Cisco Systems

Traffic Engineering with MPLS

Design, configure, and manage MPLS TE to optimize network performance

Eric Osborne, CCIE® #4122
Ajay Simha, CCIE #2970

ciscopress.com

Traffic Engineering with MPLS

Eric Osborne, CCIE No. 4122
Ajay Simha, CCIE No. 2970

Cisco Press

Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Traffic Engineering with MPLS

Eric Osborne
Ajay Simha

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America 2 3 4 5 6 7 8 9 0

Second Printing September 2004

Library of Congress Cataloging-in-Publication Number: 200-1086632

ISBN: 1-58705-031-5

Warning and Disclaimer

This book is designed to provide information about Multiprotocol Label Switching Traffic Engineering (MPLS TE). Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press and Cisco Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members of the professional technical community.

Reader feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through e-mail at feedback@ciscopress.com. Please be sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Corporate and Government Sales

Cisco Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales.

For more information please contact: U.S. Corporate and Government Sales 1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside the U.S. please contact: International Sales international@pearsoned.com

You can find additional information about this book, any errata, and Appendix B at www.ciscopress.com/1587050315.

Publisher	John Wait
Editor-In-Chief	John Kane
Cisco Systems Management	Michael Hakkert Tom Geitner
Executive Editor	Brett Bartow
Production Manager	Patrick Kanouse
Development Editor	Christopher Cleveland
Project Editor	Eric T. Schroeder
Copy Editor	Gayle Johnson
Technical Editors	Jim Guichard Alexander Marhold Jean-Philippe Vasseur
Team Coordinator	Tammi Ross
Book Designer	Gina Rexrode
Cover Designer	Louisa Klucznik
Compositor	Amy Parker
Indexer	Tim Wright

CISCO SYSTEMS

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

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
Capital Tower
168 Robinson Road
#22-01 to #29-01
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the **Cisco.com Web site at www.cisco.com/go/offices.**

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Czech Republic Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland • Israel • Italy Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Printed in the USA

About the Authors

Eric Osborne, CCIE No. 4122, has been doing Internet engineering of one sort or another since 1995. He's seen fire, he's seen rain, he's seen sunny days that he thought would never end. He joined Cisco in 1998 to work in the Cisco TAC, moved to the ISP Expert Team shortly after Ajay, and has been involved in MPLS since the Cisco IOS Software Release 11.1CT days. His BS degree is in psychology, which, surprisingly, is often more useful than you might think. He is a frequent speaker at the Cisco Networkers events in North America, having delivered the "Deploying MPLS Traffic Engineering" talk since 2000. He can be reached at eosborne@cisco.com.

Ajay Simha, CCIE No. 2970, graduated with a BS in computer engineering in India, followed by a MS in computer science. He joined the Cisco Technical Assistance Center in 1996 after working as a data communication software developer for six years. He then went on to support Tier 1 and 2 ISPs as part of the Cisco ISP Expert Team. His first exposure to MPLS TE was in early 1999. It generated enough interest for him to work with MPLS full-time. Simha has been working as an MPLS deployment engineer since October 1999. He has first-hand experience in troubleshooting, designing, and deploying MPLS. He can be reached at asimha@cisco.com.

About the Technical Reviewers

Jim Guichard, CCIE No. 2069, is an MPLS deployment engineer at Cisco Systems. In recent years at Cisco, he has been involved in the design, implementation, and planning of many large-scale WAN and LAN networks. His breadth of industry knowledge, hands-on experience, and understanding of complex internetworking architectures have enabled him to provide a detailed insight into the new world of MPLS and its deployment. He can be reached at jguichar@cisco.com.

Alexander Marhold, CCIE No. 3324, holds an MSC degree in industrial electronics and an MBA. He works as a senior consultant and leader of the Core IP Services team at PROIN, a leading European training and consulting company focused on service provider networks. His focus areas are core technologies such as MPLS, high-level routing, BGP, network design, and implementation. In addition to his role as a consultant, Marhold is also a CCSI who develops and holds specialized training courses in his area of specialization. His previous working experience also includes teaching at a polytechnic university for telecommunications, as well as working as CIM project manager in the chemical industry.

Jean-Philippe Vasseur has a French engineer degree and a master of science from the SIT (New Jersey, USA). He has ten years of experience in telecommunications and network technologies and worked for several service providers prior to joining Cisco. After two years with the EMEA technical consulting group, focusing on IP/MPLS routing, VPN, Traffic Engineering, and GMPLS designs for the service providers, he joined the Cisco engineering team. The author is also participating in several IETF drafts.

Dedications

Ajay Simha: I want to dedicate this book to my dear wife, Anitha, and loving children, Varsha and Nikhil, who had to put up with longer working hours than usual. This book is also dedicated to my parents, who provided the educational foundation and values in life that helped me attain this level.

Eric Osborne: I want to dedicate this book to the many coffee shops within walking distance of my house; without them, this book may never have had enough momentum to get finished. I would also like to dedicate this book to my mother (who taught me to make lists), my father (who taught me that addition is, indeed, cumulative), and to anyone who ever taught me anything about networking, writing, or thinking. There's a bit of all of you in here.

Acknowledgments

There are so many people to thank. To begin with, we'd be remiss if we didn't extend heartfelt thanks to the entire MPLS development team for their continual sharing of expertise, often in the face of our daunting cluelessness. Special thanks go to Carol Itturalde, Rob Goguen, and Bob Thomas for answering our questions, no matter how detailed, how obscure, or how often we asked them.

We also want to thank our primary technical reviewers—Jim Guichard, Jean-Philippe Vasseur, and Alexander Marhold. Their guidance and feedback kept us on course.

We'd also like to thank the folks who reviewed parts of this book for accuracy and relevance. In alphabetical order: Aamer Akhter, Santiago Alvarez, Vijay Bollapragada, Anna Charny, Clarence Filsfils, Jim Gibson, Carol Iturralde, Francois Le Faucheur, Gwen Marceline, Trevor Mendez, Stefano Previdi, Robert Raszuk, Khalid Raza, Mukhtiar Shaikh, George Swallow, Dan Tappan, Mosaddaq Turabi, Siva Valliappan, Shankar Vemulapalli, and Russ White.

Eric wants to thank the members of PSU Local #1 for confirming and correcting his assumptions about how large Internet backbones really work.

Our last thanks go to the Cisco Press editing team—specifically, to Chris Cleveland and Brett Bartow for shepherding us through this process. It took over a year to do, and we couldn't have done it without them.

Contents at a Glance

Foreword xvi

Introduction xvii

Chapter 1 Understanding Traffic Engineering with MPLS 3

Chapter 2 MPLS Forwarding Basics 21

Chapter 3 Information Distribution 81

Chapter 4 Path Calculation and Setup 113

Chapter 5 Forwarding Traffic Down Tunnels 207

Chapter 6 Quality of Service with MPLS TE 253

Chapter 7 Protection and Restoration 291

Chapter 8 MPLS TE Management 343

Chapter 9 Network Design with MPLS TE 391

Chapter 10 MPLS TE Deployment Tips 441

Chapter 11 Troubleshooting MPLS TE 493

Appendix A MPLS TE Command Index 541

Appendix B CCO and Other References 553

Index 568

Contents

Foreword xvi

Introduction xvii

Chapter 1 Understanding Traffic Engineering with MPLS 3

Basic Networking Concepts 3
TDM 3
Statistical Multiplexing 4

What Is Traffic Engineering? 6

Traffic Engineering Before MPLS 7
The Fish Problem 8

Enter MPLS 10
Decoupling Routing and Forwarding 11
Better Integration of the IP and ATM Worlds 11
Building Services with MPLS 13
What MPLS TE Is Not 15

Using MPLS TE in Real Life 16

Summary 17

Chapter 2 MPLS Forwarding Basics 21

MPLS Terminology 21

Forwarding Fundamentals 24
What Is a Label? 25
Control Plane Versus Data Plane 28
Classification 28
Control Planes in an MPLS Network 30
Forwarding Mechanics 31
Label Distribution Concepts 42

Label Distribution Protocol 50
LDP PDU Header 50
LDP Message Format 51
LDP's Major Functions 53
Loop Detection 68

Label Distribution Protocol Configuration 69
Configuring CEF 70

Configuring MPLS Forwarding Globally 72
Interface-Level Configuration 72
Configuring Session Attributes 77

Summary 78

Chapter 3 Information Distribution 81

MPLS Traffic Engineering Configuration 81

What Information Is Distributed 84
Available Bandwidth Information 85
Tunnel Priority 88
Attribute Flags 93
Administrative Weight 96

When Information Is Distributed 98
Flood Significant Changes Immediately 99
Flood Insignificant Changes Periodically, But More Often Than the IGP Refresh Interval 102
If a Change That Has Not Yet Been Flooded Is Known to Cause an Error, Flood Immediately 103

How Information Is Distributed 103
MPLS Traffic Engineering in OSPF 103
MPLS Traffic Engineering Flooding in IS-IS 106

Summary 110

Chapter 4 Path Calculation and Setup 113

How SPF Works 114

How CSPF Works 119
Tiebreakers in CSPF 122
Other Things That Influence CSPF 124
CSPF Knobs 128

Tunnel Reoptimization 135
Periodic Reoptimization 136
Manual Reoptimization 136
Event-Driven Reoptimization 136
Lockdown 137

Resource Reservation Protocol (RSVP) 138
RSVP Basics 139
RSVP Packets 143
RSVP Operation 168
RSVP in the Real World 181

Interarea Tunnels 187
IGP Terminology 187
What Interarea Tunnels Can Do 188
How Interarea Tunnels Work 189
What Interarea Tunnels Cannot Do 195

Link Manager 197
show mpls traffic-eng link-management admission-control Command 198
show mpls traffic-eng link-management advertisements Command 199
show mpls traffic-eng link-management bandwidth-allocation Command 200
show mpls traffic-eng link-management igp-neighbors Command 202
show mpls traffic-eng link-management interfaces Command 202
show mpls traffic-eng link-management statistics Command 203
show mpls traffic-eng link-management summary Command 204

Summary 205

Chapter 5 Forwarding Traffic Down Tunnels 207

Forwarding Traffic Down Tunnels Using Static Routes 207

Forwarding Traffic Down Tunnels with Policy-Based Routing 208

Forwarding Traffic Down Tunnels with Autoroute 209

Load Sharing 216
Equal-Cost Load Sharing 216
Unequal-Cost Load Sharing 221
How to Play with the TE Tunnel Metric 229

Forwarding Adjacency 242

Automatic Bandwidth Adjustment 246
How Auto Bandwidth Works 246
Auto Bandwidth Configuration 248
Auto Bandwidth Operation 250

Summary 251

Chapter 6 Quality of Service with MPLS TE 253

The DiffServ Architecture 255
Classification 255
Policing 256
Marking 256
Queuing 256
Dropping 257

A Quick MQC Review 257
- Configuring a Class Map 258
- Configuring a Policy Map 260
- Configuring a Service Policy 263

DiffServ and IP Packets 264

DiffServ and MPLS Packets 267

Label Stack Treatment 267
- ip2mpls 267
- mpls2mpls 268
- mpls2ip 269
- EXP and DSCP Are Independent 270
- Per-Hop Behaviors in the ip2mpls and mpls2ip Cases 271

Tunnel Modes 272
- Uniform Mode 273
- Short-Pipe Mode 274
- Pipe Mode 275

DiffServ-Aware Traffic Engineering (DS-TE) 276
- Configuring DS-TE 278

Forwarding DS-TE Traffic Down a Tunnel 284

Summary 288

Chapter 7 Protection and Restoration 291

The Need for Fast Reroute 291

What Is Protection? 293

Types of Protection 293
- Path Protection 294
- Local Protection 295
- Link Versus Node Protection 301

Link Protection 302
- Prefailure Configuration 302
- Failure Detection 309
- Connectivity Restoration 312
- Post-Failure Signalling 317
- Link Protection Configuration Summary 324

Node Protection 326
- Similarities Between Link Protection and Node Protection 326

Differences Between Link Protection and Node Protection 327

Advanced Protection Issues 329
Multiple Backup Tunnels 329
Backup Bandwidth Reservation 333
Backup Tunnel Selection Summary 336
Promotion 337
Configuring Multiple Backup Tunnels to Multiple NNHops 338

Summary 340

Chapter 8 MPLS TE Management 343

MPLS LSR MIB 343
Interface Configuration Table (mplsInterfaceConfTable) 344
Interface Performance Table (mplsInterfacePerfTable) 349
InSegment Table (mplsInSegmentTable) 350
InSegment Performance Table (mplsInSegmentPerfTable) 354
OutSegment Table (mplsOutSegmentTable) 355
OutSegment Performance Table (mplsOutSegmentPerfTable) 358
Cross-Connect Table (mplsXCTable) 359

MPLS TE MIB 361
mplsTunnelTable 362
mplsTunnelHopTable 383
mplsTunnelResourceTable 387

Summary 389

Chapter 9 Network Design with MPLS TE 391

Sample Network for Case Studies 392

Different Types of TE Design 395

Tactical TE Design 396
When You Decide to Build TE LSPs 396
Where You Put TE LSPs 398
When to Remove Your Tactical TE Tunnels 403
Useful TE Features for Tactical TE 406

Online Strategic TE Design 407
LSP Scalability 408
Other Growth Factors 420

Offline Strategic TE Design 426
Packing Problem 427
Using an Offline Tool for LSP Placement 430

Protection Scalability 431

Link Protection 432
Node Protection 433
Path Protection 433
Actual Data for Determining Scalability 434

Forwarding Adjacency Scalability 437

Summary 437

Chapter 10 MPLS TE Deployment Tips 441

Bandwidth and Delay Measurements 441
NetFlow 444
Traffic Matrix Statistics 444
Using TE Tunnels for Traffic Measurement 453
Service Assurance Agent 455

Fine-Tuning MPLS TE Parameters 462
Headend Configuration Knobs 462
Midpoint Configuration Knobs 464

Migrating IS-IS from Narrow to Wide Metrics 465
Moving from Narrow to Wide Metrics in Two Steps 466
Moving from Narrow to Wide Metrics in Three Steps 467

TE and Multicast 467

Tunnel Identification Schemes 469

Combining MPLS TE with MPLS VPNs 471
A TE Tunnel Per VRF 476
LDP on a Tunnel Interface 476

Deployment Possibilities 479
Applications of Tactical TE 479
TE for Protection 485
TE for Unequal-Cost One-Hop Load Balancing 489

Summary 490

Chapter 11 Troubleshooting MPLS TE 493

Common Configuration Mistakes 496
Common Configuration on All MPLS TE Routers 496
Headend Configuration 502

Tools for Troubleshooting MPLS TE Problems 504

Finding the Root Cause of the Problem 506
Tunnel-Down Problems 506
Tunnel-Up Problems 532

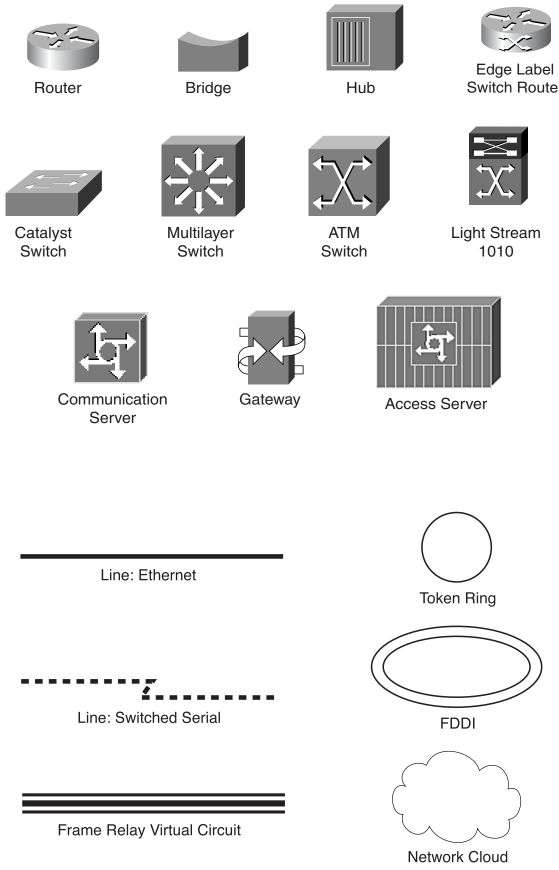
Summary 538

Appendix A MPLS TE Command Index 541

Appendix B CCO and Other References 553

Index 568

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.
- **Boldface** indicates commands and keywords that are entered literally as shown. In configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.

Foreword

Tag Switching, the Cisco proprietary technology that evolved into MPLS began in March 1996. At that time, several major ISPs were operating two-tiered networks in order to manage the traffic in their network. You see, IP always takes the shortest path to a destination. This characteristic is important to the scalability of the Internet because it permits routing to be largely an automatic process. However, the shortest path is not always the fastest path or the most lightly loaded. Furthermore, in any non-traffic-engineered network, you find a distribution of link utilizations, with a few links being very heavily loaded and many links being very lightly loaded. You end up with many network users competing for the resources of the busy links, while other links are underutilized. Neither service levels nor operational costs are optimized. In fact, one ISP claims that, with Traffic Engineering, it can offer the same level of service with only 60 percent of the links it would need without Traffic Engineering.

Thus, Traffic Engineering becomes an economic necessity, enough of a necessity to build a whole separate Layer 2 network. To engineer traffic, an ISP would create a mesh of links (virtual circuits) between major sites in its IP network and would use the Layer 2 network, either Frame Relay or ATM, to explicitly route traffic by how they routed these virtual circuits.

By April 1996, it was recognized at Cisco that tag switching offered a means of creating explicit routes within the IP cloud, eliminating the need for a two-tiered network. Because this held the potential for major cost savings to ISPs, work began in earnest shortly thereafter. Detailed requirements and technical approaches were worked out with several ISP and equipment vendors.

Eric Osborne and Ajay Simha work in the development group at Cisco that built Traffic Engineering. They have been actively involved in the deployment of Traffic Engineering in many networks. They are among those with the greatest hands-on experience with this application. This book is the product of their experience. It offers an in-depth, yet practical, explanation of the various elements that make up the Traffic Engineering application: routing, path selection, and signalling. Throughout, these explanations are related back to the actual configuration commands and examples. The result is a book of great interest to anyone curious about Traffic Engineering and an invaluable guide to anyone deploying Traffic Engineering.

George Swallow

Cisco Systems, Inc.

Architect for Traffic Engineering and Co-Chair of the IETF’s MPLS Working Group

Introduction

This book concentrates on real-world usage of MPLS TE. We spend most of our time discussing things you can configure, tools you can use to troubleshoot and manage MPLS TE, and design scenarios.

This is not an introduction to MPLS. There's enough literature out there, both Cisco and non-Cisco, that we didn't feel the need to spend time on an MPLS introduction. Although Chapter 2 reviews the basics, we generally assume that you're familiar with the three basic label operations (push, pop, and swap) and how an MPLS packet is forwarded through a network. But as soon as you're past that point, this book is for you.

You might already be using MPLS TE in your network. If so, this book is also for you. This book has many details that we hope will be useful as you continue to use and explore MPLS TE.

Or perhaps you are designing a new backbone and are considering MPLS TE for use in your network. If so, this book is also for you as well. Not only do you need to understand the protocol mechanisms to properly design a network, you also need to understand the ramifications of your design choices.

Who Should Read This Book?

Everybody! You, your friends, your grandmother, her knitting-circle friends, your kids, and their kindergarten classmates—everybody! Actually, we're not so much concerned with who *reads* this book as with who *buys* it, but to ask you to buy it and not read it is pretty crass.

In all seriousness, this book is for two kinds of people:

- **Network engineers**—Those whose job it is to configure, troubleshoot, and manage a network
- **Network architects**—Those who design networks to carry different types of traffic (voice and data) and support service-level agreements (SLAs)

We have friends who, in their respective jobs, fill both roles. To them, and to you if you do the same, we say, "Great! Buy two copies of this book!"

How This Book Is Organized

This book is designed to be read either cover-to-cover or chapter-by-chapter. It divides roughly into four parts:

- Chapters 1 and 2 discuss the history, motivation, and basic operation of MPLS and MPLS TE.
- Chapters 3, 4, and 5 cover the basic processes used to set up and build TE tunnels on your network.
- Chapters 6 and 7 cover advanced MPLS TE applications: MPLS TE and QoS, and protection using Fast Reroute (FRR).
- Chapters 8, 9, 10, and 11 cover network management, design, deployment, and troubleshooting—things you need to understand to be able to apply MPLS TE in the real world.

Here are the details on each chapter:

- **Chapter 1,** "**Understanding Traffic Engineering with MPLS**"—This chapter discusses the history of basic data networks and the motivation for MPLS and MPLS TE as the next step in the evolution of networks.
- **Chapter 2,** "**MPLS Forwarding Basics**"—This chapter is a quick review of how MPLS forwarding works. Although this book is not an introduction to MPLS, you might find it beneficial to brush up on some of the details, and that's what this chapter provides.

- **Chapter 3, "Information Distribution"**—This chapter begins the series of three chapters that are really the core of this book. The protocols and mechanisms of MPLS TE have three parts, and the first is distributing MPLS TE information in your IGP.
- **Chapter 4, "Path Calculation and Setup"**—This chapter is the second of the three core chapters. It covers what is done with information after it has been distributed by your IGP. The two prominent pieces covered in this chapter are Constrained SPF (CSPF) and Resource Reservation Protocol (RSVP).
- **Chapter 5, "Forwarding Traffic Down Tunnels"**—This chapter is the last of the three core chapters. It covers what is done with TE tunnels after they are set up. This chapter covers load sharing in various scenarios, announcing TE tunnels into your IGP as a forwarding adjacency, and automatic tunnel bandwidth adjustment using a Cisco mechanism called auto bandwidth.
- **Chapter 6, "Quality of Service with MPLS TE"**—This chapter covers the integration of MPLS and MPLS TE with the DiffServ architecture. It also covers DiffServ-Aware Traffic Engineering (DS-TE).
- **Chapter 7, "Protection and Restoration"**—This chapter covers various traffic protection and restoration mechanisms under the umbrella of Cisco's FRR—how to configure these services, how they work, and how to greatly reduce your packet loss in the event of a failure in your network.
- **Chapter 8, "MPLS TE Management"**—This chapter covers tools and mechanisms for managing an MPLS TE network.
- **Chapter 9, "Network Design with MPLS TE"**—This chapter predominantly covers scalability. It looks at different ways to deloy MPLS TE on your network, and how the various solutions scale as they grow.
- **Chapter 10, "MPLS TE Deployment Tips"**—This chapter covers various knobs, best practices, and case studies that relate to deploying MPLS TE on your network.
- **Chapter 11, "Troubleshooting MPLS TE"**—This chapter discusses tools and techniques for troubleshooting MPLS TE on an operational network.

Two appendixes are also provided. Appendix A lists all the major commands that are relevant to MPLS TE. Appendix B lists resources such as URLs and other books. Appendix B is also available at www.ciscopress.com/1587050315.

This page intentionally left blank

CHAPTER 1

Understanding Traffic Engineering with MPLS

Multiprotocol Label Switching (MPLS) has been getting a lot of attention in the past few years. It has been successfully deployed in a number of large networks, and it is being used to offer both Internet and virtual private network (VPN) services in networks around the world.

Most of the MPLS buzz has been around VPNs. Why? Because if you're a provider, it is a service you can sell to your customers.

But you can do more with MPLS than use VPNs. There's also an area of MPLS known as traffic engineering (TE). And that, if you haven't already figured it out, is what this book is all about.

Basic Networking Concepts

What is a data network? At its most abstract, a *data network* is a set of nodes connected by links. In the context of data networks, the nodes are routers, LAN switches, WAN switches, add-drop multiplexers (ADMs), and the like, connected by links from 64 Kb DS0 circuits to OC192 and 10 gigabit Ethernet.

One fundamental property of data networks is *multiplexing*. Multiplexing allows multiple connections across a network to share the same transmission facilities. Two main types of multiplexing to be concerned with are

- Time-division multiplexing (TDM)
- Statistical multiplexing (statmux)

Other kinds of multiplexing, such as frequency-division multiplexing (FDM) and wavelength-division multiplexing (WDM) are not discussed here.

TDM

Time-division multiplexing is the practice of allocating a certain amount of time on a given physical circuit to a number of connections. Because a physical circuit usually has a constant bit rate, allocating a fixed amount of time on that circuit translates directly into a bandwidth allocation.

A good example of TDM is the Synchronous Optical Network (SONET) hierarchy. An OC192 can carry four OC-48s, 16 OC-12s, 64 OC-3s, 192 DS-3s, 5376 DS-1s, 129,024 DS-0s, or various combinations. The Synchronous Digital Hierarchy (SDH) is similar.

TDM is a synchronous technology. Data entering the network is transmitted according to a master clock source so that there's never a logjam of data waiting to be transmitted.

The fundamental property of TDM networks is that they allocate a fixed amount of bandwidth for a given connection at all times. This means that if you buy a T1 from one office to another, you're guaranteed 1.544 Mbps of bandwidth at all times—no more, no less.

TDM is good, but only to a point. One of the main problems with TDM is that bandwidth allocated to a particular connection is allocated for that connection whether it is being used or not. Thirty days of T1 bandwidth is roughly 4 terabits. If you transfer less than 4 terabits over that link in 30 days, you're paying for capacity that you're not using. This makes TDM rather expensive. The trade-off is that when you want to use the T1, the bandwidth is guaranteed to be available; that's what you're paying for.

Statistical Multiplexing

The expense of TDM is one reason statistical multiplexing technologies became popular. *Statistical multiplexing* is the practice of sharing transmission bandwidth between all users of a network, with no dedicated bandwidth reserved for any connections.

Statistical multiplexing has one major advantage over TDM—it's much cheaper. With a statmux network, you can sell more capacity than your network actually has, on the theory that not all users of your network will want to transmit at their maximum bit rate at the same time.

There are several statmux technologies, but the three major ones in the last ten years or so have been

- IP
- Frame Relay
- ATM

MPLS is a fourth type of statmux technology. How it fits into the picture is explained later in this chapter.

Statmux technologies work by dividing network traffic into discrete units and dealing with each of these units separately. In IP, these units are called *packets*; in Frame Relay, they're called *frames*; in ATM, they're called *cells*. It's the same concept in each case.

Statmux networks allow carriers to oversubscribe their network, thereby making more money. They also allow customers to purchase network services that are less expensive than TDM circuits, thereby saving money. A Frame Relay T1, for example, costs far less than a

TDM T1 does. The ratio of bandwidth sold to actual bandwidth is the *oversubscription ratio*. If you have an OC-12 backbone and you sell 24 OC-3s off of it, this is a 6:1 oversubscription ratio. Sometimes, this number is expressed as a percentage—in this case, 600 percent oversubscription.

Issues That Statmux Introduces

Statmux introduces a few issues that don't exist in TDM networks. As soon as packets enter the network asynchronously, you have the potential for resource contention. If two packets enter a router at the exact same time (from two different incoming interfaces) and are destined for the same outgoing interface, that's resource contention. One of the packets has to wait for the other packet to be transmitted. The packet that's not transmitted needs to wait until the first packet has been sent out the link in question. However, the delay encountered because of simultaneous resource contention on a non-oversubscribed link generally isn't that big. If 28 T1s are sending IP traffic at line rate into a router with a T3 uplink, the last IP packet to be transmitted has to wait for 27 other IP packets to be sent.

Oversubscription greatly increases the chance of resource contention at any point in time. If five OC-3s are coming into a router and one OC-12 is going out, there is a chance of buffering because of oversubscription. If you have a sustained incoming traffic rate higher than your outgoing traffic capacity, your buffers will eventually fill up, at which point you start dropping traffic.

There's also the issue of what to do with packets that are in your buffers. Some types of traffic (such as bulk data transfer) deal well with being buffered; other traffic (voice, video) doesn't. So you need different packet treatment mechanisms to deal with the demands of different applications on your network.

Statmux technologies have to deal with three issues that TDM doesn't:

- Buffering
- Queuing
- Dropping

Dealing with these issues can get complex.

Frame Relay has the simplest methods of dealing with these issues—its concepts of committed information rate (CIR), forward and backward explicit congestion notification (FECN and BECN), and the discard eligible (DE) bit.

IP has DiffServ Code Point (DSCP) bits, which evolved from IP Precedence bits. IP also has random early discard (RED), which takes advantage of the facts that TCP is good at handling drops and that TCP is the predominant transport-layer protocol for IP. Finally, IP has explicit congestion notification (ECN) bits, which are relatively new and as of yet have seen limited use.

ATM deals with resource contention by dividing data into small, fixed-size pieces called cells. ATM also has five different service classes:

- CBR (constant bit rate)
- rt-VBR (real-time variable bit rate)
- nrt-VBR (non-real-time variable bit rate)
- ABR (available bit rate)
- UBR (unspecified bit rate)

Statmux Over Statmux

IP was one of the first statmux protocols. RFC 791 defined IP in 1981. The precursor to IP had been around for a number of years. Frame Relay wasn't commercially available until the early 1990s, and ATM became available in the mid-1990s.

One of the problems that network administrators ran into as they replaced TDM circuits with Frame Relay and ATM circuits was that running IP over FR or ATM meant that they were running one statmux protocol on top of another. This is generally suboptimal; the mechanisms available at one statmux layer for dealing with resource contention often don't translate well into another. IP's 3 Precedence bits or 6 DSCP bits give IP eight or 64 classes of service. Frame Relay has only a single bit (the DE bit) to differentiate between more- and less-important data. ATM has several different service classes, but they don't easily translate directly into IP classes. As networks moved away from running multiple Layer 3 protocols (DECnet, IPX, SNA, Apollo, AppleTalk, VINES, IP) to just IP, the fact that the Layer 2 and Layer 3 contention mechanisms don't map well became more and more important.

It then becomes desirable to have one of two things. Either you avoid congestion in your Layer 2 statmux network, or you find a way to map your Layer 3 contention control mechanisms to your Layer 2 contention control mechanisms. Because it's both impossible and financially unattractive to avoid contention in your Layer 2 statmux network, you need to be able to map Layer 3 contention control mechanisms to those in Layer 2. This is one of the reasons MPLS is playing an increasingly important part in today's networks—but you'll read more about that later.

What Is Traffic Engineering?

Before you can understand how to use MPLS to do traffic engineering, you need to understand what traffic engineering is.

When dealing with network growth and expansion, there are two kinds of engineering—network engineering and traffic engineering.

Network engineering is manipulating your network to suit your traffic. You make the best predictions you can about how traffic will flow across your network, and you then order the appropriate circuits and networking devices (routers, switches, and so on). Network engineering is typically done over a fairly long scale (weeks/months/years) because the lead time to install new circuits or equipment can be lengthy.

Traffic engineering is manipulating your traffic to fit your network. No matter how hard you try, your network traffic will never match your predictions 100 percent. Sometimes (as was the case in the mid- to late-1990s), the traffic growth rate exceeds all predictions, and you can't upgrade your network fast enough. Sometimes, a flash event (a sporting event, a political scandal, an immensely popular web site) pulls traffic in ways you couldn't have planned for. Sometimes, there's an unusually painful outage—one of your three cross-country OC-192s fails, leaving traffic to find its way from Los Angeles to New York via the other two OC-192s, and congesting one of them while leaving the other one generally unused.

Generally, although rapid traffic growth, flash events, and network outages can cause major demands for bandwidth in one place, at the same time you often have links in your network that are underutilized. Traffic engineering, at its core, is the art of moving traffic around so that traffic from a congested link is moved onto the unused capacity on another link.

Traffic engineering is by no means an MPLS-specific thing; it's a general practice. Traffic engineering can be implemented by something as simple as tweaking IP metrics on interfaces, or something as complex as running an ATM PVC full-mesh and reoptimizing PVC paths based on traffic demands across it. Traffic engineering with MPLS is an attempt to take the best of connection-oriented traffic engineering techniques (such as ATM PVC placement) and merge them with IP routing. The theory here is that doing traffic engineering with MPLS can be as effective as with ATM, but without a lot of the drawbacks of IP over ATM.

This book is about traffic engineering with MPLS; amazingly enough, that's also this book's title! Its main focus is the operational aspects of MPLS TE—how the various pieces of MPLS TE work and how to configure and troubleshoot them. Additionally, this book covers MPLS TE design and scalability, as well as deployment tips for how to effectively roll out and use MPLS TE on your network.

Traffic Engineering Before MPLS

How was traffic engineering done before MPLS? Let's look at two different statmux technologies that people use to perform traffic engineering—IP and ATM.

IP traffic engineering is popular, but also pretty coarse. The major way to control the path that IP takes across your network is to change the cost on a particular link. There is no reasonable way to control the path that traffic takes based on where the traffic is coming *from*—only where it's going *to*. Still, IP traffic engineering is valid, and many large networks use it successfully. However, as you will soon see, there are some problems IP traffic engineering cannot solve.

ATM, in contrast, lets you place PVCs across the network from a traffic source to a destination. This means that you have more fine-grained control over the traffic flow on your network. Some of the largest ISPs in the world have used ATM to steer traffic around their networks. They do this by building a full mesh of ATM PVCs between a set of routers and periodically resizing and repositioning those ATM PVCs based on observed traffic from the routers. However, one problem with doing things this way is that a full mesh of routers leads to O(N^2) flooding when a link goes down and O(N^3) flooding when a router goes down. This does not scale well and has caused major issues in a few large networks.

O(N^2)?

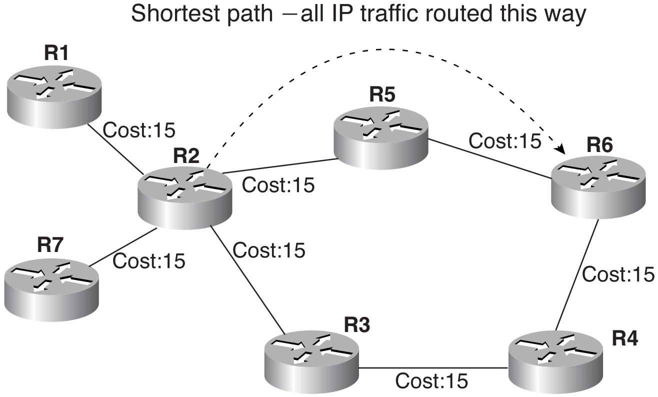
The expression O(N^2) is a way of expressing the scalability of a particular mechanism. In this case, as the number of nodes N increases, the impact on the network when a link goes down increases roughly as the square of the number of nodes—O(N^2). When a router goes down, the impact on the network increases O(N^3) as N increases.

Where do O(N^2) and O(N^3) come from? O(N^2) when a link goes down in a full-mesh environment is because the two nodes on either end of that link tell all their neighbors about the downed link, and each of those neighbors tells most of their neighbors. O(N^3) when a node goes down is because all the neighbors of that node tell all other nodes to which they are connected that a node just went away, and nodes receiving this information flood it to their neighbors. This is a well-known issue in full-mesh architectures.

The Fish Problem

Let's make things more concrete by looking at a classic example of traffic engineering (see Figure 1-1).

Figure 1-1 *The Fish Problem*



In this figure, there are two paths to get from R2 to R6:

R2→R5→R6
R2→R3→R4→R6

Because all the links have the same cost (15), with normal destination-based forwarding, all packets coming from R1 or R7 that are destined for R6 are forwarded out the same interface by R2—toward R5, because the cost of the top path is lower than that of the bottom.

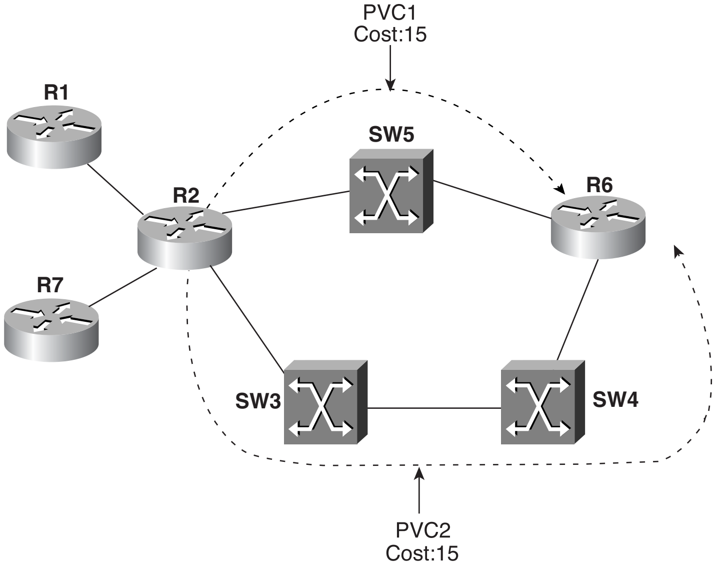
This can lead to problems, however. Assume that all links in this picture are OC-3—roughly 150 Mbps of bandwidth, after accounting for SONET overhead. And further assume that you know ahead of time that R1 sends, on average, 90 Mbps to R6 and that R7 sends 100 Mbps to R6. So what happens here? R2 tries to put 190 Mbps through a 150 Mbps pipe. This means that R2 ends up dropping 40 Mbps because it can't fit in the pipe. On average, this amounts to 21 Mbps from R7 and 19 Mbps from R1 (because R7 is sending more traffic than R1).

So how do you fix this? With destination-based forwarding, it's difficult. If you make the longer path (R2→R3→R4→R6) cost less than the shorter path, all traffic goes down the shorter path. You haven't fixed the problem at all; you just moved it.

Sure, in this figure, you could change link costs so that the short path and the long path both have the same cost, which would alleviate the problem. But this solution works only for small networks, such as the one in the figure. What if, instead of three edge routers (R1, R6, R7), you had 500? Imagine trying to set your link costs so that all paths were used! If it's not impossible, it is at least extremely difficult. So you end up with wasted bandwidth; in Figure 1-1, the longer path never gets used at all.

What about with ATM? If R3, R4, and R5 were ATM switches, the network would look like Figure 1-2.

Figure 1-2 *The Fish Problem in ATM Networks*



With an ATM network, the problem is trivial to solve. Just build two PVCs from R2 to R6, and set their costs to be the same. This fixes the problem because R2 now has two paths to R6 and is likely to use both paths when carrying a reasonably varied amount of data. The exact load-sharing mechanism can vary, but in general, CEF's per-source-destination load balancing uses both paths in a roughly equal manner.

Building two equal-cost paths across the network is a more flexible solution than changing the link costs in the ATM network, because no other devices connected to the network are affected by any metric change. This is the essence of what makes ATM's traffic engineering capabilities more powerful than IP's.

The problem with ATM TE for an IP network has already been mentioned—$O(N^2)$ flooding when a link goes down and $O(N^3)$ flooding when a router goes down.

So how do you get the traffic engineering capabilities of ATM with the routing simplicity of IP? As you might suspect, the answer is MPLS TE.

Enter MPLS

During mid-to-late 1996, networking magazine articles talked about a new paradigm in the IP world—*IP switching*. From the initial reading of these articles, it seemed like the need for IP routing had been eliminated and we could simply *switch* IP packets. The company that made these waves was Ipsilon. Other companies, such as Toshiba, had taken to ATM as a means of switching IP in their Cell-Switched Router (CSR). Cisco Systems came up with its own answer to this concept—*tag switching*. Attempts to standardize these technologies through the IETF have resulted in combining several technologies into Multiprotocol Label Switching (MPLS). Hence, it is not surprising that Cisco's tag switching implementation had a close resemblance to today's MPLS forwarding.

Although the initial motivation for creating such schemes was for improved packet forwarding speed and a better price-to-port ratio, MPLS forwarding offers little or no improvement in these areas. High-speed packet forwarding algorithms are now implemented in hardware using ASICs. A 20-bit label lookup is not significantly faster than a 32-bit IP lookup. Given that improved packet-forwarding rates are really not the key motivator for MPLS, why indulge in the added complexity of using MPLS to carry IP and make your network operators go through the pain of learning yet another technology?

The real motivation for you to consider deploying MPLS in your network is the applications it enables. These applications are either difficult to implement or operationally almost impossible with traditional IP networks. MPLS VPNs and traffic engineering are two such applications. This book is about the latter. Here are the main benefits of MPLS, as discussed in the following sections:

- Decoupling routing and forwarding
- Better integration of the IP and ATM worlds

- Basis for building next-generation network applications and services, such as provider-provided VPNs (MPLS VPN) and traffic engineering

Decoupling Routing and Forwarding

IP routing is a hop-by-hop forwarding paradigm. When an IP packet arrives at a router, the router looks at the destination address in the IP header, does a route lookup, and forwards the packet to the next hop. If no route exists, the packet is then dropped. This process is repeated at each hop until the packet reaches its destination. In an MPLS network, nodes also forward the packet hop by hop, but this forwarding is based on a fixed-length label. Chapter 2, "MPLS Forwarding Basics," covers the details of what a label is and how it is prepended to a packet. It is this capability to decouple the forwarding of packets from IP headers that enables MPLS applications such as traffic engineering.

The concept of being able to break from Layer 3-based (IP destination-based) forwarding is certainly not new. You can decouple forwarding and addressing in an IP network using concepts such as *policy-based routing* (PBR). Cisco IOS Software has had PBR support since Cisco IOS Software Release 11.0 (circa 1995). Some of the problems with using PBR to build end-to-end network services are as follows:

- The complexity in configuration management.
- PBR does not offer dynamic rerouting. If the forwarding path changes for whatever reason, you have to manually reconfigure the nodes along the new path to reflect the policy.
- The possibility of routing loops.

The limitations of PBR apply when PBR is used in an IP network to influence hop-by-hop routing behavior. PBR is easier to use in an MPLS TE-based network because PBR is used only at the tunnel headend. Using PBR in combination with MPLS does not overcome all PBR's limitations; see Chapter 5, "Forwarding Traffic Down Tunnels," for more information.

The advent of MPLS forwarding and MPLS TE enables successful decoupling of the *forwarding* process from the *routing* process by basing packet forwarding on labels rather than on an IP address.

Better Integration of the IP and ATM Worlds

From the get-go, the IP and ATM worlds seemed to clash. While ATM was being standardized, it envisioned IP coexisting with it, but always as a sideshow. Ever since the industry realized that we are not going to have our PCs and wristwatches running an ATM stack and that IP was here to stay, attempts have been made to map IP onto ATM. However, the main drawback of previous attempts to create a mapping between IP and ATM was that they either tried to keep the two worlds separate (carrying IP over ATM VCs) or tried to integrate IP and ATM with mapping services (such as ATM Address Resolution Protocol

[ARP] and Next-Hop Resolution Protocol [NHRP]). Carrying IP over ATM VCs (often called the *overlay model*) is useful, but it has scalability limits; using mapping servers introduces more points of failure into the network.

The problem with the overlay approach is that it leads to suboptimal routing unless a full mesh of VCs is used. However, a full mesh of VCs can create many routing adjacencies, leading to routing scalability issues. Moreover, independent QoS models need to be set up for IP and for ATM, and they are difficult to match.

MPLS bridges the gap between IP and ATM. ATM switches dynamically assign virtual path identifier/virtual channel identifier (VPI/VCI) values that are used as labels for cells. This solution resolves the overlay-scaling problem without the need for centralized ATM-IP resolution servers. This is called Label-Controlled ATM (LC-ATM). Sometimes it is called IP+ATM.

For further details on ATM's role in MPLS networks, read the section "ATM in Frame Mode and Cell Mode" in Chapter 2.

Traffic Engineering with MPLS (MPLS TE)

MPLS TE combines ATM's traffic engineering capabilities with IP's flexibility and class-of-service differentiation. MPLS TE allows you to build Label-Switched Paths (LSPs) across your network that you then forward traffic down.

Like ATM VCs, MPLS TE LSPs (also called TE tunnels) let the headend of a TE tunnel control the path its traffic takes to a particular destination. This method is more flexible than forwarding traffic based on destination address only.

Unlike ATM VCs, the nature of MPLS TE avoids the $O(N^2)$ and $O(N^3)$ flooding problems that ATM and other overlay models present. Rather than form adjacencies over the TE LSPs themselves, MPLS TE uses a mechanism called *autoroute* (not to be confused with the WAN switching circuit-routing protocol of the same name) to build a routing table using MPLS TE LSPs without forming a full mesh of routing neighbors. Chapter 5 covers autoroute in greater detail.

Like ATM, MPLS TE reserves bandwidth on the network when it builds LSPs. Reserving bandwidth for an LSP introduces the concept of a *consumable resource* into your network. If you build TE-LSPs that reserve bandwidth, as LSPs are added to the network, they can find paths across the network that have bandwidth available to be reserved.

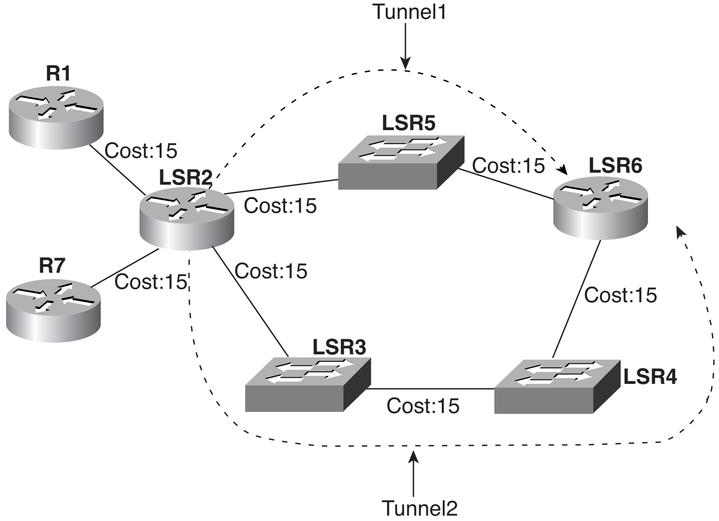
Unlike ATM, there is no forwarding-plane enforcement of a reservation. A reservation is made in the control plane only, which means that if a Label Switch Router (LSR) makes a reservation for 10 Mb and sends 100 Mb down that LSP, the network attempts to deliver that 100 Mb unless you attempt to police the traffic at the source using QoS techniques.

This concept is covered in much more depth in Chapters 3, 4, 5, and 6.

Solving the Fish Problem with MPLS TE

Figure 1-3 revisits the fish problem presented in Figure 1-1.

Figure 1-3 *The Fish Problem with LSRs*



Like ATM PVCs, MPLS TE LSPs can be placed along an arbitrary path on the network. In Figure 1-3, the devices in the fish are now LSRs.

The three major differences between ATM and MPLS TE are

- MPLS TE forwards packets; ATM uses cells. It is possible to combine both MPLS TE and MPLS/ATM integration, but currently, this is not implemented and therefore is not covered here.
- ATM requires a full mesh of routing adjacencies; MPLS TE does not.
- In ATM, the core network topology is not visible to the routers on the edge of the network; in MPLS, IP routing protocols advertise the topology over which MPLS TE is based.

All these differences are covered throughout this book; Chapter 2, specifically, talks about the nuts and bolts of MPLS forwarding.

Building Services with MPLS

In addition to its penchant for traffic engineering, MPLS can also build services across your network. The three basic applications of MPLS as a service are

- MPLS VPNs
- MPLS quality of service (QoS)
- Any Transport over MPLS (AToM)

All these applications and services are built on top of MPLS forwarding. MPLS as a service is orthogonal to MPLS for traffic engineering: They can be used together or separately.

MPLS VPNs

VPNs are nothing new to internetworking. Since the mid-to-late 1990s, service providers have offered private leased lines, Frame Relay, and ATM PVCs as a means of interconnecting remote offices of corporations. IPSec and other encryption methods have been used to create intranets over public or shared IP networks (such as those belonging to an Internet service provider [ISP]). Recently, MPLS VPNs have emerged as a standards-based technology that addresses the various requirements of VPNs, such as private IP; the capability to support overlapping address space; and intranets, extranets (with optimal routing), and Internet connectivity, while doing so in a scalable manner. A detailed explanation of MPLS VPNs is outside the scope of this book. However, you are encouraged to read *MPLS and VPN Architectures* by Jim Guichard and Ivan Pepelnjak (Cisco Press) and the other references listed in Appendix B, "CCO and Other References."

MPLS QoS

In the area of QoS, the initial goal for MPLS was to simply be able to provide what IP offered—namely, Differentiated Services (DiffServ) support. When the MPLS drafts first came out, they set aside 3 bits in the MPLS header to carry class-of-service information. After a protracted spat in the IETF, these bits were officially christened the "EXP bits," or experimental bits, even though Cisco and most other MPLS implementations use these EXP bits as you would use IP Precedence. EXP bits are analogous to, and are often a copy of, the IP Precedence bits in a packet. Chapter 6, "Quality of Service with MPLS TE," covers MPLS QoS in greater detail.

Any Transport over MPLS (AToM)

AToM is an application that facilitates carrying Layer 2 traffic, such as Frame Relay (FR), Ethernet, and ATM, over an MPLS cloud. These applications include

- Providing legacy ATM and FR circuit transport
- Point-to-point bandwidth, delay, and jitter guarantees when combined with other techniques such as DS-TE and MPLS QoS
- Extending the Layer 2 broadcast domain

- Remote point of presence (POP) connectivity, especially for ISPs to connect to remote Network Access Points (NAPs)
- Support for multi-dwelling connections, such as apartment buildings, university housing, and offices within a building

Use the URLs provided in Appendix B if you want to learn more about AToM.

What MPLS TE Is Not

You just read a lot about what MPLS TE can do. It's important to understand what MPLS is *not* so that you don't take it for more than it is:

- MPLS TE is not QoS.
- MPLS TE is not ATM.
- MPLS TE is not magic.

MPLS TE Is Not QoS

"Quality of service" means different things to different people. At an architectural level, QoS is composed of two things:

- Finding a path through your network that can provide the service you offer
- Enforcing that service

Finding the path can be as simple as using your IGP metric to determine the best route to a destination. Enforcing that service can be as simple as throwing so much bandwidth at your network that there's no need to worry about any other sort of resource contention tools. This is sometimes called "quantity of service," but in the most generic sense, it is a method of providing good service quality, and therefore good quality of service.

Or you can make things complex. You can find a path through your network with an offline TE-LSP placement tool, much like ATM PVC placement. Enforcing that path can be done using DiffServ mechanisms such as policing, marking, queuing, and dropping. MPLS (specifically, MPLS TE) is only a tool you can use to help provide high-quality service.

There's a range of options in between these two choices. In general, the more time and money you spend on path layout, provisioning, and DiffServ mechanisms, the less money you need to spend on bandwidth and the associated networking equipment. Which direction you decide to go is up to you.

MPLS TE Is Not ATM

No, it's really not. MPLS TE (as a subset of all things MPLS) has some of ATM's traffic engineering properties, but MPLS TE is not ATM. MPLS as a whole is more like Frame

Relay than ATM, if for no other reason than both MPLS and Frame Relay carry entire packets with a switching header on them, and ATM divides things into cells. Although MPLS has been successfully used to replace ATM in some networks (replacing an ATM full mesh with an MPLS TE full mesh) and complement it in others (moving from IP over ATM to IP+ATM), MPLS is not a 1:1 drop-in replacement for ATM.

As mentioned earlier, it is possible to integrate MPLS TE with MPLS ATM forwarding (in Cisco parlance, the latter is called IP+ATM). This is still not the same as carrying IP over traditional ATM networks, as with IP+ATM (also called Label-Controlled ATM, or LC-ATM) and TE integration, there's still no full mesh of routing adjacencies.

MPLS TE Is Not Magic

That's right—you heard it here first. MPLS stands for Multiprotocol Label Switching, not "Magic Problem-solving Labor Substitute," as some would have you believe. As you might expect, adding a new forwarding layer between Layer 2 and IP (some call it Layer 2.5; we prefer to stay away from the entire OSI model discussion) does not come without cost. If you're going to tactically apply MPLS TE, you need to remember what tunnels you put where and why. If you take the strategic track, you have signed up for a fairly large chunk of work, managing a full mesh of TE tunnels in addition to IGP over your physical network. Network management of MPLS TE is covered in Chapter 8, "MPLS TE Management."

But MPLS TE solves problems, and solves them in ways IP can't. As we said a few pages back, MPLS TE is aware of both its own traffic demands and the resources on your network.

If you've read this far, you're probably at least interested in finding out more about what MPLS TE can do for you. To you, we say, "Enjoy!"

NOTE Or maybe you're not interested. Maybe you're genetically predisposed to have an intense dislike for MPLS and all things label-switched. That's fine. To you we say, "Know thine enemy!" and encourage you to buy at least seven copies of this book anyway. You can always burn them for heat and then go back to the bookstore and get more.

Using MPLS TE in Real Life

Three basic real-life applications for MPLS TE are

- Optimizing your network utilization
- Handling unexpected congestion
- Handling link and node failures

Optimizing your network utilization is sometimes called the *strategic* method of deploying MPLS TE. It's sometimes also called the full-mesh approach. The idea here is that you build a full mesh of MPLS TE-LSPs between a given set of routers, size those LSPs according to how much bandwidth is going between a pair of routers, and let the LSPs find the best path in your network that meets their bandwidth demands. Building this full mesh of TE-LSPs in your network allows you to avoid congestion as much as possible by spreading LSPs across your network along bandwidth-aware paths. Although a full mesh of TE-LSPs is no substitute for proper network planning, it allows you to get as much as you can out of the infrastructure you already have, which might let you delay upgrading a circuit for a period of time (weeks or months). This translates directly into money saved by not having to buy bandwidth.

Another valid way to deploy MPLS TE is to handle unexpected congestion. This is known as the *tactical* approach, or *as needed*. Rather than building a full mesh of TE-LSPs between a set of routers ahead of time, the tactical approach involves letting the IGP forward traffic as it will, and building TE-LSPs only after congestion is discovered. This allows you to keep most of your network on IGP routing only. This might be simpler than a full mesh of TE-LSPs, but it also lets you work around network congestion as it happens. If you have a major network event (a large outage, an unexpectedly popular new web site or service, or some other event that dramatically changes your traffic pattern) that congests some network links while leaving others empty, you can deploy MPLS TE tunnels as you see fit, to remove some of the traffic from the congested links and put it on uncongested paths that the IGP wouldn't have chosen.

A third major use of MPLS TE is for quick recovery from link and node failures. MPLS TE has a component called Fast Reroute (FRR) that allows you to drastically minimize packet loss when a link or node (router) fails on your network. You can deploy MPLS TE to do just FRR, and to not use MPLS TE to steer traffic along paths other than the ones your IGP would have chosen.

Chapters 9 and 10 discuss strategic and tactical MPLS TE deployments; Chapter 7 covers Fast Reroute.

Summary

This chapter was a whirlwind introduction to some of the concepts and history behind MPLS and MPLS TE. You now have a feel for where MPLS TE came from, what it's modeled after, and what sort of problems it can solve.

More importantly, you also have a grasp on what MPLS is not. MPLS has received a tremendous amount of attention since its introduction into the networking world, and it has been exalted by some and derided by others. MPLS and MPLS TE are no more and no less than tools in your networking toolbox. Like any other tool, they take time and knowledge to apply properly. Whether you use MPLS TE in your network is up to you; the purpose of this book is to show you how MPLS TE works and the kinds of things it can do.

Although this book is not an introduction to MPLS as a whole, you might need to brush up on some MPLS basics. That's what Chapter 2 is for: It reviews basic label operations and label distribution in detail to prepare you for the rest of the book. If you're familiar with basic MPLS operation (push/pop/swap and the basic idea of LDP), you might want to skip to Chapter 3, "Information Distribution," where you can start diving into the nuts and bolts of how MPLS TE works and how it can be put to work for you.

This page intentionally left blank

This chapter covers the following topics:

- MPLS Terminology
- Forwarding Fundamentals
- Label Distribution Protocol
- Label Distribution Protocol Configuration

CHAPTER 2

MPLS Forwarding Basics

Chapter 1, “Understanding Traffic Engineering with MPLS,” provided the history and motivation for MPLS. This chapter familiarizes you with the fundamental concepts of MPLS-based forwarding. It serves as a refresher if you are already familiar with MPLS and it is a good introduction if you are not. Chapters 3 through 11 deal with MPLS Traffic Engineering. You should read the MPLS drafts, RFCs, and other reference materials listed in Appendix B, “CCO and Other References,” to obtain a more complete understanding of other MPLS topics.

MPLS Terminology

Before jumping into MPLS concepts, it is a good idea to familiarize yourself with the terminology and lingo used in MPLS.

Table 2-1 defines some common MPLS-related terms you must know in order to understand the concepts in this chapter and book.

Table 2-1 *MPLS Terminology*

Term	Definition
Upstream	A router that is closer to the source of a packet, relative to another router.
Downstream	A router that is farther from the source of a packet, relative to another router. As a packet traverses a network, it is switched from an upstream router to its downstream neighbor.
Control plane	Where control information such as routing and label information is exchanged.
Data plane/forwarding plane	Where actual forwarding is performed. This can be done only after the control plane is established.
Cisco Express Forwarding (CEF)[1]	The latest switching method used in Cisco IOS. It utilizes an *mtrie*-based organization and retrieval structure. CEF is the default forwarding method in all versions of Cisco IOS Software Release 12.0 and later.

continues

Table 2-1 *MPLS Terminology (Continued)*

Term	Definition
Label	A fixed-length tag that MPLS forwarding is based on. The term *label* can be used in two contexts. One term refers to 20-bit labels. The other term refers to the label header, which is 32 bits in length. For more details on labels, see the later section "What Is a Label?".
Label binding	An association of an FEC (prefix) to a label. A label distributed by itself has no context and, therefore, is not very useful. The receiver knows to apply a certain label to an incoming data packet because of this association to an FEC.
Label imposition	The process of adding a label to a data packet in an MPLS network. This is also referred to as "pushing" a label onto a packet.
Label disposition	The process of removing a label from a data packet. This is also referred to as "popping" a label off a packet.
Label swapping	Changing the value of the label in the MPLS header during MPLS forwarding.
Label Switch Router (LSR)	Any device that switches packets based on the MPLS label.
Label Edge Router (LER)	An LSR that accepts unlabeled packets (IP packets) and imposes labels on them at the ingress side. An LER also removes labels at the edge of the network and sends unlabeled packets to the IP network on the egress side.
Forwarding Equivalence Class (FEC)	Any set of properties that map incoming packets to the same outgoing label. Generally, an FEC is equivalent to a route (all packets destined for anything inside 10.0.0.0/8 match the same FEC), but the definition of FEC can change when packets are routed using criteria other than just the destination IP address (for example, DSCP bits in the packet header).
Label-Switched Path (LSP)	The path that a labeled packet traverses through a network, from label imposition to disposition.

Table 2-1 *MPLS Terminology (Continued)*

Term	Definition
Label stack	Apart from the label exchanged between LSRs and their neighbors, for applications such as MPLS-VPN, an end-to-end label is exchanged. As a result, a label stack is used instead of a single MPLS label. An important concept to keep in mind is that the forwarding in the core is based just on the top-level label. In the context of MPLS TE, label stacking is required when a labeled packet enters an MPLS TE tunnel.
Forwarding Information Base (FIB)[1]	The table that is created by enabling CEF on the Cisco routers.
Label Information Base (LIB)	The table where the various label bindings that an LSR receives over the LDP protocol are stored. It forms the basis of populating the FIB and LFIB tables.
Tag Information Base (TIB)	The older, "tag-switching" name for the LIB.
Explicit null	The opposite of implicit null. In the control plane, the last hop sends a label value of 0 (for IPv4) to the penultimate hop. The label value is never used for lookup. Explicit null provides some advantages that implicit null doesn't. It is used in network devices that don't support implicit null, or to carry EXP bits all the way to the tunnel tail.
Implicit null	The concept of not using a label on the last hop of an LSP in the forwarding plane. Implicit null has some performance advantages. In the control plane, the last hop of the LSP advertises a label value of 3 to indicate implicit null.
Penultimate Hop Popping (PHP)	After receiving the egress router, a labeled packet pops off the label and does an IP lookup in the CEF[1] table. This means that the egress router must do two lookups for every packet exiting the network. To reduce this burden placed on the egress router, PHP allows the penultimate hop router to remove the top-level label, which allows the LER to forward the packet based on a single lookup. The router that is immediately upstream of the tail of an MPLS TE tunnel also performs PHP.

continues

Table 2-1 *MPLS Terminology (Continued)*

Term	Definition
P/PE and C/CE	P and PE routers are LSRs and LERs in the context of MPLS-VPN. The term P comes from routers being in the provider network. C routers are routers found in the customer network. CE routers are the routers on the customer edge facing the provider. PE routers are provider edge routers, which connect to the CE routers. CE routers normally run plain IP (not required to be MPLS-aware).
Label Distribution Protocol (LDP)	One of the many protocols in place to distribute the label bindings between an LSR and its neighbor. Other mechanisms include RSVP, used in MPLS TE, and MP-BGP, used in MPLS-VPN.
Tag Distribution Protocol (TDP)	The predecessor of LDP, TDP is a Cisco-proprietary protocol that acts much like LDP. You can use TDP if interoperability between Cisco and non-Cisco devices is not important.
Resource Reservation Protocol (RSVP)	This protocol was originally intended as a signaling protocol for the Integrated Services (IntServ) quality of service (QoS) model, wherein a host requests a specific QoS from the network for a particular flow. This reservation could be within an enterprise network or over the Internet. RSVP with a few extensions has been adapted by MPLS to be the signalling protocol that supports MPLS TE within the core. RSVP theory is standardized in RFC 2205 and RFC 3209. It is covered in greater detail in Chapter 4, "Path Calculation and Setup."
Constrained Routing LDP (CR-LDP)	This is an alternative approach to RSVP that acts as a signalling protocol to achieve MPLS TE. Cisco routers support RSVP rather than CR-LDP for traffic engineering LSP setup. CR-LDP is not covered in this book.

[1] The terms *CEF* and *CEF table* are used interchangeably with *FIB*. Although CEF is the name given to the forwarding mechanism, FIB is the term used to reference the table and the internal data structures.

Forwarding Fundamentals

Table 2-1 provided an introductory glance at MPLS through terminology and definitions. This section goes into more depth about how all these concepts come together.

What Is a Label?

Labels, as you can probably guess, are an integral part of Multiprotocol *Label* Switching. The label allows the decoupling of routing from forwarding, which lets you do all sorts of neat things.

But what *is* a label? Before we define what a label is, you should know that MPLS can operate in one of two modes:

- Frame mode
- Cell mode

Frame Mode

Frame mode is the term used when you forward a *packet* with a label prepended to the packet in front of the Layer 3 header (the IP header, for example).

RFC 3031, "Multiprotocol Label Switching Architecture," defines a label as "a short fixed length physically contiguous identifier which is used to identify a FEC, usually of local significance."

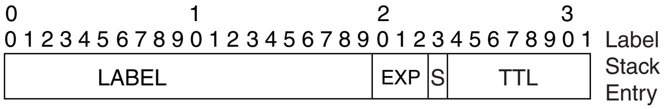
Put simply, a label is a value prepended to a packet that tells the network where the packet should go. A label is a 20-bit value, which means that there can be 2^{20} possible label values, or just over 1,000,000.

A packet can have multiple labels, carried in what's known as a *label stack*. A label stack is a set of one or more labels on a packet. At each hop in a network, only the outermost label is considered. The label that an LSR uses to forward the packet in the data plane is the label it assigned and distributed in the control plane. Hence, the inner labels have no meaning as far as the midpoints are concerned.

When labels are placed on a packet, the 20-bit label value itself is encoded with some additional pieces of information that assist in the forwarding of the labeled packet through a network.

Figure 2-1 illustrates the encoded MPLS header packet format.

Figure 2-1 *MPLS Header Packet Format*



LABEL = 20 bits
EXP = Experimental, 3 bits
S = Bottom of stack, 1 bit
TTL = Time To Live, 8 bits

This 32-bit quantity is known as a *label stack entry*, but is often referred to as just a *label*. So, when labels are discussed, the discussion could be either about the 20-bit label value or the 32-bit label stack entry. The additional 12 bits are made up of the following:

- **EXP**—EXP bits are technically reserved for experimental use. Cisco IOS Software (and pretty much every MPLS implementation) uses these EXP bits to hold a QoS indicator—often a direct copy of the IP precedence bits in an underlying IP packet. When MPLS packets are queued up, it is possible to use EXP bits in the same way that IP precedence bits are used now. You'll read more about this in Chapter 6, "Quality of Service with MPLS TE."
- **S**—The S bit is the bottom-of-stack bit. It is possible (and common) to have more than one label attached to a packet. The bottommost label in a stack has the S bit set to 1; other labels have the S bit set to 0. The S bit is there because it's sometimes useful to know where the bottom of the label stack is, and the S bit is the tool to use to find it.
- **TTL**—Time To Live bits are often (but not always) a direct copy of the IP TTL header. They are decremented at every hop to prevent routing loops from creating infinite packet storms; this is just like IP. TTL bits can also be set to something *other* than the TTL on the IP packet. This is most often used when a network operator wants to hide the underlying network topology from traceroutes from the outside world.

NOTE In some cases, such as for security concerns or to meet service-level agreements (SLAs) (although this might come across as a deception), you might need to hide the core of a service provider's network from the user community. You can do this on Cisco routers using the command **no mpls ip propagate-ttl** {**forwarded** | **local**}. This command, when used with the **forwarded** option, affects only traffic forwarded through the router. This lets TTL be used in **traceroute** commands to troubleshoot problems in the core.

Cell Mode

Cell mode is the term used when you have a network consisting of ATM LSRs that use MPLS in the control plane to exchange VPI/VCI information instead of using ATM signalling.

In cell mode, the label is said to be *encoded* in a cell's VPI/VCI fields (see Figure 2-2). After label exchange is done in the control plane, in the forwarding plane, the ingress router segments the packet into ATM cells, applying the appropriate VPI/VCI value that was exchanged in the control plane, and transmits the cells. Midpoint ATM LSRs behave like normal ATM switches—they forward a cell based on the incoming VPI/VCI and the incoming port information. Finally, the egress router reassembles the cells into a packet.

Cell mode is also called Label-Controlled ATM (LC-ATM). LC-ATM label distribution is discussed in more depth in the section "Label Distribution Concepts." The cell-mode discussion was included for the sake of completeness. It is not required for understanding MPLS traffic engineering concepts in this book because MPLS TE is not supported in cell mode on Cisco routers as of this writing.

NOTE In some of the examples containing MPLS-related output in this chapter, you'll notice that ATM VPI/VCI values show up in the *outgoing tag* column. These are cases in which a VPI/VCI was exchanged in an MPLS control plane over an ATM interface and the downstream neighbor on that interface expects to see that VPI/VCI value on the cell it receives.

ATM in Frame Mode and Cell Mode

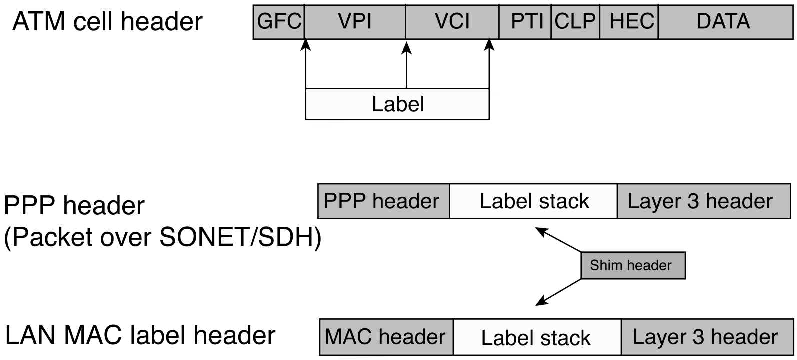
As you have seen so far, ATM switches can act as LSRs. When ATM switches are a part of the core, they can operate in two modes:

- Frame mode
- Cell mode

When a conventional ATM PVC is built to achieve classic IP over ATM (aal5snap encapsulation, for example) and MPLS is sent over that PVC, this is still called *frame-mode MPLS*. To understand this better, refer to the MPLS header format, also known as the label stack entry, illustrated in Figure 2-1.

Figure 2-2 shows the MPLS label in relation to Layer 2 and Layer 3 headers. The PPP and LAN headers show the label being inserted between the Layer 2 and Layer 3 headers (Ethernet and IP, for example). This is called a *shim* header. When operating in frame-mode MPLS, you always see a shim header. This is also applicable when you are simply connecting routers over ATM PVCs and doing MPLS in a classic IP-over-ATM environment.

Figure 2-2 *MPLS Layer 2 Encapsulation*



When running in cell mode, ATM LSRs act as routers in the control plane. In other words, they need to exchange routing information through IGP protocols, such as OSPF, and need to run a label distribution protocol, such as TDP or LDP.

NOTE You might think that ATM switches forward only ATM cells, so whenever ATM switches are involved in the MPLS core, they *must* be acting as ATM LSRs, in cell mode. This is not true. The reason this is not always true is because the ATM switch could be used to build a conventional ATM point-to-point PVC between two routers. When this is done, the routers on either end of the PVC can be directly connected LSRs. When forwarding packets to each other, they would first have to build the IP packet and insert an MPLS header in front of it and then segment the entire packet (IP packet plus MPLS header) into ATM cells. When these cells reach the router at the other end of the PVC, they are reassembled into a packet. If further forwarding is required, the forwarding is based on the label value inside the label header. In this case, even though the MPLS packets were segmented into ATM cells, there was no mapping of MPLS label to the VPI/VCI fields of the ATM cell. Thus, this would be considered frame mode.

Control Plane Versus Data Plane

The control plane is where the routing information and other control information, such as label bindings, are exchanged between LSRs. MPLS is a control plane-driven protocol, meaning that the control information exchange must be in place before the first data packet can be forwarded. The forwarding of data packets is done in the data plane.

Classification

When an IP packet arrives at a LER (the ingress router), just as in the case of normal IP forwarding, a longest-match lookup is performed by comparing the entries in the FIB against the destination IP address of the received packet. In MPLS terminology, this process is called classifying the packet. This section explains the term FEC (Forwarding Equivalence Class), as well as where classification is performed and how it differs from classification in conventional IP networks.

FEC

When IP packets destined for the same subnet arrive at an ingress router, the classification for all these packets is the same—it is based on the longest-match lookup in the FIB. For example, assume you have an entry in the FIB for 171.68.0.0/16 with a next-hop address of 12.12.12.12. If you now receive two packets with destination IP addresses 171.68.1.1 and 171.68.23.5, both these packets are forwarded to the same next hop—12.12.12.12. In most cases, it could be said that 171.68.1.1 and 171.68.23.5 share the same FEC.

However, the classification into a particular FEC need not be restricted to the destination IP address of the received packet. Classification into a FEC could be based on the interface on which the packet arrived, the IP precedence values in the packet's IP header, the packet's destination port number, or any arbitrary scheme you can imagine. Regardless of the basis of the classification, all the packets that are classified into the same FEC receive the same treatment. This treatment can be forwarding the packet down a certain path, providing the packet some preferential treatment within the core, or even dropping the packet. The current Cisco IOS Software implementation classifies IP packets based on their destination IP address, in the absence of any tools such as policy-based routing.

Translating MPLS terminology into IP terminology, the FEC is nothing but the route (also called the prefix) found in the FIB that was the best match for the incoming packet.

Living on the Edge

In conventional IP networks, the forwarding of a packet is based on the packet's destination IP address. Each node along the packet's path can forward the packet only after examining the destination IP address contained in the packet. This means that each node along the packet's path classifies the packet. This is discussed in further detail in the section "MPLS Versus IP."

In MPLS-based forwarding, after the ingress LER at the edge of the network does the classification, it pushes a label on the data packet that matches that packet's FEC. This process is called *label imposition* or *label pushing*. The LSRs in the network's core are not required to reclassify the packet. When a core router receives a labeled packet, it does three things:

- It does a label lookup on the incoming label.
- It finds the outgoing interface and outgoing label for this packet.
- It swaps the received (incoming) label for the proper outgoing label and sends the packet out the outgoing interface.

This process is known as *label swapping*.

How an LSR knows what label the downstream LSR expects is based on the label bindings that are exchanged in the control plane using a label distribution protocol (LDP, RSVP, BGP, and so on) prior to forwarding packets.

When a packet reaches the end of the network, the packet's outermost label is removed, and the remainder of the packet is forwarded to the next hop. The act of removing a label from a packet is called *label popping* or *label disposition*.

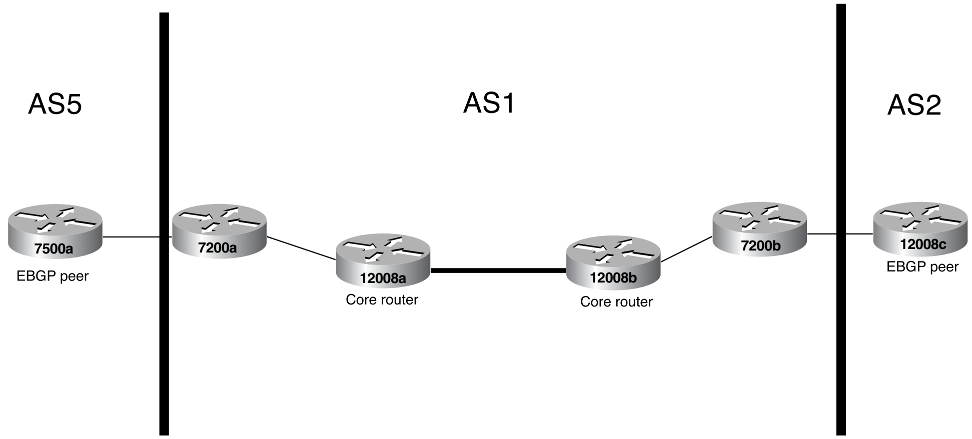
The three fundamental label operations (push/impose, swap, and pop/dispose) are all that is needed for MPLS. Label imposition/disposition and forwarding allow for an arbitrarily complex classification scheme that needs higher processing power to be enforced at the edge, while keeping the core simply forwarding MPLS packets.

Control Planes in an MPLS Network

This section looks at the processes needed to get a packet through a network—first an IP network and then an MPLS network.

For both the IP network and the MPLS network, consider the topology shown in Figure 2-3. It represents a service provider network in which gateway routers 7200a and 7200b peer with external BGP peers 7500a and 12008c. The core routers in AS1 (12008a and 12008b) are only involved in IBGP peering.

Figure 2-3 *Packet Life: Both IP and MPLS*



In order for any data packets to be passed through this network, first the control plane mechanisms have to be set up.

In an IP network, the control plane mechanisms consist of the following:

- **Interior Gateway Protocol (IGP)**—Most often OSPF or IS-IS in service provider networks. Can also be EIGRP, RIP, or just static routing.
- **Border Gateway Protocol (BGP)**—Used to advertise routes that are learned from external neighbors. External BGP (EBGP) is spoken between 7200b and 12008c, as shown in Figure 2-3. 7200b then communicates what it has learned to all other routers in AS1. In this example, 7200b has all other routers as IBGP neighbors; in real-life networks, a Route Reflector (RR) would probably be used. The important point here is that all the routers in AS1 need to learn the route from 7200b.

In an MPLS network, the control plane mechanisms are as follows:

- **IGP**—This is no different from the IGP used for an IP-only network. If the MPLS network were using traffic engineering, the IGP would have to be a link-state protocol, either OSPF or IS-IS. Because traffic engineering is not being considered in this example, the IGP doesn't matter.

- **Label distribution protocol**—The three principal label distribution protocols in an MPLS network are

  — Tag Distribution Protocol (TDP)

  — Label Distribution Protocol (LDP)

  — RSVP

  RSVP is used for traffic engineering and is not considered in this example. TDP and LDP are actually two different versions of the same thing; TDP is older, and LDP is standardized. So, assume that LDP is used to distribute labels.

  What exactly does *label distribution* mean? A *label binding* is an association of a label to a prefix (route). LDP works in conjunction with the IGP to advertise label bindings for all non-BGP routes to its neighbors. LDP neighbors are established over links enabled for LDP. So, when 12008a and 12008b in Figure 2-3 become LDP neighbors, they advertise labels for their IGP-learned routes to each other, but not the BGP routes learned from 7200b.

- **BGP**—Here's where the key difference is between MPLS and non-MPLS networks. Instead of needing to put BGP on every router, BGP is needed only at the edges of the network. Instead of 7200b having three BGP peers (7200a, 12008a, 12008b), it has only one—7200a.

  Why is BGP unnecessary in the core? Because an ingress LER, which has to have full BGP routes, knows the next hop for all BGP-learned routes. A label is put on the packet that corresponds to a packet's BGP next hop, and the packet is delivered across the network to that next hop using MPLS. The section "MPLS Versus IP" deals with this issue in great detail.

  Scaling issues because of large IBGP meshing can be solved using route reflectors or confederations; BGP scales well when deployed properly. However, some people like to totally avoid running BGP in the core. Route flaps outside the network can lead to instability in the core, and the fewer BGP speakers you have, the less you have to manage. In certain cases, the core routers might still need to run BGP for other reasons, such as for multicast.

Forwarding Mechanics

This section explains the differences between forwarding a packet in an IP network and forwarding a packet in an MPLS network. A sample service provider network is used to clarify this concept. So far, you have read about FIB and its role in forwarding packets in a Cisco router. This section covers the role of FIB, LIB, and LFIB tables in forwarding packets in an MPLS-enabled network.

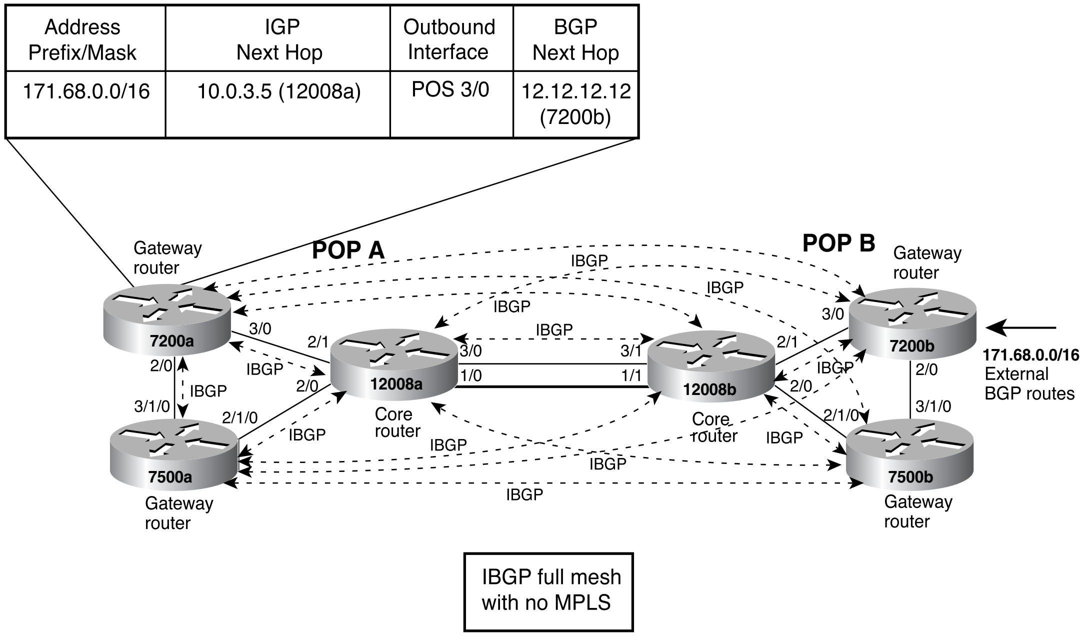
MPLS Versus IP

RFC 3031 defines the MPLS architecture. The points where MPLS forwarding deviates from IP forwarding are as follows:

- IP forwarding is based on the destination IP address and the FIB.
- MPLS forwarding is based on the MPLS label and the Label Forwarding Information Base (LFIB).
- Both MPLS and IP forwarding are done hop-by-hop. IP forwarding involves packet classification at every hop, whereas in MPLS forwarding, the classification is done only by the ingress LSR.

Figure 2-4 illustrates a typical ISP backbone, in which external routes are learned through EBGP and are distributed to the core routers through full IBGP mesh. (Route reflectors or confederations are used in larger cores where a full IBGP mesh would not scale.) The route 171.68.0.0/16 is learned from an external peer by gateway router 7200b. All other routers in the core learn about this route through IBGP. Also, the core routers know how to reach each other from routes learned over IGP routing protocols, such as OSPF or IS-IS.

Figure 2-4 *Forwarding Table on Ingress Router 7200a*



Address Prefix/Mask	IGP Next Hop	Outbound Interface	BGP Next Hop
171.68.0.0/16	10.0.3.5 (12008a)	POS 3/0	12.12.12.12 (7200b)

NOTE Although OSPF and IS-IS seem to be the choice of IGP routing protocols in the service provider backbone, MPLS forwarding doesn't care what your IGP is. For traffic engineering, you need to run IS-IS or OSPF (see Chapter 3, "Information Distribution," for details on why), but if you're not using traffic engineering, you can use any IGP you want.

In Figure 2-4, 7200a is the ingress router that receives packets destined for network 172.68.0.0. Example 2-1 shows the output that displays the contents of the routing table (RIB) on 7200a. As you can see, the entry for 172.68.0.0/16 is the external route that 7200a learned through IBGP.

NOTE You know that 172.168.0.0 is an IBGP-learned route, not an EBGP-learned route because the administrative distance field in the table shows 200, which indicates that it is an IBGP-learned route, not an EBGP-learned route, whose administrative distance is 20.

Example 2-1 *Router 7200a Routing Table*

```
7200a#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

Gateway of last resort is 7.1.5.1 to network 0.0.0.0
B    171.68.0.0/16 [200/0] via 12.12.12.12, 01:10:44
     3.0.0.0/32 is subnetted, 1 subnets
```

When it comes to actually forwarding a data packet, 7200a consults the FIB that is built using the routing table. Example 2-2 shows the FIB entry for 171.68.0.0/16 on 7200a.

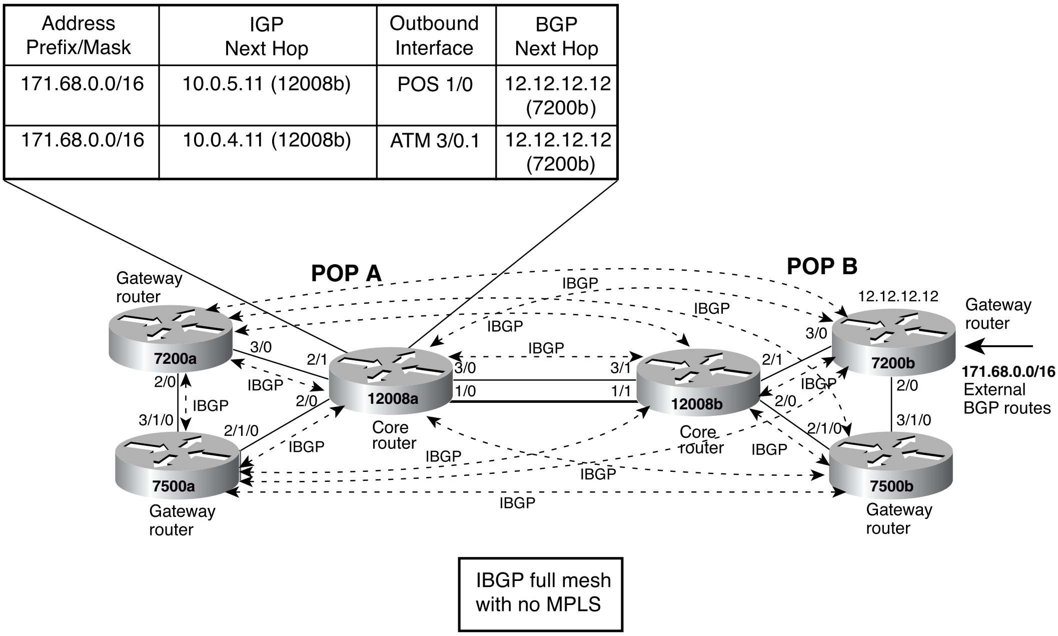
Example 2-2 *FIB Entry for 171.68.0.0/16 on 7200a*

```
7200a#show ip cef 171.68.0.0
171.68.0.0/16, version 69, cached adjacency to POS3/0
0 packets, 0 bytes, wccp tag 139
  via 12.12.12.12, 0 dependencies, recursive
    next hop 10.0.3.5, POS3/0 via 12.12.12.12/32
    valid cached adjacency
```

Now that you have examined the contents of the RIB and FIB on 7200a, you know that the control information has been exchanged and that 7200a is ready to forward data destined for network 171.68.0.0. Similarly, the forwarding tables are created on each of the routers in the core (12008a and 12008b in this example). Next, you need to know how forwarding works step by step.

Consider an IP packet with a destination IP address of 171.68.1.1 arriving at the ingress router (7200a in Figure 2-4). When the packet arrives at the ingress port, the router consults the FIB. The destination IP address of 171.68.1.1 is compared to the FIB entries, and the longest-match entry is selected. As a result of this operation, the ingress router (7200a) knows that the packet has to eventually reach 7200b (the egress router) to exit the network. This usually involves forwarding the packet to one of the immediately connected neighbors, which in this case happens to be 12008a. This process is repeated until the packet reaches the exit point. Figure 2-5 shows the next-hop router 12008a consulting the FIB in order to forward the packet. Note that 12008a has two outbound interface entries in the forwarding table, resulting in load sharing.

Figure 2-5 *Forwarding Table on Core Router 12008a*

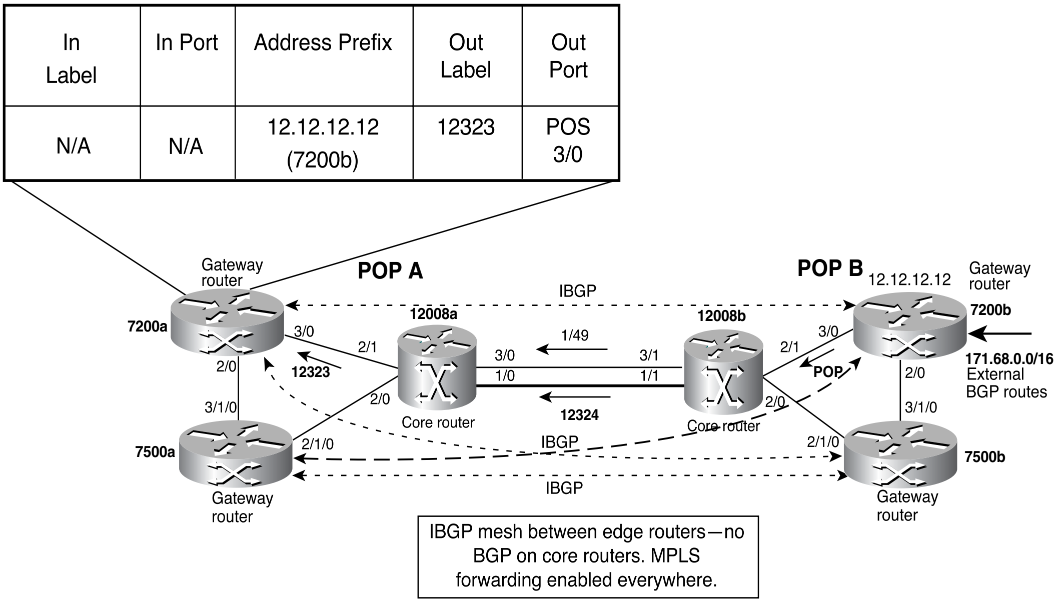


Address Prefix/Mask	IGP Next Hop	Outbound Interface	BGP Next Hop
171.68.0.0/16	10.0.5.11 (12008b)	POS 1/0	12.12.12.12 (7200b)
171.68.0.0/16	10.0.4.11 (12008b)	ATM 3/0.1	12.12.12.12 (7200b)

Because you are so accustomed to IP forwarding, you might take for granted how the packet reaches the network's exit point. On closer observation, this process of consulting the FIB for a longest match and mapping each set of destinations to a next-hop router happens on every router in the forwarding path.

Now take a look at Figure 2-6. MPLS forwarding is now turned on in the core. IBGP is now only between gateway routers and need not be run on the core routers. Right after the IGP converges, the loopback address of 7200b 12.12.12.12 has been learned by 7200a. At this time, LDP also converges. As a result, 7200a receives a label of 12323, corresponding to 7200b's loopback address 12.12.12.12 from 12008a. 12008a itself has received a similar label of 12324 from 12008b. 12008b has received a label of POP from 7200b because it is the penultimate hop router and is responsible for removing the top-level label.

Figure 2-6 *Internet Service Provider Backbone with MPLS Forwarding Enabled*



In Label	In Port	Address Prefix	Out Label	Out Port
N/A	N/A	12.12.12.12 (7200b)	12323	POS 3/0

IBGP mesh between edge routers—no BGP on core routers. MPLS forwarding enabled everywhere.

NOTE Labels distributed by TDP/LDP and RSVP are, in most cases, link-local—meaning it is between any two neighbors and not flooded like OSPF or ISIS. This means that the label value 12323 distributed by 12008a that maps to 12.12.12.12/32 has no relation to the label value 12324 that's received by 12008a, other than the fact that 12008a associated the incoming label value 12323 with the outgoing label value 12324. In other words, 12000b could have given 12008a the label value 42, 967, or 41243, and 12008a still could have distributed the label value 12323.

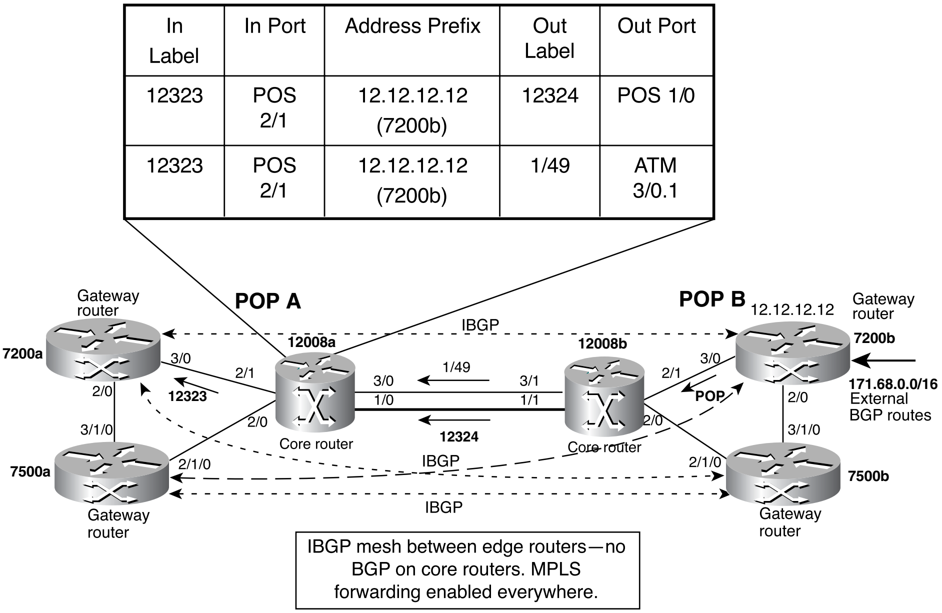
Next, focus your attention on the data plane. Consider the data packet destined for 171.68.1.1 entering the network at 7200a. 7200a still consults the FIB table because the incoming packet is an IP packet. The difference this time is that 7200a is responsible for label imposition. The longest-match IP lookup in the FIB table occurs. As in the case when MPLS forwarding was not turned on in the core, 7200a concludes that the packet needs to eventually reach 7200b—the exit point for this packet. However, now the FIB table has an entry for the label to be imposed for packets destined for 7200b. This is the value of the Out Label column in Figure 2-6, which happens to be 12323. Example 2-3 shows the FIB table on 7200a. If you focus your attention on the highlighted portion of the output, you'll notice the **tags imposed** field that is now present after MPLS forwarding was enabled in the core. This means that if 7200a receives either an IP packet that needs to be forwarded to 12.12.12.12 or an MPLS packet that has a label value of 36, 7200a switches that packet out as an MPLS packet on POS 3/0 with a label value of 12323.

Example 2-3 *FIB Entry for 171.68.0.0 on 7200a After MPLS Forwarding Has Been Turned On*

```
7200a#show ip cef 171.68.0.0 detail
171.68.0.0/16, version 1934, cached adjacency to POS3/0
0 packets, 0 bytes
  tag information from 12.12.12.12/32, shared
    local tag: 36
    fast tag rewrite with PO3/0, point2point, tags imposed {12323}
  via 12.12.12.12, 0 dependencies, recursive
    next hop 10.0.3.5, POS3/0 via 12.12.12.12/32
    valid cached adjacency
    tag rewrite with PO3/0, point2point, tags imposed {12323}
```

Figure 2-7 shows the packet as it enters 12008a, a core router. The packet is an MPLS packet with a label of 12323.

Figure 2-7 *LFIB on Core Router 12008a*



In Label	In Port	Address Prefix	Out Label	Out Port
12323	POS 2/1	12.12.12.12 (7200b)	12324	POS 1/0
12323	POS 2/1	12.12.12.12 (7200b)	1/49	ATM 3/0.1

IBGP mesh between edge routers—no BGP on core routers. MPLS forwarding enabled everywhere.

People who are new to MPLS often wonder how a router knows that this is an MPLS packet and not an IP packet. If you ask yourself how a router knows an IP packet from an IPX packet, you have answered the question. The Layer 2 encapsulation that precedes the IP or IPX header contains a *protocol type* field. In LAN environments, this is *ethertype*. For PPP encapsulation, the Network Control Protocol (NCP) identified what type of Layer 3 packet was being carried, and so on. For MPLS packets, new ethertypes and NCPs have been defined. They are listed in Table 2-2.

Table 2-2 *Layer 2 MPLS Protocol Types*

Encapsulation	Value (in Hexadecimal)
MPLS Control Packet (MPLSCP) for PPP	0x8281
PPP Unicast	0x0281
PPP Multicast	0x0283
LAN Unicast	0x8847
LAN Multicast	0x8848

12008a no longer needs to look at the Layer 3 IP address. It simply consults the LFIB table and knows that there are two places to send an incoming MPLS packet with a label of 12323—POS 1/0 with a label of 12324, and ATM 3/0.1 on the VPI/VCI 1/49. Why are there two paths for this label? Because two equal-cost routes to the destination exist—one via POS and one via ATM. 12008a forwards the packet down POS 1/0 using frame mode. For the path down the ATM 3/0.1 interface, 12008a segments the packet into ATM cells, with each cell using a VPI/VCI value of 1/49. The following discussion focuses on forwarding the packet down the POS 1/0 interface in frame mode. The interesting aspects of cell-mode MPLS are discussed in the section “Label Distribution Concepts.” In this chapter, the term *label*, when used in the context of cell-mode MPLS, refers to ATM VPI/VCI.

When the packet with label 12324 enters 12008b on POS1/1, it goes through the same exercise 12008a went through and consults the LFIB table, as shown in Figure 2-7. But because 12008b is the penultimate-hop router and has received a label of POP from 7200b, it removes the label of 12324, exposes the IP header, and forwards the packet to 7200b. It is important to note that all along, the packet’s destination IP address was 171.68.1.1, for which neither 12008a nor 12008b had a RIB/FIB entry after BGP was removed from their configs. When the packet enters router 7200b, because the packet is IP, again the FIB is consulted. Because 7200b is a gateway (edge) router, it is running BGP and has learned 171.68.0.0/16 over the EBGP connection. Therefore, it can forward the packet.

Example 2-4 shows the LFIB table that router 12008a uses to forward labeled packets. To forward packets to the 171.68.0.0 network, packets need to be sent to 12.12.12.12 (7200b)—the egress router. Upstream routers, such as 7200a, impose a label 12323 that corresponds to the next-hop address 12.12.12.12 (7200b). Notice, in the highlighted part of Example 2-4, that label 12323 falls under the **Local** column because it was what 12008a assigned for FEC 12.12.12.12 and it distributed this label to 7200a—its upstream neighbor.

Example 2-4 *Displaying 12008a’s LFIB Table on the Router*

```
12008a#show mpls forwarding
Local  Outgoing    Prefix            Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
12318  Pop tag     10.0.57.0/24      0          PO1/0       point2point
       1/43        10.0.57.0/24      0          AT3/0.1     point2point
12319  12320       10.0.86.0/24      0          PO1/0       point2point
       1/44        10.0.86.0/24      0          AT3/0.1     point2point
12320  12321       10.1.1.1/32       0          PO1/0       point2point
       1/45        10.1.1.1/32       0          AT3/0.1     point2point
12321  12322       10.1.1.2/32       0          PO1/0       point2point
       1/46        10.1.1.2/32       0          AT3/0.1     point2point
12322  12326       16.16.16.16/32    0          PO1/0       point2point
       1/51        16.16.16.16/32    0          AT3/0.1     point2point
12323  12324       12.12.12.12/32    575        PO1/0       point2point
       1/49        12.12.12.12/32    0          AT3/0.1     point2point
12324  12325       13.13.13.13/32    0          PO1/0       point2point
       1/50        13.13.13.13/32    0          AT3/0.1     point2point
12325  12327       17.17.17.17/32    144        PO1/0       point2point
```

As the output in Example 2-4 shows, 12008a has two ways of reaching 12.12.12.12 (the loopback interface of 7200b).

With MPLS forwarding, just as in IP forwarding, the CEF table (the same as FIB) is consulted. If there are multiple outbound links to the next hop, load sharing is possible, as demonstrated in Example 2-5. The highlighted portion shows that 12008a is doing per-destination load sharing.

Example 2-5 *Router 12008a's CEF Table Shows Load Sharing for Labeled Packets*

```
12008a#show ip cef 12.12.12.12 internal
12.12.12.12/32, version 385, per-destination sharing
0 packets, 0 bytes
  tag information set, shared
    local tag: 12323
  via 10.0.5.11, POS1/0, 0 dependencies
    traffic share 1
    next hop 10.0.5.11, POS1/0
    unresolved
    valid adjacency
    tag rewrite with PO1/0, point2point, tags imposed {12324}
  via 10.0.4.11, ATM3/0.1, 1 dependency
    traffic share 1
    next hop 10.0.4.11, ATM3/0.1
    unresolved
    valid adjacency
    tag rewrite with ATM3/0.1, point2point, tags imposed {1/49(vcd=65)}

  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
  Load distribution: 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 (refcount 2)

  Hash  OK  Interface                 Address         Packets  Tags imposed
  1     Y   POS1/0                    point2point           0    {12324}
  2     Y   ATM3/0.1                  point2point           0    {1/49}
  3     Y   POS1/0                    point2point           0    {12324}
  4     Y   ATM3/0.1                  point2point           0    {1/49}
  5     Y   POS1/0                    point2point           0    {12324}
  6     Y   ATM3/0.1                  point2point           0    {1/49}
  7     Y   POS1/0                    point2point           0    {12324}
  8     Y   ATM3/0.1                  point2point           0    {1/49}
  9     Y   POS1/0                    point2point           0    {12324}
  10    Y   ATM3/0.1                  point2point           0    {1/49}
  11    Y   POS1/0                    point2point           0    {12324}
  12    Y   ATM3/0.1                  point2point           0    {1/49}
  13    Y   POS1/0                    point2point           0    {12324}
  14    Y   ATM3/0.1                  point2point           0    {1/49}
  15    Y   POS1/0                    point2point           0    {12324}
  16    Y   ATM3/0.1                  point2point           0    {1/49}
  refcount 5
```

As shown in Example 2-5, packets destined for 12.12.12.12 on 12008a are load shared on the two outbound links. The load sharing is CEF's standard *per-source-destination* by default. This means that the packet's source and destination IP addresses are hashed. As a result, it uses one of the 16 buckets. You can also turn on *per-packet* load sharing by configuring the outbound interfaces in question.

FIB, LIB, and LFIB and Their Roles in MPLS Forwarding

If you are wondering how the FIB, LIB, and LFIB tables relate to each other, this section summarizes the roles of each of these tables and how they are populated.

The FIB table knows only about IP packets and therefore is consulted only when the incoming packet is an IP packet. Although the incoming packet is an IP packet, the outgoing packet might not be! If one or more label bindings have been received for the packet's destination, the packet is MPLS forwarded. Looking at the CEF table entry for destination 12.12.12.12 on 12008a (as in Example 2-5) tells you whether the outgoing packet is an IP packet or an MPLS packet. If it has an entry of **tags imposed** against the CEF entry, the outgoing packet is MPLS.

In the case of 7200a, destination 12.12.12.12 has an outgoing label of 12323. This results in the packet's entering the next-hop router—12008a with an MPLS label on it. This time, the LFIB table is consulted on 12008a. Example 2-6 shows that if an MPLS packet came in with a label of 12323, it would have to be switched out of the ATM 3/0.1 interface with a VPI/VCI value of 1/49 or with a label of 12324 on interface POS1/0.

Example 2-6 shows a segment of the LFIB table corresponding to 12.12.12.12.

Example 2-6 *Segment of 12008a's LFIB Table Corresponding to 12.12.12.12*

```
Local  Outgoing    Prefix             Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id       switched   interface
12323  12324       12.12.12.12/32     575        PO1/0       point2point
       1/49        12.12.12.12/32     0          AT3/0.1     point2point
```

Now consider the case of the packet being switched over the POS link with an MPLS label of 12324.

Where did all these labels come from? Labels can be distributed between LSRs using various methods. If LDP or TDP protocols are used, label bindings are exchanged between LSRs and their neighbors. This information is stored in the LIB. You can view the LIB's contents using the **show mpls ip bindings** *address* command. You can see in Example 2-7 the contents of the LIB that holds the label bindings for 12.12.12.12.

Example 2-7 *Viewing LIB Contents*

```
12008a#show mpls ip binding 12.12.12.12 32
  12.12.12.12/32
        in label:     12325
        out label:    36         lsr: 4.4.4.4:0
        out label:    12324      lsr: 11.11.11.11:0
        out label:    37         lsr: 3.3.3.3:0
        out vc label: 1/49       lsr: 11.11.11.11:2    ATM3/0.1
                      Active     ingress 1 hop (vcd 18)
```

Notice in Example 2-7 that several *remote bindings* exist in the LIB, but the forwarding table shows only two entries. This is because only the bindings that are received from the current IGP next-hop router are used, even though all the bindings are retained on the Cisco routers because they employ the liberal retention mode (discussed further in the section "Liberal and Conservative Retention Modes"). Look at 12008a's routing entry for 12.12.12.12 in Example 2-8.

Example 2-8 *RIB Entry for 12.12.12.12 on 12008a*

```
12008a#show ip route 12.12.12.12
Routing entry for 12.12.12.12/32
  Known via "ospf 100", distance 110, metric 3, type intra area
  Last update from 10.0.4.11 on ATM3/0.1, 00:41:50 ago
  Routing Descriptor Blocks:
  * 10.0.5.11, from 12.12.12.12, 00:41:50 ago, via POS1/0
      Route metric is 3, traffic share count is 1
    10.0.4.11, from 12.12.12.12, 00:41:50 ago, via ATM3/0.1
      Route metric is 3, traffic share count is 1
      Route metric is 3, traffic share count is 1
```

To figure out exactly how labels 12324 and 1/49 became the outgoing labels for 12.12.12.12, as shown in Example 2-6, you have to first look at the next hops for 12.12.12.12 from Example 2-8. They happen to be 10.0.5.11 and 10.0.4.11 (which are highlighted in Example 2-8). Incidentally, these next hops happen to be two links of 12008b, whose router ID is 11.11.11.11. Using this information, you can go back to the LIB (refer to Example 2-7) to look for what label bindings 12008a received from 11.11.11.11. You'll find labels 12324 and 1/49 over the two interfaces POS 1/0 and ATM 3/0.1, respectively.

Table 2-3 summarizes the input and output packet types and the table used for forwarding.

Table 2-3 *I/O Packet Types and Related Forwarding Tables*

Packet Type	Table Used for Packet Lookup	How to Look at This Table
IP to IP	FIB	**show ip cef**
IP to MPLS	FIB	**show ip cef**
MPLS to MPLS	LFIB	**show mpls forwarding-table**
MPLS to IP	LFIB	**show mpls forwarding-table**

Label Distribution Concepts

The preceding section elaborated on how forwarding works after the FIB and LFIB tables have been populated. This section covers the various methods of distributing label bindings.

When labels are distributed, what's actually distributed is a label, an IP prefix, and a mask length. Generally, this entire process is called *label distribution* rather than *label, prefix, and mask distribution*.

To understand how LSRs generate and distribute labels, you need to understand some terminology introduced in RFC 3031.

Ordered Versus Independent Control

As far as generating labels is concerned, regardless of what control method is applicable, the LSRs generate labels independently and have no relation to the received labels. As you would anticipate, there have to be *reserved* label values that are either used for control or have some special meaning.

The label values 0 to 15 are reserved. This means that the lowest label number you see that maps to an IP prefix is 16. Because the label space is 20 bits, the highest label you ever see advertised is 2^{20}–1, or 1,048,575. This is subject to change, though. As long as an allocated label value is between 16 and 1,048,575, it's legal.

Only four out of the 16 reserved label values are currently defined in RFC 3032, "MPLS Label Stack Encoding":

0—IPv4 Explicit Null Label
1—Router Alert Label
2—IPv6 Explicit Null Label
3—Implicit Null Label

Except for MPLS edge applications, labels are generated only for IGP-learned prefixes (including static routes) in the routing. Why aren't labels allocated for BGP-learned routes? Because doing so is completely unnecessary. Again, for IPv4 routes (the non-MPLS-VPN case), if the egress LER set next-hop-self in BGP, all that is needed is a label and an IGP route for the next hop of the BGP-learned route. For example, consider the external route 171.68.0.0 that is learned by 7200b in Figure 2-6. By doing next-hop-self, 7200b sets the next hop for 171.68.0.0 to 12.12.12.12 (7200b's BGP router ID and Loopback0) before it advertises 171.68.0.0 to 7200a with IBGP. Because IGP routes have been exchanged and label distribution has occurred, 7200a has a label for 12.12.12.12. If it uses this label for packets destined for 171.68.0.0, the packet is delivered to 7200b as a result of MPLS forwarding. There is no need for a label for 171.68.0.0. Any packets destined for 171.68.0.0 are simply delivered to 12.12.12.12, which then routes the IP packet normally.