policy-map MAN-QoS-DSCP
  class Voice
     priority percent 4
     set cos 5
  class Bulk-Data
    bandwidth percent 20
    set cos 3
  class Crit-Data
    bandwidth percent 20
    set cos 7
  class Net-Management
    bandwidth percent 2
    set cos 2
  class Video
    bandwidth percent 5
    set cos 4
  class Routing
    bandwidth percent 2
    set cos 6
  class Scavenger
    bandwidth percent 1
    set cos 1
  class class-default
    bandwidth percent 45
    set cos 0
```

Configuring QoS

This section describes the tasks for configuring the ML-Series card QoS functions using the 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 14-12

For QoS configuration examples, see the "QoS Configuration Examples" section on page 14-18.

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-map-name*
no class-map [**match-any** | **match-all**] *class-map-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 the **match-any** keyword is specified, the traffic class behaves in a manner consistent with the **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 **match** commands in Table 14-1, as needed.

Table 14-1 Traffic Class Commands

Command	Purpose
`Router(config)# class-map class-map-name`	Specifies the user-defined name of the traffic class. Names can be a maximum of 40 alphanumeric characters. If neither **match-all** nor **match-any** is 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 class-map-name`	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 class-map-name`	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 bridge-group-number`	Specifies the bridge-group-number against whose contents packets are checked to determine if they belong to the class.
`Router(config-cmap)# match cos cos-number`	Specifies the CoS value against whose contents packets are checked to determine if they belong to the class.

Table 14-1 Traffic Class Commands (continued)

Command	Purpose
`Router(config-cmap)# match input-interface interface-name`	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 used in Cisco proprietary RPR (SPR1) is a valid interface-name for the ML-Series card. For more information on the SPR interface, see Chapter 17, "Configuring Cisco Proprietary 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 ip-dscp-value`	Specifies up to eight DSCP values used as match criteria. The value of each service code point is from 0 to 63.
`Router (config-cmap)# match ip precedence ip-precedence-value`	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 that 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 shown in Example 14-2 and Example 14-3.

Example 14-2 Policy-map syntax

```
policy-map policy-name
no policy-map policy-name
```

Example 14-3 Class command syntax

```
class class-map-name
no class class-map-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 commands in Table 14-2 as needed.

Table 14-2 Traffic Policy Commands

Command	Purpose
`Router (config)# policy-map policy-name`	Specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
`Router (config-pmap)# class class-map-name`	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 {bandwidth-kbps | percent percent}`	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. Valid choices for the ML-Series cards are: • Rate in kilobits per second • Percent of total available bandwidth (1 to 100) 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). **Note** When using the **bandwidth** command, excess traffic (beyond the configured commit) is 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 has 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. **Note** The true configurable bandwidth in kilobits or megabits per second is per port and depends on how the ML-Series card is configured. The **show interface** command shows the maximum bandwidth of a port (for example, BW 100000 Kbit). The sum of all bandwidth and priority actions applied to the interface, plus the cos priority-mcast bandwidth, is not allowed to exceed the maximum bandwidth of the port.

Table 14-2 Traffic Policy Commands (continued)

Command	Purpose
`Router (config-pmap-c)#` **`police`** *`cir-rate-bps normal-burst-byte`* `[`*`max-burst-byte`*`] [`***`pir`***` `*`pir-rate-bps`*`] [`**`conform-action`**` {`**`set-cos-transmit`**` | `**`transmit`**` | `**`drop`**`}] [`**`exceed-action`**` {`**`set-cos-transmit`**` | `**`drop`**`}] [`**`violate-action`**` {`**`set-cos-transmit`**` | `**`drop`**`}]`	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. • For *cir-rate-bps,* specify the average committed information rate (cir) in bits per second (bps). The range is 96000 to 800000000. • For *normal-burst-byte,* specify the cir burst size in bytes. The range is 8000 to 64000. • (Optional) For *maximum-burst-byte*, specify the peak information rate (pir) burst in bytes. The range is 8000 to 64000. • (Optional) For *pir-rate-bps*, specify the average pir traffic rate in bps where the range is 96000 to 800000000. • (Optional) Conform action options are: – Set a CoS priority value and transmit – Transmit packet (default) – Drop packet • (Optional) Exceed action options are: – Set a CoS value and transmit – Drop packet (default) • (Optional) The violate action is only valid if pir is configured. Violate action options are: – Set a CoS value and transmit – Drop packet (default)

Table 14-2 Traffic Policy Commands (continued)

Command	Purpose
`Router (config-pmap-c)# priority kbps`	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. **Note** This **priority** command does not apply to the default class. **Note** When using the priority action, the traffic in that class is given a 100 percent CIR, regardless of the rate entered as the priority rate. To ensure that other bandwidth 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. **Note** The true configurable bandwidth in kilobits or megabits per second is per port and depends on how the ML-Series card is configured. The **show interface** command shows the maximum bandwidth of a port (for example, BW 100000 Kbit). The sum of all bandwidth and priority actions applied to the interface, plus the cos priority-mcast bandwidth, is not allowed to exceed the maximum bandwidth of the port.
`Router (config-pmap-c)# set cos cos-value`	Specifies a CoS value or values to associate with the packet. The number is in the range from 0 to 7. This command can 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 therefore overrides the CoS set by a policer action. 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 takes precedence in setting the IEEE 802.1p CoS field. This command also sets the CoS value in the Cisco proprietary RPR header for packets exiting the ML-Series on the Cisco proprietary RPR interface.

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 global configuration mode, as needed:

Step 1	`Router(config)# interface interface-id`	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 Bridge Group Virtual Interfaces (BVIs).
Step 2	`Router(config-if)# service-policy output policy-map-name`	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.
Step 3	`Router(config-if)# service-policy input policy-map-name`	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

The global **cos commit** *cos-value* 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, as shown in Table 14-3.

Table 14-3 CoS Commit Command

Command	Purpose
`Router(config)# cos-commit cos-value`	Labels packets that come in with a CoS equal to or higher than the *cos-value* as CIR and packets with a lower CoS as DE.

Monitoring and Verifying QoS Configuration

After configuring QoS on the ML-Series card, the configuration of class maps and policy maps can be viewed through a variety of **show** commands. To display the information relating to a traffic class or traffic policy, use one of the commands in Table 14-4 in EXEC mode, as needed. Table 14-4 describes the commands that are related to QoS status.

Table 14-4 Commands for QoS Status

Command	Purpose
`Router# show class-map name`	Displays the traffic class information of the user-specified traffic class.
`Router# show policy-map`	Displays all configured traffic policies.
`Router# show policy-map name`	Displays the user-specified policy map.
`Router# show policy-map interface interface`	Displays configurations of all input and output policies attached to an interface. Statistics displayed with this command are unsupported and show zero.

Example 14-4 show examples of the QoS commands.

Example 14-4 QoS Status Command Examples

```
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
```

QoS Configuration Examples

This section provides the specific command and network configuration examples:

Traffic Classes Defined Example

Example 14-5 shows how to create a class map called class1 that matches incoming traffic entering interface fastethernet0.

Example 14-5 Class Interface Command Examples

```
Router(config)# class-map class1
Router(config-cmap)# match input-interface fastethernet0
```

Example 14-6 shows how to create a class map called class2 that matches incoming traffic with IP-precedence values of 5, 6, and 7.

Example 14-6 Class IP-Precedence Command Examples

```
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 that 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 class-map, an error message is printed and the class is ignored. The supported commands that allow multiple values are **match cos**, **match ip precedence**, and **match ip dscp**.

Example 14-7 shows how to create a class map called class3 that matches incoming traffic based on bridge group 1.

Example 14-7 Class Map Bridge Group Command Examples

```
Router(config)# class-map class3
Router(config-cmap)# match bridge-group 1
```

Traffic Policy Created Example

In Example 14-8, 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 14-12.

Example 14-8 Traffic Policy Created Example

```
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.

Example 14-9 shows a traffic class configured with the **class-map match-all** command.

Example 14-9 Class Map Match All Command Examples

```
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.

In traffic class called cisco2, the match criteria are evaluated consecutively until a successful match criterion is located. The packet is first evaluated to the 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. When 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.

Example 14-10 shows a traffic class configured with the **class-map match-any** command.

Example 14-10 Class Map Match Any Command Examples

```
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
```

match spr1 Interface Example

In Example 14-11, the SPR interface is specified as a parameter to the **match input-interface** CLI when defining a class-map.

Example 14-11 Class Map SPR Interface Command Examples

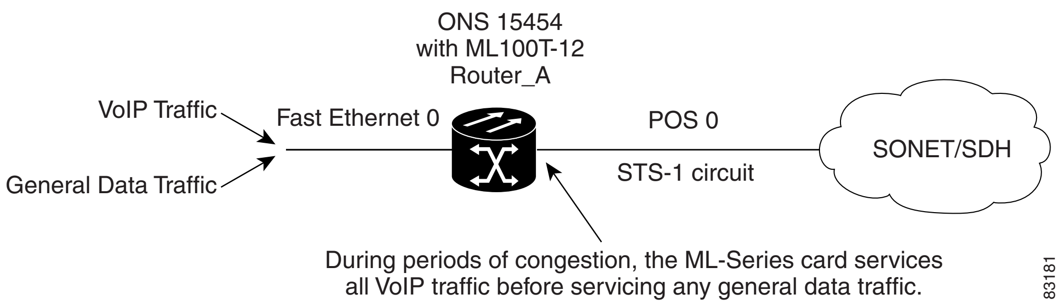
```
Router(config)# class-map spr1-cos1
Router(config-cmap)# match input-interface spr1
Router(config-cmap)# match cos 1
Router(config-cmap)# end
Router# sh class-map spr1-cos1
 Class Map match-all spr1-cos1 (id 3)
```

```
Match input-interface SPR1
Match cos  1
```

ML-Series VoIP Example

Figure 14-7 shows an example of ML-Series QoS configured for VoIP. The associated commands are provided in Example 14-12.

Figure 14-7 ML-Series VoIP Example



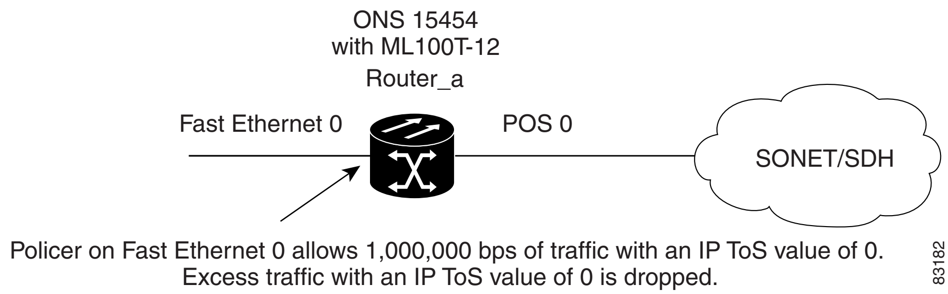
Example 14-12 ML-Series VoIP Commands

```
Router(config)# class-map match-all voip
Router(config-cmap)# match ip precedence 5
Router(config-cmap)# exit
Router(config)# class-map match-any default
Router(config-cmap)# match any
Router(config-cmap)# exit
Router(config)# policy-map pos0
Router(config-pmap)# class default
Router(config-pmap-c)# bandwidth   1000
Router(config-pmap-c)# class voip
Router(config-pmap-c)# priority 1000
Router(config-pmap-c)# interface FastEthernet0
Router(config-if)# ip address 1.1.1.1 255.255.255.0
Router(config-if)# interface POS0
Router(config-if)# ip address 2.1.1.1 255.255.255.0
Router(config-if)# service-policy output pos0
Router(config-if)# crc 32
Router(config-if)# no cdp enable
Router(config-if)# pos flag c2 1
```

ML-Series Policing Example

Figure 14-8 shows an example of ML-Series policing. The example shows how to configure a policer that restricts traffic with an IP precedence of 0 to 1,000,000 bps. The associated code is provided in Example 14-13.

Figure 14-8 ML-Series Policing Example



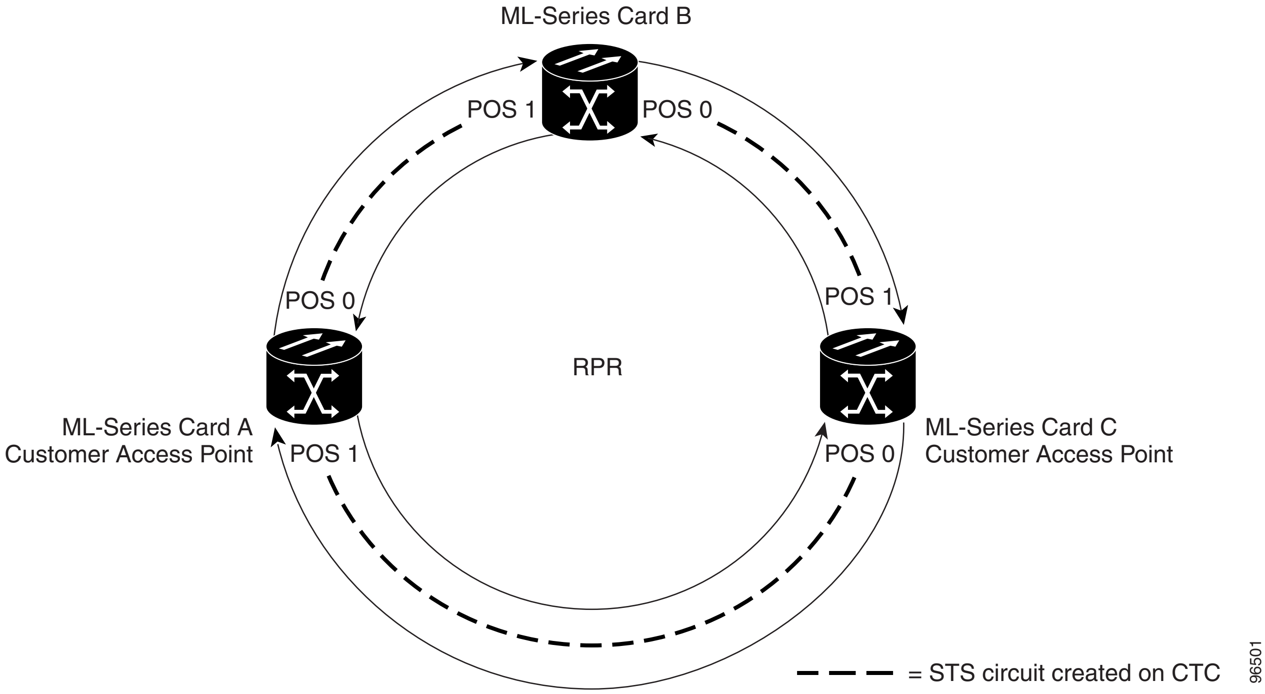
Example 14-13 ML-Series Policing Commands

```
Router(config)# class-map match-all policer
Router(config-cmap)# match ip precedence 0
Router(config-cmap)# exit
Router(config)# policy-map police_f0
Router(config-pmap)# class policer
Router(config-pmap-c)# police 1000000 10000 conform-action transmit exceed-action drop
Router(config-pmap-c)# interface FastEthernet0
Router(config-if)# service-policy input police_f0
```

ML-Series CoS-Based QoS Example

Figure 14-9 shows an example of ML-Series CoS-based QoS. The associated code is provided in the examples that follow the figure. The CoS example assumes that the ML-Series cards are configured into an Cisco proprietary RPR and that the ML-Series card POS ports are linked by point-to-point SONET circuits. ML-Series Card A and ML-Series Card C are customer access points. ML-Series Card B is not a customer access point. For more information on configuring Cisco proprietary RPR , see Chapter 17, "Configuring Cisco Proprietary Resilient Packet Ring."

Figure 14-9 ML-Series CoS Example



Example 14-14 shows the code used to configure ML-Series card A in Figure 14-9.

Example 14-14 ML-Series Card A Configuration (Customer Access Point)

```
ML_Series_A(config)# cos commit 2
ML_Series_A(config)# policy-map Fast5_in
ML_Series_A(config-pmap)# class class-default
ML_Series_A(config-pmap-c)# police 5000 8000 8000 pir 10000 conform-action
set-cos-transmit 2 exceed-action set-cos-transmit 1 violate-action drop
```

Example 14-15 shows the code used to configure ML-Series card B in Figure 14-9.

Example 14-15 ML-Series Card B Configuration (Not Customer Access Point)

```
ML_Series_B(config)# cos commit 2
```

Example 14-16 shows the code used to configure ML-Series card C in Figure 14-9.

Example 14-16 ML-Series Card C Configuration (Customer Access Point)

```
ML_Series_B(config)# cos commit 2
ML_Series_B(config)# policy-map Fast5_in
ML_Series_B(config-pmap)# class class-default
ML_Series_B(config-pmap-c)# police 5000 8000 8000 pir 10000 conform-action
set-cos-transmit 2 exceed-action set-cos-transmit 1 violate-action drop
```

Understanding Multicast QoS and Priority Multicast Queuing

ML-Series card QoS supports the creation of two priority classes for multicast traffic in addition to the default multiclass traffic class. Creating a multicast priority queuing class of traffic configures the ML-Series card to recognize an existing CoS value in ingressing multicast traffic for priority treatment.

The multicast priority queuing CoS match is based on the “internal" CoS value of each packet. This value is normally the same as the egress CoS value (after policer marking if enabled) but differs in two cases. The internal CoS value is not the same as the egress value when dot1q-tunneling is used. Under dot1q-tunnel, the internal CoS value is always the value of the outer tag CoS, both when entering the dot1q tunnel and leaving the dot1q tunnel. The internal CoS value is also not the same as the egress value if a packet is transported over a VLAN, and the VLAN tag is removed on egress to send the packet untagged. In this case, the internal CoS is the CoS of the removed tag (including ingress policing and marking if enabled).

The **cos priority-mcast** command does not modify the CoS of the multicast packets, but only the bandwidth allocation for the multicast priority queuing class. The command guarantees a minimum amount of bandwidth and is queued separately from the default multicast/broadcast queue.

Creating a multicast priority queuing class allows for special handling of certain types of multiclass traffic. This is especially valuable for multicast video distribution and service provider multicast traffic. For example, a service provider might want to guarantee the protection of their own multicast management traffic. To do this, they could create a multicast priority queuing class on the ML-Series card for the CoS value of the multicast management traffic and guarantee its minimum bandwidth. For multicast video distribution, a multicast priority queuing class on the ML-Series card for the CoS value of the multicast video traffic enables networks to efficiently manage multicast video bandwidth demands on a network shared with VoIP and other Ethernet services.

Note Multicast priority queuing traffic uses port-based load-balancing over Cisco proprietary RPR and EtherChannel. Default multicast traffic is load-balanced over Cisco proprietary RPR , but not over EtherChannel. Multicast load balancing maps GigabitEthernet Port 0 to POS Port 0 and GigabitEthernet Port 1 to POS Port 1. Multicast load balancing maps Fast Ethernet Port 0 and all even-numbered Fast Ethernet ports to POS 0 and all odd-numbered Fast Ethernet ports to POS 1.

Note Multicast priority queuing bandwidth should not be oversubscribed for sustained periods with traffic from multiple sources. This can result in reduced multicast priority queuing throughput.

Priority multicast feature is not required and is not supported in ML card while it is in IEEE-RPR mode, as in this mode each queue created for a port can handle all of multicast,broadcast and unicast traffic.

Default Multicast QoS

Default multicast traffic is any multicast traffic (including flooded traffic) that is not classified as multicast priority queuing. The default multicast class also includes broadcast data traffic, control traffic, Layer 2 protocol tunneling, and flooding traffic of the unknown MAC during MAC learning.

With no QoS configured (no multicast priority queuing and no output policy map) on the ML-Series card, the default multicast bandwidth is a 10 percent minimum of total bandwidth.

When bandwidth is allocated to multicast priority queuing but no output policy map is applied, the default multicast congestion bandwidth is a minimum of 10 percent of the bandwidth not allocated to multicast priority queuing.

When an output policy-map is applied to an interface, default multicast and default unicast share the minimum bandwidth assigned to the default class. This default class is also known as the match-any class. The minimum bandwidth of default multicast is 10 percent of the total default class bandwidth.

Multicast Priority Queuing QoS Restrictions

The following restrictions apply to multicast priority queuing QoS:

- The bandwidth allocation and utilization configured for multicast priority queuing traffic is global and applies to all the ports on the ML-Series card, both POS and Fast Ethernet or Gigabit Ethernet, regardless of whether these ports carry multicast priority queuing traffic. The rate of traffic can be reduced for all ports on the ML-Series card when this feature is configured. Default multicast traffic uses bandwidth only on the ports where it egresses, not globally like multicast priority queuing.
- Multicast priority queuing QoS is supported only for Layer 2 bridging.
- The ML-Series card supports a maximum of two multicast priority queuing classes.
- Unlike the rest of the ML-Series card QoS, multicast priority queuing QoS is not part of the Cisco IOS MQC.
- Priority-mcast bandwidth allocation is per port and the maximum bandwidth configurable on an ML1000-2 with **cos priority-mcast** is 1000 Mbps. But the load-balancing of multicast priority queuing increases the effective bandwidth. For example, with an ML1000-2 with Gigabit EtherChannel (GEC) and STS-24c circuits, the user can allocate 1000 Mbps per port, but will be able to get 2000 Mbps total effective bandwidth due to the load-balancing.

Configuring Multicast Priority Queuing QoS

To configure a priority class for multicast traffic, use the global configuration **cos priority-mcast** command, defined in Table 14-5.

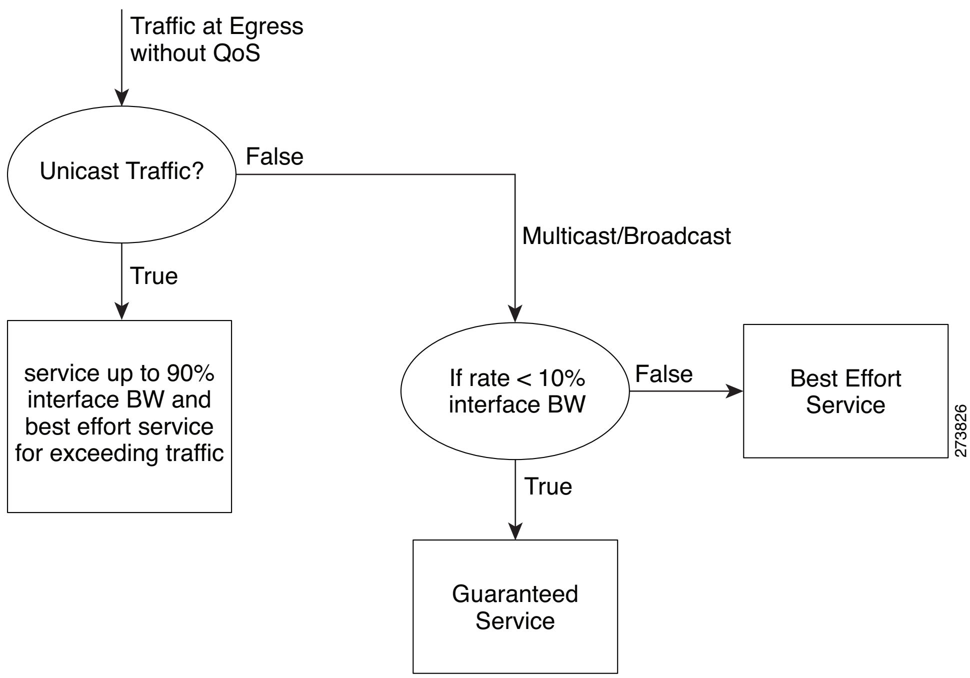
Table 14-5 CoS Multicast Priority Queuing Command

Command	Purpose
`Router (config)# [`**`no`**`] `**`cos priority-mcast`**` `*`cos-value`*` {`*`bandwidth-kbps`*` | `**`mbps`**` `*`bandwidth-mbps`*` | `**`percent`**` `*`percent`*`}`	Creates a priority class of multicast traffic based on a multicast CoS value and specifies a minimum bandwidth guarantee to a traffic class in periods of congestion. *cos-value* specifies the CoS value of multicast packets that will be given the bandwidth allocation. The value matches only a single CoS of traffic (not a range). The supported CoS range is 0 to 7. A minimum bandwidth guarantee can be specified in kbps, in Mbps, or by a percentage of the overall available bandwidth. Valid choices for the ML-Series card are: • Rate in kilobits per second • Rate in megabits per second • Percent of total available port bandwidth (1 to 100) Reentering the command with the same *cos-value* but a different bandwidth rate will modify the bandwidth of the existing class. Reentering the command with a different *cos-value* creates a separate multicast priority queuing class with a maximum of two multicast priority queuing classes. The **no** form of this command removes the multicast priority queuing class. **Note** The true configurable bandwidth in kilobits or megabits per second is per port and depends on how the ML-Series card is configured. The **show interface** command shows the maximum bandwidth of a port (for example, BW 100000 Kbit). The sum of all bandwidth and priority actions applied to the interface, plus the cos priority-mcast bandwidth, is not allowed to exceed the maximum bandwidth of the port. **Note** Attempting to configure a priority-mcast bandwidth that exceeds the true configurable bandwidth on any port will cause the priority-mcast configuration change to fail, and the multicast priority queuing bandwidth guarantee will not be changed.

QoS not Configured on Egress

The QoS bandwidth allocation of multicast and broadcast traffic is handled separately from unicast traffic. On each interface, the aggregate multicast and broadcast traffic are given a fixed bandwidth commit of 10% of the interface bandwidth. This is the optimum bandwidth that can be provided for traffic exceeding 10% of the interface bandwidth.

Figure 14-10 QoS not Configured on Egress



ML-Series Egress Bandwidth Example

This section explains with examples the utilization of bandwidth across different queues with or without Priority Multicast.

Case 1: QoS with Priority and Bandwidth Configured Without Priority Multicast

Strict Priority Queue is always serviced first. The remaining interface bandwidth is utilized to service other configured traffic.

In the following example, after servicing unicast `customer_voice` traffic, the remaining interface bandwidth is utilized for other WRR queues such as `customer_core_traffic`, `customer_data`, and `class-default` in the ratio of 1:3:5.

At any given time, the sum of the bandwidth assigned cannot exceed the interface bandwidth (in kbps). The bandwidth share allocated to `class-default` will be utilized by default unicast traffic (in this example, unicast traffic with CoS values other than 2, 5, 7) and all multicast/broadcast traffic (all CoS values). The default unicast and all multicast/broadcast traffic will be serviced in the ratio of 9:1.

For example, if 18x bandwidth is available after servicing priority unicast traffic (CoS 5), then the remaining bandwidth will be allocated as follows:

Unicast traffic with CoS 2 : 2x

Unicast traffic with CoS 7: 6x

Unicast default (without CoS 2, CoS 5, CoS 7): 9x

All multicast/broadcast (any CoS value): 1x

Example 14-17 QoS with Priority and Bandwidth Configured without Priority Multicast

```
!
class-map match-all customer_voice
  match cos  5
class-map match-all customer_data
  match cos  7
class-map match-all customer_core_traffic
  match cos  2
!
!
policy-map policy_egress_bandwidth
  class customer_core_traffic
    bandwidth   1000
  class customer_voice
    priority 1000
  class customer_data
    bandwidth   3000
  class class-default
    bandwidth   5000
!
!
interface POS0
 no ip address
  crc 32
 service-policy output policy_egress_bandwidth
!
```

Case 2: QoS with Priority and Bandwidth Configured with Priority Multicast

In this case, only multicast traffic of CoS 3 is allocated a guaranteed bandwidth. This multicast traffic will now participate in the queue along with other WRR queues. After servicing the `customer_voice` traffic, the remaining interface bandwidth is utilized for WRR queues, such as `customer_core_traffic`, `customer_data`, `class-default`, and multicast CoS 3 traffic in the ratio of 1:3:5:2.

At any given time, the sum of the bandwidth assigned cannot exceed the interface bandwidth (in kbps).

Example 14-18 QoS with Priority and Bandwidth configured with Priority Multicast

```
cos priority-mcast 3 2000
!
class-map match-all customer_voice
  match cos  5
class-map match-all customer_data
  match cos  7
class-map match-all customer_core_traffic
  match cos  2
!
!
```

```
policy-map policy_egress_bandwidth
  class customer_core_traffic
    bandwidth   1000
  class customer_voice
    priority 1000
  class customer_data
    bandwidth   3000
  class class-default
    bandwidth   5000
!
!
interface POS0
 no ip address
  crc 32
 service-policy output policy_egress_bandwidth
!
```

Understanding CoS-Based Packet Statistics

Enhanced performance monitoring displays per-CoS packet statistics on the ML-Series card interfaces when CoS accounting is enabled. Per-CoS packet statistics are only supported for bridged services, not for IP routing or Multiprotocol Label Switching (MPLS). CoS-based traffic utilization is displayed at the Fast Ethernet or Gigabit Ethernet interface or subinterface (VLAN) level, or at the POS interface level. It is not displayed at the POS subinterface level. Cisco proprietary RPR statistics are not available at the SPR interface level, but statistics are available for the individual POS ports that make up the SPR interface. EtherChannel (port-channel) and BVI statistics are available only at the member port level. Table 14-6 shows the types of statistics available at specific interfaces.

Table 14-6 Packet Statistics on ML-Series Card Interfaces

Statistics Collected	Gigabit/Fast Ethernet Interface	Gigabit/Fast Ethernet Subinterface (VLAN)	POS Interface	POS Subinterface
Input—Packets and Bytes	Yes	Yes	No	No
Output—Packets and Bytes	Yes	Yes	No	No
Drop Count—Packets and Bytes[1]	Yes	No	Yes	No

1. Drop counts only include discards caused by output congestion and are counted at the output interface.

CoS-based packet statistics are available through the Cisco IOS CLI and Simple Network Management Protocol (SNMP), using an extension of the CISCO-PORT-QOS MIB. They are not available through CTC.

Configuring CoS-Based Packet Statistics

Note CoS-based packet statistics require the enhanced microcode image to be loaded onto the ML-Series card.

Note For IEEE 802.1Q (QinQ) enabled interfaces, CoS accounting is based only on the CoS value of the outer metro tag imposed by the service provider. The CoS value inside the packet sent by the customer network is not considered for CoS accounting.

For information on the enhanced microcode image, see the "Multiple Microcode Images" section on page 3-11.

To enable CoS-based packet statistics on an interface, use the interface configuration level command defined in Table 14-7.

Table 14-7 CoS-Based Packet Statistics Command

Command	Purpose
`Router(config-if)# cos accounting`	Enables CoS-based packet statistics to be recorded at the specific interface and for all the subinterfaces of that interface. This command is supported only in interface configuration mode and not in subinterface configuration mode. The **no** form of the command disables the statistics.

After configuring CoS-based packet statistics on the ML-Series card, the statistics can be viewed through a variety of **show** commands. To display this information, use one of the commands in Table 14-8 in EXEC mode.

Table 14-8 Commands for CoS-Based Packet Statistics

Command	Purpose
Router# **show interface** *type number cos*	Displays the CoS-based packet statistics available for an interface.
Router# **show interface** *type number.subinterface-number* **cos**	Displays the CoS-based packet statistics available for a FastEthernet or Gigabit Ethernet subinterface. POS subinterfaces are not eligible.

Example 14-19 shows examples of these commands.

Example 14-19 Commands for CoS-Based Packet Statistics Examples

```
Router# show interface gigabitethernet 0.5 cos
GigabitEthernet0.5
    Stats by Internal-Cos
    Input: Packets        Bytes
        Cos 0: 31         2000
        Cos 1:
        Cos 2: 5          400
        Cos 3:
        Cos 4:
        Cos 5:
        Cos 6:
        Cos 7:
    Output: Packets       Bytes
        Cos 0: 1234567890  1234567890
        Cos 1: 31         2000
        Cos 2:
```

```
        Cos 3:
        Cos 4:
        Cos 5:
        Cos 6: 10          640
        Cos 7:

Router# show interface gigabitethernet 0 cos
GigabitEthernet0
    Stats by Internal-Cos
    Input: Packets         Bytes
        Cos 0: 123         3564
        Cos 1:
        Cos 2: 3           211
        Cos 3:
        Cos 4:
        Cos 5:
        Cos 6:
        Cos 7:
    Output: Packets        Bytes
        Cos 0: 1234567890  1234567890
        Cos 1: 3           200
        Cos 2:
        Cos 3:
        Cos 4:
        Cos 5:
        Cos 6: 1           64
        Cos 7:
    Output: Drop-pkts      Drop-bytes
        Cos 0: 1234567890  1234567890
        Cos 1:
        Cos 2:
        Cos 3:
        Cos 4:
        Cos 5: 1               64
        Cos 6: 10              640
        Cos 7:

Router# show interface pos0 cos
    POS0
    Stats by Internal-Cos
    Output: Drop-pkts          Drop-bytes
        Cos 0: 12              1234
        Cos 1: 31              2000
        Cos 2:
        Cos 3:
        Cos 4:
        Cos 5:
        Cos 6: 10              640
        Cos 7:
```

Understanding IP SLA

Cisco IP SLA, formerly known as the Cisco Service Assurance Agent, is a Cisco IOS feature to assure IP service levels. Using IP SLA, service provider customers can measure and provide service level agreements, and enterprise customers can verify service levels, verify outsourced service level agreements, and understand network performance for new or existing IP services and applications. IP SLAs use unique service level assurance metrics and methodology to provide highly accurate, precise service level assurance measurements.

Depending on the specific IP SLAs operation, statistics of delay, packet loss, jitter, packet sequence, connectivity, path, server response time, and download time are monitored within the Cisco device and stored in both CLI and SNMP MIBs. The packets have configurable IP and application layer options such as source and destination IP address, User Datagram Protocol (UDP)/TCP port numbers, a ToS byte (including DSCP and IP Prefix bits), Virtual Private Network (VPN) routing/forwarding instance (VRF), and URL web address.

IP SLAs use generated traffic to measure network performance between two networking devices such as routers. IP SLAs starts when the IP SLAs device sends a generated packet to the destination device. After the destination device receives the packet, and depending on the type of IP SLAs operation, the device will respond with time-stamp information for the source to make the calculation on performance metrics. An IP SLAs operation is a network measurement to a destination in the network from the source device using a specific protocol such as UDP for the operation.

Because IP SLA is accessible using SNMP, it also can be used in performance monitoring applications for network management systems (NMSs) such as CiscoWorks2000 (CiscoWorks Blue) and the Internetwork Performance Monitor (IPM). IP SLA notifications also can be enabled through Systems Network Architecture (SNA) network management vector transport (NMVT) for applications such as NetView.

For general IP SLA information, refer to the Cisco IOS IP Service Level Agreements technology page at http://www.cisco.com/warp/public/732/Tech/nmp/ipsla. For information on configuring the Cisco IP SLA feature, see the "Network Monitoring Using Cisco Service Assurance Agent" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2*.

IP SLA on the ML-Series

The ML-Series card has a complete IP SLA Cisco IOS subsystem and offers all the normal features and functions available in Cisco IOS Release 12.2S. It uses the standard IP SLA Cisco IOS CLI commands. The SNMP support will be equivalent to the support provided in the IP SLA subsystem 12.2(S), which is the rttMon MIB.

IP SLA Restrictions on the ML-Series

The ML-Series card supports only features in the Cisco IOS 12.2S branch. It does not support functions available in later Cisco IOS versions, such as the IP SLA accuracy feature or the enhanced Cisco IOS CLI support with updated IP SLA nomenclature.

Other restrictions are:

- Setting the CoS bits is supported, but set CoS bits are not honored when leaving or entering the CPU when the sender or responder is an ONS 15454, ONS 15454 SDH, or ONS 15310-CL platform. Set CoS bits are honored in intermediate ONS nodes.
- On Cisco proprietary RPR , the direction of the data flow for the IP SLA packet might differ from the direction of customer traffic.
- The system clock on the ML-Series card synchronizes with the clock on the TCC2/TCC2P card. Any NTP server synchronization is done with the TCC2/TCC2P card clock and not with the ML-Series card clock.
- The average Round Trip Time (RTT) measured on an ML-Series IP SLA feature is more than the actual data path latency. In the ML-Series cards, IP SLA is implemented in the software. The IP SLA messages are processed in the CPU of the ML-Series card. The latency time measured includes

the network latency and CPU processing time. For very accurate IP SLA measurements, it is recommended that a Cisco Router or Switch be used as an external probe or responder to measure the RTT of the ML-Series cards in a network.